
Amazon Simple Storage Service

Guia do usuário

Versão da API 2006-03-01



Amazon Simple Storage Service: Guia do usuário

Copyright © 2022 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

O que é o Amazon S3?	1
Recursos do Amazon S3	1
Classes de armazenamento	1
Gerenciamento de armazenamento	1
Gerenciamento de acesso	2
Processamento de dados	2
Registro e monitoramento do armazenamento	3
Análise e insights	3
Consistência forte	3
Como funciona o Amazon S3	4
Buckets	4
Objetos	5
Chaves	5
Versionamento do S3	5
ID da versão	5
Política de bucket	5
Pontos de acesso do S3	6
Listas de controle de acesso (ACLs)	6
Regiões	6
Modelo de consistência de dados do Amazon S3	7
Aplicações simultâneas	8
Serviços relacionados da	9
Acesso ao Amazon S3	9
AWS Management Console	10
AWS Command Line Interface	10
AWS SDKs	10
API REST do Amazon S3	10
Pagar pelo Amazon S3	11
Conformidade do PCI DSS	11
Começar a usar	12
Configuração	12
Cadastro na AWS	13
Criar um usuário do IAM	13
Fazer login como usuário do IAM	14
Etapa 1: Criar um bucket	15
Etapa 2: Fazer upload de um objeto	17
Etapa 3: fazer download de um objeto	17
Uso do console do S3	17
Etapa 4: copiar um objeto	18
Etapa 5: excluir os objetos e o bucket	19
Excluir um objeto	19
Esvaziar o bucket	19
Excluir bucket	20
Próximas etapas	20
Entender casos de uso comuns	20
Controlar o acesso a buckets e objetos	21
Explore o treinamento e o suporte	21
Gerencie e monitore seu armazenamento	22
Desenvolvimento com o Amazon S3	22
Controle de acesso	24
Criar um bucket	24
Armazenar e compartilhar dados	26
Compartilhar recursos	27
Proteger dados	27

Tutoriais	30
Como transformar dados com o S3 Object Lambda	30
Pré-requisitos	31
Etapa 1: criar um bucket do S3	33
Etapa 2: fazer upload do arquivo para seu bucket do S3	33
Etapa 3: criar um ponto de acesso do S3	34
Etapa 4: Criar uma função Lambda	35
Etapa 5: configurar uma política do IAM para a função de execução da função Lambda	39
Etapa 6: criar um ponto de acesso do Object Lambda do S3	40
Etapa 7: exibir os dados transformados	41
Etapa 8: limpar	43
Next steps (Próximas etapas)	45
Detecção e edição de dados PII	45
Pré-requisitos: criar um usuário do IAM com permissões	47
Etapa 1: criar um bucket do S3	48
Etapa 2: fazer upload do arquivo para seu bucket do S3	49
Etapa 3: criar um ponto de acesso do S3	49
Etapa 4: configurar e implantar uma função Lambda pré-construída	50
Etapa 5: criar um ponto de acesso do Object Lambda do S3	51
Etapa 6: usar o ponto de acesso do S3 Object Lambda para recuperar o arquivo editado	52
Etapa 7: limpeza	53
Next steps (Próximas etapas)	55
Hospedagem de streaming de vídeo	56
Pré-requisitos: registrar e configurar um domínio personalizado com o Route 53	57
Etapa 1: criar um bucket do S3	58
Etapa 2: carregar um vídeo no bucket do S3	59
Etapa 3: Criar uma identidade do acesso de origem do CloudFront	59
Etapa 4: criar uma distribuição do CloudFront	60
Etapa 5: Acessar o vídeo por meio da distribuição do CloudFront	61
Etapa 6: Configurar sua distribuição do CloudFront para usar o seu nome de domínio personalizado	62
Etapa 7: acessar o vídeo do S3 por meio da distribuição do CloudFront com o nome de domínio personalizado	66
(Opcional) Etapa 8: exibir dados sobre solicitações recebidas pela distribuição do CloudFront	66
Etapa 9: limpeza	67
Next steps (Próximas etapas)	70
Vídeos de transcodificação em lote	70
Pré-requisitos	72
Etapa 1: Criar um bucket do S3 para os arquivos de mídia de saída	72
Etapa 2: criar uma função do IAM para MediaConvert	74
Etapa 3: criar uma função do IAM para a função Lambda	74
Etapa 4: Criar uma função do Lambda para transcodificação de vídeo	76
Etapa 5: Configurar o inventário do Amazon S3 para seu bucket de origem do S3	88
Etapa 6: criar uma função do IAM para operações em lote do S3	91
Etapa 7: criar e executar um trabalho de operações em lote do S3	93
Etapa 8: Conferir os arquivos de mídia de saída do bucket de destino do S3	97
Etapa 9: limpeza	97
Next steps (Próximas etapas)	99
Configurar um site estático	100
Etapa 1: Criar um bucket	100
Etapa 2: Habilitar hospedagem de site estático	100
Etapa 3: editar as configurações do Bloqueio de acesso público	101
Etapa 4: Adicionar política de bucket que torna o conteúdo do bucket publicamente disponível	102
Etapa 5: Configurar um documento de índice	103
Etapa 6: configurar um documento de erros	104
Etapa 7: testar o endpoint do site	105
Etapa 8: Limpar	105

Configurar um site estático usando um domínio personalizado	105
Antes de começar	106
Etapa 1: Registrar um domínio personalizado no Route 53	107
Etapa 2: Criar dois buckets	107
Etapa 3: Configurar o bucket de domínio raiz	108
Etapa 4: Configurar o bucket de subdomínio para redirecionamento	109
Etapa 5: Configurar o registro em log	109
Etapa 6: Fazer upload do conteúdo do site e do índice	110
Etapa 7: carregar um documento de erros	111
Etapa 8: Editar o bloqueio de acesso público	111
Etapa 9: Anexar uma política de bucket	113
Etapa 10: Testar o endpoint de domínio	114
Etapa 11: Adicionar registros de alias	114
Etapa 12: Testar o site	117
Acelerar seu site com o Amazon CloudFront	118
Limpar recursos de exemplo	121
Trabalhar com buckets	123
Visão geral dos buckets	123
Sobre permissões	124
Gerenciar o acesso público aos buckets	125
Configuração do bucket	125
Regras de nomeação	127
Exemplo de nomes de bucket	128
Criação de um bucket	129
Visualizar propriedades de buckets	134
Métodos de acesso a um bucket	135
Acesso no estilo de hospedagem virtual	136
Acesso ao estilo de caminho	136
Acessar um bucket do S3 por IPv6	136
Acessar um bucket por meio de pontos de acesso do S3	137
Acessar um bucket usando S3://	137
Esvaziar um bucket	137
Excluir um bucket	139
Definir criptografia de bucket padrão	142
Usar criptografia para operações entre contas	143
Como usar a criptografia padrão com a replicação	143
Usar Chaves de bucket do Amazon S3 com criptografia padrão	144
Habilitar criptografia padrão	144
Monitorar a criptografia padrão	147
Configurar o Transfer Acceleration	147
Por que usar o Transfer Acceleration?	148
Requisitos para usar o Transfer Acceleration	148
Conceitos básicos	149
Habilitar o Transfer Acceleration	150
Ferramenta de comparação de velocidade	155
Uso de Pagamento pelo solicitante	155
Como funcionam as cobranças de Pagamento pelo solicitante	156
Configurar Pagamento pelo solicitante	156
Recuperação da configuração de requestPayment	158
Fazer download de objetos em buckets de Pagamento pelo solicitante	158
Restrições e limitações	159
Trabalhar com objetos	161
Objetos	161
Sub-recursos	162
Criar chaves de objeto	162
Diretrizes de nomeação de chave de objeto	163
Trabalhar com metadados	166

Metadados do objeto definidos pelo sistema	166
Metadados do objeto definidos pelo usuário	168
Editar os metadados do objeto	169
Fazer upload de objetos	171
Usar multipart upload	180
Processo multipart upload	180
Somas de verificação com operações de carregamento fracionado	182
Operações simultâneas de multipart upload	182
Multipart upload e definição de preço	182
Suporte de API para multipart upload	183
Suporte da AWS Command Line Interface para carregamento fracionado	183
Suporte do AWS SDK para upload fracionado	183
API de multipart upload e permissões	183
Configurar uma política de ciclo de vida	185
Fazer upload de um objeto usando multipart upload	187
Fazer upload de um diretório	200
Listar multipart uploads	202
Monitorar um multipart upload	204
Abortar um multipart upload	206
Copiar um objeto	210
Limites do multipart upload	214
Cópia de objetos	215
Para copiar um objeto	216
Fazer download de um objeto	223
Verificar a integridade do objeto	229
Usar algoritmos de soma de verificação compatíveis	229
Usar Content-MD5 ao carregar objetos	235
Usar Content-MD5 e a ETag para verificar objetos carregados	235
Usar somas de verificação à direita	236
Usar somas de verificação de nível parcial para carregamentos fracionados	236
Excluir objetos	237
Excluir objetos de um bucket habilitado para versionamento de maneira programática	238
Excluir objetos de um bucket com MFA habilitada	238
Excluir um único objeto	239
Excluir vários objetos	246
Organizar e listar objetos	258
Usar prefixos	258
Listar objetos	260
Usar pastas	269
Exibir uma visão geral do objeto	271
Visualizar propriedades de objeto	272
Usar pre-signed URLs	273
Limitar recursos de pre-signed URLs	273
Quem pode criar um URL pré-assinado	274
Quando o Amazon S3 confere a data e hora de validade de um URL pré-assinado?	274
Compartilhar objetos	275
Carregar objetos	278
Excluir um objeto	287
Transformar objetos	288
Criar pontos de acesso do Object Lambda	290
Usar pontos de acesso do Amazon S3 Object Lambda	295
Conceitos básicos de um template do AWS CloudFormation	296
Configurar políticas do IAM	299
Escrever funções do Lambda	302
Usar funções criadas pela AWS	314
Práticas recomendadas e diretrizes para o S3 Object Lambda	316
Considerações sobre segurança	317

Tutoriais do S3 Object Lambda	318
Trabalhar com pontos de acesso	320
Configurar políticas do IAM	320
Chaves de condição	321
Delegar controle de acesso a pontos de acesso	321
Exemplos de política de ponto de acesso	322
Criar pontos de acesso	325
Regras para nomear pontos de acesso do Amazon S3	325
Criar um ponto de acesso	325
Criar pontos de acesso restritos a uma VPC	327
Gerenciar o acesso público	329
Usar pontos de acesso	329
Monitorar e registrar	330
Gerenciar pontos de acesso	331
Usar um alias em estilo de bucket para seu ponto de acesso	333
Usar pontos de acesso	335
Restrições e limitações	337
Como trabalhar com pontos de acesso de várias regiões	339
Criação de pontos de acesso multirregionais	340
Regras para nomear pontos de acesso multirregionais do Amazon S3	341
Regras para escolher buckets para pontos de acesso multirregionais do Amazon S3	342
Bloqueio de acesso público de pontos de acesso multirregionais do Amazon S3	343
Criação de pontos de acesso multirregionais do Amazon S3	343
Como configurar o AWS PrivateLink	344
Como usar um ponto de acesso multirregional	346
Nomes de host do ponto de acesso multirregional	347
Pontos de acesso multirregionais e Amazon S3 Transfer Acceleration	348
Permissões do ponto de acesso multirregional	348
Roteamento de solicitação	350
Replicação do bucket	351
Operações compatíveis	352
Gerenciamento de pontos de acesso de várias regiões	352
Monitorar e registrar	353
Monitoramento e registro de solicitações feitas para APIs de gerenciamento de pontos de acesso multirregionais	354
Uso do CloudTrail	354
Restrições e limitações	355
Segurança	357
Proteção de dados	358
Criptografia de dados	358
Criptografia do lado do servidor	359
Uso de criptografia do lado do cliente	405
Privacidade entre redes	409
Tráfego entre clientes de serviço e on-premises e os aplicativos	409
Tráfego entre recursos da AWS na mesma região	409
AWS PrivateLink for Amazon S3	410
Tipos de VPC endpoints	410
Restrições e limitações do AWS PrivateLink do Amazon S3	411
Criar um endpoint da VPC	411
Acessar endpoints da interface do Amazon S3	411
Acessar buckets e pontos de acesso do S3 a partir de endpoints de interface S3	412
Atualizar uma configuração de DNS on-premises	415
Criar uma política de VPC endpoint	416
Identity and Access Management	419
Visão geral	419
Diretrizes para políticas de acesso	425
Solicitar autorização	429

Políticas de bucket e políticas de usuário	437
Políticas gerenciadas pela AWS	585
Gerenciar o acesso com ACLs	587
Usar o CORS	607
Bloquear o acesso público	618
Revisar o acesso ao bucket	629
Verificar a propriedade do bucket	634
Controlar a propriedade do objeto	638
Configurações do Object Ownership	639
Alterações introduzidas ao desabilitar ACLs	640
Pré-requisitos para desabilitar ACLs	642
Permissões do Object Ownership	643
Desabilitar ACLs para todos os novos buckets	644
Replicação e Object Ownership	644
Configurar o Object Ownership	644
Pré-requisitos para desabilitar ACLs	645
Criação de um bucket	653
Configurar o Object Ownership	656
Visualizar configurações do Object Ownership	659
Desabilitar ACLs para todos os novos buckets	660
Solução de problemas	662
Registro em log e monitoramento	664
Validação de conformidade	666
Resiliência	667
Criptografia de backup	669
Segurança da infraestrutura	670
Análise de configuração e vulnerabilidade	671
Melhores práticas de segurança	672
Melhores práticas de segurança preventivas do Amazon S3	672
Melhores práticas de auditoria e monitoramento do Amazon S3	675
Gerenciar o armazenamento	678
Usando o versionamento do S3	678
Buckets não versionados, habilitados para versão e suspensos de versão	679
Usando o versionamento do S3 com o ciclo de vida do S3	679
Versionamento do S3	680
Habilitar o versionamento em buckets	683
Configurando a exclusão de MFA	688
Trabalhando com objetos habilitados para versão	689
Trabalhando com objetos suspensos de versão	708
Usar o AWS Backup para o Amazon S3	711
Trabalhando com objetos arquivados	712
Opções de recuperação de arquivamento	713
Restaurar um objeto arquivado	715
Consultar objetos arquivados	719
Usar o bloqueio de objetos	722
Bloqueio de objetos do S3	723
Configurar o bloqueio de objetos no console	727
Gerenciar o bloqueio de objetos	728
Gerenciamento de classes de armazenamento	731
Objetos acessados frequentemente	732
Otimização automática de dados com padrões de acesso alterados ou desconhecidos	732
Objetos acessados com pouca frequência	733
Arquivamento de objetos	734
Amazon S3 on Outposts	735
Comparação de classes de armazenamento	735
Configurar a classe de armazenamento de um objeto	736
Amazon S3 Intelligent-Tiering	737

Como o S3 Intelligent-Tiering funciona	737
Como usar o S3 Intelligent-Tiering	739
Como gerenciar o S3 Intelligent-Tiering	742
Gerenciando o ciclo de vida	746
Gerenciando o ciclo de vida do objeto	747
Criando uma configuração de ciclo de vida	747
Fazer a transição de objetos	748
Expirando objetos	753
Definir a configuração do ciclo de vida	753
Usando outras configurações de bucket	765
Configurar notificações de evento de ciclo de vida	766
Elementos de configuração do ciclo de vida	767
Exemplos de configuração do S3 Lifecycle	775
Gerenciamento de inventário	786
Buckets do Amazon S3 Inventory	787
Listas de inventário	787
Configurar o inventário do Amazon S3	789
Configuração de notificações para conclusão de inventário	794
Localização de inventário	794
Consultar o inventário com o Athena	797
Converter strings de ID de versão vazias em cadeias de caracteres null	799
Replicação de objetos	801
Por que usar a replicação	802
Quando usar a replicação entre regiões	803
Quando usar Replicação na mesma região	803
Quando usar S3 Batch Replication	803
Requisitos para replicação	804
O que é replicado?	804
Configuração da replicação	807
Replicar objetos existentes	849
Configurações adicionais	857
Obtenção de status da replicação	873
Solução de problemas	875
Considerações adicionais	877
Uso de tags de objeto	878
Operações de API relacionadas à marcação de objetos	880
Configurações adicionais	881
Controle de acesso	882
Gerenciar tags de objeto	884
Usar tags de alocação de custos	888
Mais informações	889
Relatórios de uso e faturamento	889
Uso do Amazon S3 Select	908
Requisitos e limites	908
Criar uma solicitação	909
Erros	909
Exemplos do S3 Select	910
Referência SQL	912
Uso do Batch Operations	938
Conceitos básicos do Batch Operations	938
Conceder permissões	939
Criar um trabalho	946
Operações compatíveis	953
Gerenciar trabalhos	979
Rastreamento de relatórios de status e conclusão	982
Usar tags	992
Gerenciar o Bloqueio de objetos do S3	1003

Tutorial Operações em lote do S3	1019
Monitorar o Amazon S3	1020
Ferramentas de monitoramento	1020
Ferramentas automatizadas	1020
Ferramentas manuais	1021
Opções de registro em log	1021
Registrar em log com o CloudTrail	1023
Usar logs do CloudTrail com logs de acesso ao servidor do Amazon S3 e CloudWatch Logs	1023
Rastreamento do CloudTrail com chamadas de API SOAP do Amazon S3	1024
Eventos do CloudTrail	1025
Exemplos de arquivos de log	1029
Habilitar o CloudTrail	1033
Identificar solicitações do S3	1035
Registrando acesso ao servidor	1041
Como faço para habilitar a entrega de logs?	1041
Formato da chave de objeto de log	1042
Como os logs são entregues?	1042
Entrega de logs do servidor de melhor esforço	1043
As alterações do status do registro de bucket em logs entram em vigor ao longo do tempo	1043
Habilitar o log de acesso ao servidor do	1043
Formato do log	1052
Excluir arquivos de log	1063
Identificar solicitações do S3	1063
Monitoramento de métricas com o CloudWatch	1067
Métricas e dimensões	1069
Acessar métricas do CloudWatch	1076
Configurações de métricas do CloudWatch	1077
Notificações de eventos do Amazon S3	1083
Visão geral	1084
Tipos e destinos de notificações	1085
Usar o SQS, o SNS e o Lambda	1089
Uso do EventBridge	1108
Usar análises e insights	1116
Análise de classe de armazenamento	1116
Como configurar a análise de classe de armazenamento	1117
Análise de classe de armazenamento	1117
Como exportar dados de análise de classe de armazenamento?	1119
Configurar análise de classe de armazenamento	1120
S3 Storage Lens	1121
Noções básicas sobre o S3 Storage Lens	1123
Trabalhar com organizações	1128
Permissões do S3 Storage Lens	1130
Exibição de métricas de armazenamento	1133
Como usar o Amazon S3 Storage Lens para otimizar seus custos de armazenamento	1149
Glossário de métricas	1152
Trabalhar com o S3 Storage Lens	1158
Rastreamento de solicitações usando X-Ray	1186
Como o X-ray funciona com o Amazon S3	1187
Regiões disponíveis	1187
Hospedar sites estáticos	1188
Endpoints de site	1188
Exemplos de endpoint de site	1189
Adicionar um registro DNS CNAME	1189
Usar um domínio personalizado com o Route 53	1190
Principais diferenças entre um endpoint de site e um endpoint de API REST	1190
Habilitar a hospedagem de sites	1190
Configurar um documento de índice	1194

Pastas e documentos de índice	1194
Configurar um documento de índice	1195
Configurar um documento de erro personalizado	1196
Códigos de resposta HTTP do Amazon S3	1196
Configurar um documento de erro personalizado	1198
Configuração de permissões para acesso ao site	1199
Etapa 1: Editar configurações de bloqueio de acesso público do S3	1199
Etapa 2: Adicionar uma política de bucket	1200
Listas de controle de acesso de objetos	1201
Registro de tráfego da Web	1202
Configuração de um redirecionamento	1203
Redirecionar solicitações para outro host	1203
Configurar regras de redirecionamento	1204
Redirecionar solicitações para um objeto	1209
Desenvolvimento com o Amazon S3	1211
Fazer solicitações	1211
Sobre as chaves de acesso	1211
Endpoints de solicitações	1213
Fazer solicitações por meio do IPv6	1213
Fazer solicitações usando os AWS SDKs	1220
Fazer solicitações usando a API REST	1247
Usar a AWS CLI	1257
Uso da SDKs AWS	1258
Como trabalhar com AWS SDKs	1259
Especificar a versão da assinatura na autenticação de solicitações	1260
Usar a AWS SDK for Java	1266
Usar a AWS SDK for .NET	1267
Usar o AWS SDK for PHP e executar exemplos do PHP	1268
Usar o AWS SDK for Ruby - versão 3	1269
Usar a AWS SDK for Python (Boto)	1271
Usar os AWS Mobile SDKs for iOS e Android	1271
Uso da biblioteca JavaScript do AWS Amplify	1271
Usar a AWS SDK for JavaScript	1271
Uso dos REST API	1272
Roteamento de solicitação	1272
Tratamento de erros	1276
A resposta de erro de REST	1276
A resposta de erro de SOAP	1278
Melhores práticas com relação a erros do Amazon S3	1278
Referência	1279
Apêndice A: Usar a API SOAP	1279
Apêndice B: Autenticação de solicitações (AWS Signature Version 2)	1282
Otimizar a performance do Amazon S3	1312
Diretrizes de performance	1313
Avaliar a performance	1313
Dimensionar na horizontal	1313
Usar consulta na escala de bytes	1313
Solicitações de repetição	1314
Combinar o Amazon S3 e o Amazon EC2 na mesma região	1314
Usar o Transfer Acceleration para minimizar a latência	1314
Usar os AWS SDKs mais recentes	1314
Padrões de design de performance	1315
Armazenar conteúdo acessado com frequência em cache	1315
Tempo limite e repetição para aplicativos sensíveis à latência	1315
Dimensionamento horizontal e paralelização de solicitações	1316
Acelerar transferências de dados geograficamente dispersas	1317
O que é o S3 on Outposts?	1318

Como funciona o S3 on Outposts	1318
Regiões	1318
Buckets	1319
Objetos	1319
Chaves	1319
Classe de armazenamento e criptografia	1320
Política de bucket	1320
Pontos de acesso do S3 on Outposts	1320
Recursos do S3 on Outposts	1321
Gerenciamento de acesso	1321
Registro e monitoramento do armazenamento	1321
Consistência forte	1321
Serviços relacionados da	1322
Acessar o S3 no Outposts	1322
AWS Management Console	1322
AWS Command Line Interface	1322
AWS SDKs	1322
Pagar pelo S3 on Outposts	1323
Next steps (Próximas etapas)	1323
Configurar seu Outpost	1323
Pedir um novo Outpost	1323
Adicionar armazenamento do Amazon S3 a um Outpost existente	1324
Em que aspectos o S3 on Outposts é diferente?	1324
Especificações	1324
Operações de API compatíveis	1325
Recursos não compatíveis do Amazon S3	1325
Restrições de rede	1326
Conceitos básicos do S3 no Outposts	1326
Configurar o IAM	1327
Uso do console do S3	1331
Usar a AWS CLI e o SDK para Java	1333
Redes para S3 on Outposts	1338
Escolher seu tipo de acesso às redes	1338
Acessar buckets e objetos do S3 on Outposts	1339
Gerenciar conexões usando interfaces de rede elástica entre contas	1339
Trabalhar com buckets do S3 on Outposts	1339
Buckets	1339
Pontos de acesso	1340
Endpoints	1340
Operações de API no S3 on Outposts	1340
Criar e gerenciar o bucket do S3 no Outposts	1341
Criação de um bucket	1342
Adicionar etiquetas	1344
Criar e gerenciar uma configuração de ciclo de vida	1345
Uso de políticas de bucket	1350
Listar buckets	1353
Obter um bucket	1354
Excluir bucket	1355
Trabalhar com pontos de acesso	1356
Trabalhar com endpoints do	1362
Trabalhar com objetos usando o S3 on Outposts	1366
Copiar um objeto	1367
Obter um objeto	1368
Listar objetos	1370
Excluir objetos	1372
Usar o HeadBucket	1375
Executar um carregamento fracionado	1376

Usar pre-signed URLs	1381
Segurança	1390
Criptografia de dados	1391
AWS PrivateLink para S3 on Outposts	1391
Chaves de política do Signature Version 4 (SigV4)	1395
Gerenciar o armazenamento do S3 on Outposts	1398
Métricas do CloudWatch	1398
Amazon CloudWatch Events	1399
Logs do CloudTrail	1400
Compartilhar o S3 on Outposts	1401
Outros serviços	1404
Desenvolver com o S3 on Outposts	1404
APIs do S3 on Outposts	1404
Configurar o cliente de controle do S3	1406
Exemplos de código	1407
Ações	1408
Adicionar regras de CORS a um bucket	1409
Adicionar uma configuração de ciclo de vida a um bucket	1414
Adicionar uma política a um bucket	1417
Copiar um objeto de um bucket para outro	1422
Criar um bucket	1431
Excluir regras de CORS de um bucket	1440
Excluir uma política de um bucket	1441
Excluir um bucket vazio	1445
Excluir um objeto	1451
Excluir vários objetos	1457
Excluir a configuração de ciclo de vida de um bucket	1465
Excluir a configuração de site de um bucket	1466
Determinar a existência e o tipo de conteúdo de um objeto	1469
Determinar a existência de um bucket	1470
Obter regras de CORS para um bucket	1471
Obter um objeto de um bucket	1473
Obter a ACL de um bucket	1483
Obter a ACL de um objeto	1487
Obter o local da região de um bucket	1491
Obter a configuração de ciclo de vida de um bucket	1492
Obter a política de um bucket	1493
Obter a configuração do site de um bucket	1497
Listar buckets	1499
Listar carregamentos fracionados em andamento	1504
Listar as versões de objeto em um bucket	1504
Listar objetos em um bucket	1505
Restaurar uma cópia arquivada de um objeto	1514
Definir uma nova ACL para um bucket	1515
Definir a ACL de um objeto	1518
Definir a configuração de site de um bucket	1519
Carregar um objeto em um bucket	1523
Cenários	1537
Criar um URL pré-assinado	1537
Começar a usar buckets e objetos do Amazon S3	1545
Gerenciar objetos versionados em lotes com uma função do Lambda	1580
Usar um gerenciador de transferências para carregar e baixar arquivos	1581
Trabalhar com objetos versionados	1594
Exemplos entre serviços	1599
Criar uma aplicação Amazon Transcribe	1599
Converter texto em fala e de volta em texto	1600
Criar um cluster do Amazon EMR de longa duração e executar várias etapas	1600

Criar um cluster do Amazon EMR de curta duração e executar uma etapa usando	1601
Criar uma aplicação de exploração do Amazon Textract	1602
Detectar EPI em imagens	1603
Detectar entidades em texto extraído de uma imagem	1604
Detectar faces em uma imagem	1604
Detectar objetos em imagens	1605
Detectar pessoas e objetos em um vídeo	1607
Salvar o EXIF e outras informações de imagem	1608
Solução de problemas	1610
Solução de problemas do Amazon S3 por sintoma	1610
Aumentos significativos em respostas HTTP 503 para solicitações para buckets com versionamento habilitado	1610
Comportamento inesperado ao acessar buckets definidos com CORS	1611
Obter IDs de solicitação do Amazon S3 para AWS Support	1611
Usar o HTTP para obter IDs de solicitação	1611
Usar um navegador da web para obter IDs de solicitação	1612
Uso dos AWS SDKs para obter IDs de solicitação	1612
Usar o AWS CLI para obter IDs de solicitação	1614
Tópicos relacionados	1614
Histórico do documento	1615
Atualizações anteriores	1627
Glossário da AWS	1643

O que é o Amazon S3?

O Amazon Simple Storage Service (Amazon S3) é um serviço de armazenamento de objetos que oferece escalabilidade líder do setor, disponibilidade de dados, segurança e performance. Clientes de todos os tamanhos e setores podem usar o Amazon S3 para armazenar e proteger qualquer volume de dados para uma variedade de casos de uso, como data lakes, sites, aplicações móveis, backup e restauração, arquivamento, aplicações corporativas, dispositivos IoT e análises de big data. O Amazon S3 fornece recursos de gerenciamento para que você possa otimizar, organizar e configurar o acesso aos seus dados para atender aos seus requisitos específicos de negócios, organizacionais e de compatibilidade.

Tópicos

- [Recursos do Amazon S3 \(p. 1\)](#)
- [Como funciona o Amazon S3 \(p. 4\)](#)
- [Modelo de consistência de dados do Amazon S3 \(p. 7\)](#)
- [Serviços relacionados da \(p. 9\)](#)
- [Acesso ao Amazon S3 \(p. 9\)](#)
- [Pagar pelo Amazon S3 \(p. 11\)](#)
- [Conformidade do PCI DSS \(p. 11\)](#)

Recursos do Amazon S3

Classes de armazenamento

O Amazon S3 oferece uma ampla variedade de classes de armazenamento para diferentes casos de uso. Por exemplo, você pode armazenar dados de produção essenciais à missão no S3 Standard para acesso frequente, economizar custos armazenando dados acessados com pouca frequência no S3 Standard-IA ou S3 One Zone-IA e arquivar dados com os menores custos no S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval e no S3 Glacier Deep Archive.

Você pode armazenar dados com padrões de acesso alterados ou desconhecidos no S3 Intelligent-Tiering, o que otimiza os custos de armazenamento movendo automaticamente seus dados entre quatro camadas de acesso quando seus padrões de acesso mudam. Esses quatro níveis de acesso incluem dois níveis de acesso de baixa latência otimizados para acesso frequente e infrequente e dois níveis de acesso de arquivamento de inclusão projetados para acesso assíncrono para dados acessados raramente.

Para mais informações, consulte [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#). Para obter mais informações sobre o S3 Glacier Flexible Retrieval, consulte o [Guia do desenvolvedor do Amazon S3 Glacier](#).

Gerenciamento de armazenamento

O Amazon S3 tem recursos de gerenciamento de armazenamento que você pode usar para gerenciar custos, atender aos requisitos normativos, reduzir a latência e salvar várias cópias distintas de seus dados para requisitos de compatibilidade.

- [S3 Lifecycle](#): configura uma política de ciclo de vida para gerenciar seus objetos e armazená-los de maneira econômica durante todo o ciclo de vida. Você pode fazer a transição de objetos para outras classes de armazenamento do S3 ou expirar objetos que atingem o fim de suas vidas.
- [Bloqueio de objetos do S3](#): evita que os objetos do Amazon S3 sejam excluídos ou substituídos por um período de tempo fixo ou indefinidamente. Você pode usar o bloqueio de objetos para ajudar a

atender aos requisitos regulamentares que exigem armazenamento write-once-read-many (WORM) ou simplesmente adicionar outra camada de proteção contra alterações e exclusão de objetos.

- [Replicação do S3](#): replica objetos e seus respectivos metadados e etiquetas de objeto para um ou mais buckets de destino nas mesmas Regiões da AWS, ou diferentes, para reduzir a latência, compatibilidade, segurança e outros casos de uso.
- [Operações em lote do S3](#): gerencia bilhões de objetos em escala com uma única solicitação de API do S3 ou com alguns cliques no console do Amazon S3. Você pode usar operações em lote para executar operações como Copy (Copiar), Invoke AWS Lambda function (Chamar função Lambda da) e Restore (Restaurar) em milhões ou bilhões de objetos.

Gerenciamento de acesso

O Amazon S3 fornece recursos para auditoria e gerenciamento de acesso a seus buckets e objetos. Por padrão, os buckets do S3 e os objetos deles são privados. Você tem acesso somente aos recursos do S3 criados. Para conceder permissões de recursos detalhadas que suportam seu caso de uso específico ou para auditar as permissões de seus recursos do Amazon S3, você pode usar os seguintes recursos.

- [Bloqueio de acesso público do S3](#): bloqueie o acesso público a buckets e objetos do S3. Por padrão, as configurações de bloqueio de acesso público são ativadas no nível da conta e do bucket.
- [AWS Identity and Access Management \(IAM\)](#): crie usuários do IAM para sua Conta da AWS para gerenciar o acesso aos recursos do Amazon S3. Por exemplo, você pode usar o IAM com o Amazon S3 para controlar o tipo de acesso de um usuário ou grupo de usuários a um bucket do S3 pertencentes à sua Conta da AWS.
- [Políticas de buckets](#): use a linguagem de política baseada em IAM para configurar permissões baseadas em recursos para os buckets do S3 e os objetos neles contidos.
- [Pontos de acesso do Amazon S3](#): configure endpoints nomeados com políticas de acesso dedicadas para gerenciar o acesso a dados em escala para conjuntos de dados compartilhados no Amazon S3.
- [Listas de controle de acesso \(ACLs\)](#): conceda permissões de leitura e gravação para buckets e objetos individuais a usuários autorizados. Como regra geral, recomendamos o uso de políticas baseadas em recursos do S3 (políticas de bucket e políticas de ponto de acesso) ou políticas do IAM para controle de acesso, em vez de ACLs. ACLs são mecanismos de controle de acesso que antecedem políticas baseadas em recursos e IAM. Para obter mais informações sobre quando você usaria ACLs, em vez de políticas baseadas em recursos ou políticas do IAM, consulte [Diretrizes para políticas de acesso](#) (p. 425).
- [S3 Object Ownership](#): desabilite as ACLs e assuma a propriedade de cada objeto de seu bucket simplificando o gerenciamento de acesso para dados armazenados no Amazon S3. Como proprietário do bucket, você automaticamente tem a propriedade e o controle total sobre todos os objetos de seu bucket, e o controle de acesso a seus dados é baseado em políticas.
- [Analizador de acesso para S3](#): avalie e monitore suas políticas de acesso ao bucket do S3, garantindo que as políticas fornecem apenas o acesso pretendido aos seus recursos do S3.

Processamento de dados

Para transformar dados e acionar fluxos de trabalho para automatizar uma variedade de outras atividades de processamento em escala, você pode usar os seguintes recursos.

- [S3 Object Lambda](#): adiciona seu próprio código às solicitações GET do S3 para modificar e processar dados, conforme eles são retornados para uma aplicação. Filtra linhas, redimensiona imagens dinamicamente, edita dados confidenciais e muito mais.
- [Notificações de eventos](#): aciona fluxos de trabalho que usam o Amazon Simple Notification Service (Amazon SNS), o Amazon Simple Queue Service (Amazon SQS) e o AWS Lambda quando uma alteração for feita em seus recursos do S3.

Registro e monitoramento do armazenamento

O Amazon S3 fornece ferramentas de registro e monitoramento que você pode usar para monitorar e controlar como seus recursos do Amazon S3 estão sendo usados. Para obter mais informações, consulte [Ferramentas de monitoramento](#).

Ferramentas de monitoramento automatizadas

- [Métricas do Amazon CloudWatch para o Amazon S3](#): acompanha a integridade operacional de seus recursos do S3 e configura alertas de faturamento quando as cobranças estimadas atingirem um limite definido pelo usuário.
- [AWS CloudTrail](#): registra ações executadas por um usuário, uma função ou um AWS service (Serviço da AWS) no Amazon S3. Os logs do CloudTrail fornecem rastreamento detalhado de API para operações no nível de bucket e objeto do S3.

Ferramentas de monitoramento manual

- [Log de acesso ao servidor](#): fornece detalhes sobre as solicitações que são feitas a um bucket. Você pode usar logs de acesso ao servidor para muitos casos de uso, como conduzir auditorias de segurança e acesso, saber mais sobre sua base de clientes e entender sua fatura do Amazon S3.
- [AWSTrusted Advisor](#): avalia sua conta usando verificações de práticas recomendadas da AWS para identificar maneiras de otimizar sua infraestrutura da AWS, melhorar a segurança e a performance, reduzir custos e monitorar cotas de serviço. Em seguida, você pode seguir as recomendações para otimizar seus serviços e recursos.

Análise e insights

O Amazon S3 oferece recursos para ajudá-lo a obter visibilidade do uso do armazenamento, o que permite que você entenda melhor, analise e otimize seu armazenamento em escala.

- [Amazon S3 Storage Lens](#): entende, analisa e otimiza seu armazenamento. O S3 Storage Lens fornece mais de 29 métricas de uso e atividade e painéis interativos para agregar dados de toda a sua organização, contas específicas, Regiões da AWS, buckets ou prefixos.
- [Análise de classe de armazenamento](#): analisa padrões de acesso ao armazenamento para decidir quando é hora de mover seus dados para uma classe de armazenamento mais econômica.
- [Inventário do S3](#): audita e relata sobre objetos e seus metadados correspondentes e configura outros recursos do Amazon S3 para executar ações nos relatórios de inventário. Por exemplo, você pode gerar relatórios sobre o status da replicação e da criptografia de seus objetos. Para obter uma lista de todos os metadados disponíveis para cada objeto nos relatórios de inventário, consulte [Lista de inventário do Amazon S3 \(p. 787\)](#).

Consistência forte

O Amazon S3 oferece uma forte consistência de leitura após gravação para solicitações de PUT e DELETE de objetos no bucket do Amazon S3 em todas as Regiões da AWS. Isso se aplica a ambas as gravações em novos objetos, bem como solicitações PUT que sobrescrevem objetos existentes e solicitações DELETE. Além disso, as operações de leitura no Amazon S3 Select, listas de controle de acesso (ACLs) do Amazon S3, etiquetas de objeto do Amazon S3 e metadados de objeto (por exemplo, objeto HEAD) são fortemente consistentes. Para mais informações, consulte [Modelo de consistência de dados do Amazon S3 \(p. 7\)](#).

Como funciona o Amazon S3

O Amazon S3 é um serviço de armazenamento de objetos que armazena dados como objetos em buckets. Um objeto é um arquivo e quaisquer metadados que descrevam o arquivo. Um bucket é um contêiner de objetos.

Para armazenar seus dados no Amazon S3, crie um bucket e especifique um nome de bucket e a Região da AWS. Em seguida, carregue seus dados para esse bucket como objetos no Amazon S3. Cada objeto tem uma chave (ou nome de chave), que é um identificador exclusivo do objeto no bucket.

O S3 fornece recursos que você pode configurar para oferecer suporte ao seu caso de uso específico. Você pode usar o versionamento do S3 para manter várias versões de um objeto no mesmo bucket que permite que você restaure objetos excluídos ou substituídos acidentalmente.

Os buckets e os objetos neles são privados e poderão ser acessados somente se você conceder explicitamente permissões de acesso. Você pode usar políticas de bucket, políticas do AWS Identity and Access Management (IAM), listas de controle de acesso (ACLs) e pontos de acesso do S3 para gerenciar o acesso.

Tópicos

- [Buckets \(p. 4\)](#)
- [Objetos \(p. 5\)](#)
- [Chaves \(p. 5\)](#)
- [Versionamento do S3 \(p. 5\)](#)
- [ID da versão \(p. 5\)](#)
- [Política de bucket \(p. 5\)](#)
- [Pontos de acesso do S3 \(p. 6\)](#)
- [Listas de controle de acesso \(ACLs\) \(p. 6\)](#)
- [Regiões \(p. 6\)](#)

Buckets

Um bucket é um contêiner para objetos armazenados no Amazon S3. Você pode armazenar qualquer número de objetos em um bucket e pode ter até 100 buckets na sua conta. Para solicitar um aumento, visite o [Service Quotas Console](#) (Console de cotas de serviço).

Cada objeto está contido em um bucket. Por exemplo, se o objeto chamado `photos/puppy.jpg` estiver armazenado no bucket `DOC-EXAMPLE-BUCKET` da região oeste dos EUA (Oregon), ele poderá ser endereçado usando o URL `https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/photos/puppy.jpg`. Para obter mais informações, consulte [Accessing a Bucket \(p. 135\)](#) (Como acessar um bucket).

Ao criar um bucket, você insere um nome de bucket e escolhe a Região da AWS onde o bucket residirá. Assim que você cria um bucket, não pode mais alterar o nome do bucket ou sua região. Os nomes de bucket devem seguir as [regras de nomeação de bucket](#). Você também pode configurar um bucket para usar o [Versionamento do S3 \(p. 678\)](#) ou outros recursos do [gerenciamento de armazenamento](#).

Os buckets também:

- Organizam o namespace do Amazon S3 no nível mais elevado.
- Identificam a conta responsável por cobranças de transferência de dados e armazenamento.
- Fornecem opções de controle de acesso, como políticas de bucket, listas de controle de acesso (ACLs) e pontos de acesso do S3, que podem ser usados para gerenciar o acesso aos recursos do Amazon S3.
- Serve como a unidade de agregação para relatório de uso.

Para obter mais informações sobre buckets, consulte [Visão geral dos buckets \(p. 123\)](#).

Objetos

Os objetos são as entidades fundamentais armazenadas no Amazon S3. Os objetos consistem em metadados e dados de objeto. Os metadados são um conjunto de pares de nome e valor que descrevem o objeto. Esses pares incluem alguns metadados padrão, como a data da última modificação, e metadados HTTP padrão, como o 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 \(nome\) \(p. 5\)](#) e um [ID da versão \(p. 5\)](#) (se o Versionamento do S3 estiver habilitado no bucket). Para obter mais informações sobre objetos, consulte [Visão geral de objetos Amazon S3 \(p. 161\)](#).

Chaves

Uma chave de objeto (ou nome da chave) é um identificador exclusivo de um objeto em um bucket. Cada objeto em um bucket tem exatamente uma chave. A combinação de um bucket, chave de objeto e, opcionalmente, o ID de versão (se o Versionamento do S3 estiver habilitado para o bucket) identifica exclusivamente cada objeto. Portanto, é possível pensar no Amazon S3 como um mapa de dados básico entre "bucket + chave + versão" e o objeto em si.

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, no URL `https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/photos/puppy.jpg`, DOC-EXAMPLE-BUCKET é o nome do bucket e photos/puppy.jpg é a chave.

Para obter mais informações sobre chaves de objeto, consulte [Criar nomes de chave de objeto \(p. 162\)](#).

Versionamento do S3

Use o versionamento do S3 para manter diversas variantes de um objeto no mesmo bucket. Com o versionamento do S3, você pode preservar, recuperar e restaurar todas as versões de cada objeto armazenado em seus buckets. Você pode se recuperar facilmente de ações não intencionais do usuário e de falhas da aplicação.

Para mais informações, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

ID da versão

Se você habilitar o versionamento do S3 em um bucket, o Amazon S3 gerará um ID de versão exclusivo para cada objeto adicionado ao bucket. Os objetos que já existiam no bucket no momento em que você habilita o controle de versão têm um ID de versão null. Se você modificar esses (ou quaisquer outros) objetos com outras operações, como [CopyObject](#) e [PutObject](#), os novos objetos obtêm um ID de versão exclusivo.

Para mais informações, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

Política de bucket

Uma política de bucket é baseada em recursos do AWS Identity and Access Management (IAM) que você pode usar para conceder permissões de acesso ao bucket e aos objetos nele contidos. Só o proprietário do bucket pode associar uma política a um bucket. As permissões anexadas ao bucket se aplicam a todos os objetos do bucket que pertencem ao proprietário do bucket. As políticas de bucket são limitadas a 20 KB.

As políticas de bucket usam uma linguagem de políticas de acesso baseada em JSON que é padrão na AWS. Você pode usar políticas de bucket para adicionar ou negar permissões para os objetos em um bucket. As políticas de bucket permitem ou negam solicitações com base nos elementos da política, incluindo o solicitante, ações do S3, recursos e aspectos ou condições da solicitação (por exemplo, o endereço IP usado para fazer a solicitação). Por exemplo, você pode criar uma política de bucket que conceda permissões entre contas para carregar objetos em um bucket do S3 enquanto garante que o proprietário do bucket tenha controle total dos objetos carregados. Para mais informações, consulte [Exemplos de políticas de bucket \(p. 519\)](#).

Na política de bucket, você pode usar caracteres curinga nos nomes de recursos da Amazon (ARNs) e outros valores para conceder permissões a um subconjunto de objetos. Por exemplo, você pode controlar o acesso a grupos de objetos que começam com um [prefixo](#) ou termine com uma determinada extensão, como `.html`.

Pontos de acesso do S3

Os pontos de acesso do Amazon S3 são nomeados endpoints de rede com políticas de acesso dedicadas que descrevem como os dados podem ser acessados usando esse endpoint. Os pontos de acesso são anexados a buckets que você pode usar para executar operações de objeto do S3, como `GetObject` e `PutObject`. Os pontos de acesso simplificam o gerenciamento do acesso a dados em escala para conjuntos de dados compartilhados no Amazon S3.

Cada ponto de acesso tem sua própria política. Você também pode configurar definições de [Bloqueio de acesso público \(p. 618\)](#) para cada ponto de acesso. Você pode configurar qualquer ponto de acesso para aceitar solicitações somente de uma nuvem virtual privada (VPC) para restringir o acesso a dados do Amazon S3 a uma rede privada.

Para mais informações, consulte [Gerenciamento de acesso a dados com pontos de acesso do Amazon S3 \(p. 320\)](#).

Listas de controle de acesso (ACLs)

Você pode usar as ACLs para conceder permissões de leitura e gravação para buckets individuais e objetos a usuários autorizados. Cada bucket e objeto tem uma ACL anexada como um sub-recurso. Uma ACL define a quais grupos ou Contas da AWS é concedido acesso, bem como o tipo de acesso. As ACLs são mecanismos de controle de acesso que antecedem o IAM. Para obter mais informações sobre ACLs, consulte [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#).

Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. É possível usar Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Como resultado, o controle de acesso para seus dados é baseado em políticas, como políticas do IAM, políticas de bucket do S3, políticas de endpoint da Virtual Private Cloud (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations.

A maioria dos casos de uso modernos no Amazon S3 não requer mais o uso de ACLs, e recomendamos desabilitar as ACLs, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com Object Ownership, é possível desabilitar ACLs e confiar em políticas para controle de acesso. Ao desabilitar ACLs, você pode facilmente manter um bucket com objetos carregados por diferentes Contas da AWS. Como proprietário do bucket, você possui todos os objetos do bucket e pode gerenciar o acesso a eles usando políticas. Para mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Regiões

Você pode escolher a região da Região da AWS geográfica onde o Amazon S3 armazena os buckets criados. É possível escolher uma [região](#) para otimizar a latência, minimizar os custos ou atender a

requisitos regulatórios. Os objetos armazenados em uma Região da AWS nunca saem dela, a não ser que você os transfira ou os replique explicitamente para outra região. Por exemplo, os objetos armazenados na região da UE (Irlanda) nunca saem dela.

Note

Só é possível acessar o Amazon S3 e seus recursos em Regiões da AWS que estão habilitadas para sua conta. Para obter mais informações sobre como habilitar uma região para criar e gerenciar recursos da AWS, consulte [Como gerenciar Regiões da AWS](#) na Referência geral da AWS.

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

Modelo de consistência de dados do Amazon S3

O Amazon S3 oferece uma forte consistência de leitura após gravação para solicitações de PUT e DELETE de objetos no bucket do Amazon S3 em todas as Regiões da AWS. Isso se aplica a ambas as gravações em novos objetos, bem como solicitações PUT que sobrescrevem objetos existentes e solicitações DELETE. Além disso, as operações de leitura no Amazon S3 Select, listas de controle de acesso (ACLs) do Amazon S3, etiquetas de objeto do Amazon S3 e metadados de objeto (por exemplo, objeto HEAD) são muito consistentes.

As atualizações em uma única chave são atômicas. Por exemplo, se você executar uma solicitação PUT em uma chave existente de um thread e executar uma solicitação GET na mesma chave de um segundo thread simultaneamente, você obterá os dados antigos ou os novos dados, mas nunca dados parciais ou corrompidos.

O Amazon S3 atinge alta disponibilidade replicando dados entre vários servidores nos datacenters da AWS. Se uma solicitação PUT for bem-sucedida, os dados serão armazenados com segurança. Qualquer leitura (solicitação de GET ou LIST) iniciada após o recebimento de uma resposta PUT bem-sucedida retornará os dados escritos pelo PUT. Veja alguns exemplos desse comportamento:

- Um processo grava um novo objeto no Amazon S3 e imediatamente lista as chaves em seu bucket. O novo objeto aparecerá na lista.
- Um processo substitui um objeto existente e imediatamente tenta lê-lo. O Amazon S3 retorna os novos dados.
- Um processo exclui um objeto existente e imediatamente tenta lê-lo. O Amazon S3 não retorna dados porque o objeto foi excluído.
- Um processo exclui um objeto existente e imediatamente lista as chaves em seu bucket. O objeto não aparecerá na listagem.

Note

- O Amazon S3 não oferece suporte ao bloqueio de objetos para escritores simultâneos. 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 sua aplicação.
- As atualizações são baseadas em chave. 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.

As configurações de bucket têm um modelo de consistência eventual. Especificamente, isso significa que:

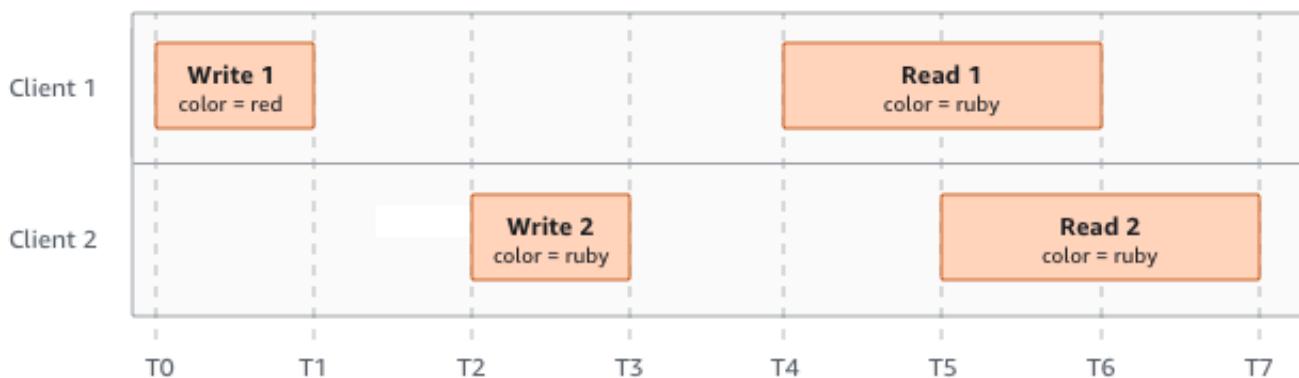
- Por exemplo, se você excluir um bucket e listar imediatamente todos os buckets, o bucket excluído ainda poderá ser exibido na lista.
- Se você ativar o versionamento em um bucket pela primeira vez, pode levar um curto período de tempo para que a alteração seja totalmente propagada. Recomendamos que você aguarde 15 minutos após ativar o versionamento antes de emitir operações de gravação (solicitações PUT ou DELETE) em objetos no bucket.

Aplicações simultâneas

Esta seção fornece exemplos de comportamento a serem esperados do Amazon S3 quando vários clientes estão gravando nos mesmos itens.

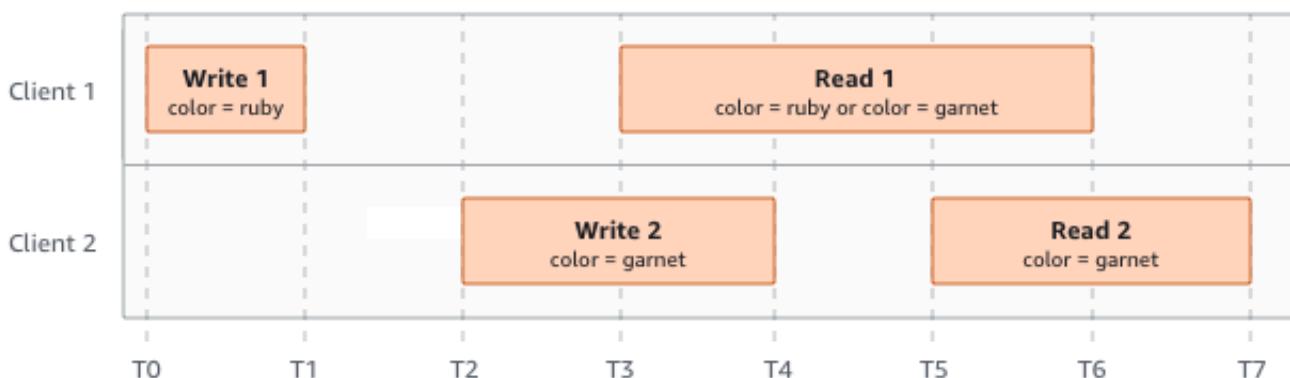
Neste exemplo, W1 (gravação 1) e W2 (gravação 2) são concluídas antes do início de R1 (leitura 1) e R2 (leitura 2). Como o S3 é fortemente consistente, R1 e R2 retornam `color = ruby`.

Domain = MyDomain, Item = StandardFez



No próximo exemplo, a W2 não é encerrada antes do início da R1. Portanto, R1 pode retornar `color = ruby` ou `color = garnet`. No entanto, como W1 e W2 terminam antes do início do R2, o R2 retorna `color = garnet`.

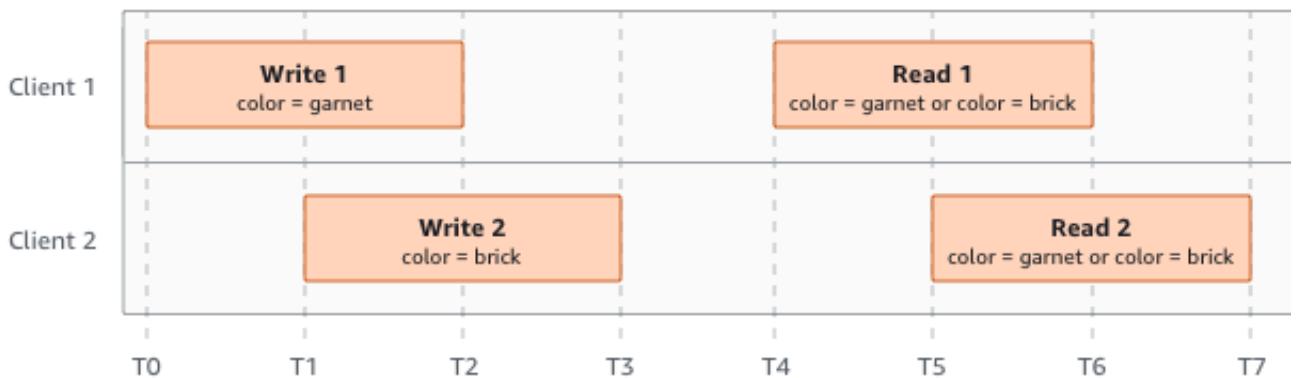
Domain = MyDomain, Item = StandardFez



No último exemplo, o W2 começa antes que o W1 tenha recebido um reconhecimento. Portanto, essas gravações são consideradas simultâneas. O Amazon S3 usa internamente a semântica do último escritor para determinar qual gravação tem precedência. No entanto, a ordem em que o Amazon S3 recebe as

solicitações e a ordem em que as aplicações recebem confirmações não podem ser previstas devido a fatores como latência de rede. Por exemplo, o W2 pode ser iniciado por uma instância do Amazon EC2 na mesma região, enquanto o W1 pode ser iniciado por um host que está mais longe. A melhor maneira de determinar o valor final é realizar uma leitura após ambas as gravações terem sido confirmadas.

Domain = MyDomain, Item = StandardFez



Serviços relacionados da

Depois de carregar os dados no Amazon S3, você poderá usá-los com outros serviços da AWS. Os serviços a seguir podem ser usados com mais frequência:

- O [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) oferece uma capacidade de computação escalável na Nuvem AWS. O uso do Amazon EC2 elimina a necessidade de investir em hardware inicialmente, portanto, você pode desenvolver e implantar aplicativos com mais rapidez. É possível usar o Amazon EC2 para executar quantos servidores virtuais forem necessários, configurar a segurança e as redes e gerenciar o armazenamento.
- [Amazon EMR](#): ajuda empresas, pesquisadores, analistas de dados e desenvolvedores a processar de maneira fácil e econômica grandes quantidades de dados. O Amazon EMR usa um framework do Hadoop hospedado que é executado na infraestrutura de escala da Web do Amazon EC2 e do Amazon S3.
- [Família do AWS Snow](#): ajuda os clientes que precisam executar operações em ambientes austeros e não datacenter e em locais onde há falta de conectividade de rede consistente. Você pode usar dispositivos da família do AWS Snow para acesso ao armazenamento localmente e de potência computacional da Nuvem AWS em lugares onde talvez não haja conexão à Internet.
- [AWS Transfer Family](#): fornece suporte totalmente gerenciado para transferências de arquivos diretamente para dentro e fora do Amazon S3 ou Amazon Elastic File System (Amazon EFS) usando Secure Shell (SSH), File Transfer Protocol (SFTP), File Transfer Protocol over SSL (FTPS) e File Transfer Protocol (FTP).

Acesso ao Amazon S3

Você pode trabalhar com o Amazon S3 de qualquer uma das seguintes formas:

AWS Management Console

O console é uma interface de usuário baseada na Web para gerenciar o Amazon S3 e os recursos da AWS. Se você se inscreveu em uma Conta da AWS, pode acessar o console do Amazon S3 fazendo login no AWS Management Console e escolhendo S3 na página inicial do AWS Management Console.

AWS Command Line Interface

Você pode usar as ferramentas de linha de comando da AWS para emitir comandos ou criar scripts na linha de comando de seu sistema e executar tarefas da AWS (incluindo o S3).

A [AWS Command Line Interface \(AWS CLI\)](#) fornece comandos para um amplo conjunto de Serviços da AWS. A AWS CLI é compatível com Windows, macOS e Linux. Para começar a usar, consulte o [Guia do usuário do AWS Command Line Interface](#). Para obter mais informações sobre os comandos do Amazon S3, consulte [s3api](#) e [s3control](#) na Referência de comandos da AWS CLI.

AWS SDKs

A AWS fornece SDKs (kits de desenvolvimento de software) que consistem em bibliotecas e códigos de exemplo para várias linguagens de programação e plataformas (Java, Python, Ruby, .NET, iOS, Android etc.). Os AWS SDKs constituem uma forma conveniente de criar acesso programático para o S3 e a AWS. O Amazon S3 é um serviço REST. Você pode enviar solicitações para o Amazon S3 usando a API REST ou as bibliotecas SDK da AWS que envolvem a API REST do Amazon S3, simplificando as tarefas de programação. Por exemplo, os SDKs processam tarefas como calcular assinaturas, assinar solicitações de forma criptográfica, gerenciar erros e novas tentativas automáticas de solicitações. Para obter informações sobre os AWS SDKs, incluindo como baixar e instalá-los, consulte [Ferramentas da AWS](#).

Cada interação com o Amazon S3 é autenticada ou anônima. Se você estiver usando os AWS SDKs, as bibliotecas calculam a assinatura das chaves fornecidas. Para obter mais informações sobre como fazer solicitações ao Amazon S3, consulte [Fazer solicitações \(p. 1211\)](#).

API REST do Amazon S3

A arquitetura do Amazon S3 foi desenvolvida para ser neutra em termos de linguagem de programação, usando nossas interfaces compatíveis com a AWS para armazenar e recuperar objetos. Você pode acessar o S3 e a AWS de forma programada usando a API REST do Amazon S3. A API REST é uma interface HTTP para o Amazon S3. Usando a API REST, você usa solicitações HTTP padrão para criar, buscar e excluir buckets 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.

Se fizer chamadas diretas da API REST na aplicação, você deverá escrever o código para calcular a assinatura e adicioná-la à solicitação. Para obter mais informações sobre como fazer solicitações ao Amazon S3, consulte [Fazer solicitações \(p. 1211\)](#).

Note

O suporte da API de SOAP via HTTP está defasado, mas continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com o SOAP. Recomendamos usar a API REST ou os AWS SDKs.

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 da aplicação. A maioria dos provedores de armazenamento exige que você adquira uma quantidade predeterminada de armazenamento e capacidade de transferência de rede. Nesse cenário, se você exceder essa capacidade, o serviço é desativado ou 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. Este modelo fornece a você um serviço com custo variável que pode aumentar com seus negócios enquanto proporciona a você as vantagens de custos de infraestrutura da AWS. Para obter mais informações, consulte [Preços do Amazon S3](#).

Ao cadastrar-se na AWS, sua Conta da AWS é automaticamente cadastrada em todos os serviços da AWS incluindo o Amazon S3. Entretanto, você será cobrado apenas pelos serviços que usar. Se você for um novo cliente do Amazon S3, você pode começar a usar o Amazon S3 gratuitamente. Para obter mais informações, consulte [nível gratuito da AWS](#).

Para ver sua fatura, acesse o Painel de Billing and Cost Management no [console da AWS Billing and Cost Management](#). Para saber mais sobre o faturamento da Conta da AWS, consulte o [Guia do usuário do AWS Billing](#). Se tiver dúvidas sobre o faturamento da AWS e as Contas da AWS, entre em contato com o [AWS Support](#).

Conformidade do PCI DSS

O Amazon S3 é compatível com o processamento, o armazenamento e a transmissão de dados de cartão de crédito por um comerciante ou um provedor de serviços e foi validada como em conformidade com o Data Security Standard (DSS, Padrão de segurança de dados) da Payment Card Industry (PCI, Padrão de cartão de crédito). Para obter mais informações sobre o PCI DSS, incluindo como solicitar uma cópia do pacote de conformidade com o PCI da AWS, consulte [Nível 1 do PCI DSS](#).

Conceitos básicos do Amazon S3

Você pode começar a usar o Amazon S3 trabalhando com buckets e objetos. Um bucket é um contêiner de objetos. Um objeto é um arquivo e qualquer metadado que descreva esse arquivo.

Para armazenar um objeto no Amazon S3, crie um bucket e faça upload do objeto para o bucket. Quando o objeto estiver no bucket, você poderá abri-lo, fazer download dele e movê-lo. Quando você não precisar mais de um objeto ou um bucket, poderá limpar seus recursos.

Com o Amazon S3, você paga somente pelo que for usado. Para obter mais informações sobre os recursos e a definição de preço do Amazon S3, consulte [Amazon S3](#). Se você for um novo cliente do Amazon S3, você pode começar a usar o Amazon S3 gratuitamente. Para obter mais informações, consulte [Nível gratuito da AWS](#).

Pré-requisitos

Antes de começar, confirme que concluiu as etapas em [Pré-requisito: Configuração do Amazon S3 \(p. 12\)](#).

Tópicos

- [Pré-requisito: Configuração do Amazon S3 \(p. 12\)](#)
- [Etapa 1: criar seu primeiro bucket do S3 \(p. 15\)](#)
- [Etapa 2: fazer upload de um objeto para o seu bucket \(p. 17\)](#)
- [Etapa 3: fazer download de um objeto \(p. 17\)](#)
- [Etapa 4: copiar seu objeto para uma pasta \(p. 18\)](#)
- [Etapa 5: excluir seus objetos e bucket \(p. 19\)](#)
- [Próximas etapas \(p. 20\)](#)
- [Práticas recomendadas de controle de acesso \(p. 24\)](#)

Pré-requisito: Configuração do Amazon S3

Ao cadastrar-se na AWS, sua Conta da AWS é automaticamente cadastrada em todos os serviços da AWS incluindo o Amazon S3. Você será cobrado apenas pelos serviços que usar.

Com o Amazon S3, você paga somente pelo que for usado. Para obter mais informações sobre os recursos e a definição de preço do Amazon S3, consulte [Amazon S3](#). Se você for um novo cliente do Amazon S3, você pode começar a usar o Amazon S3 gratuitamente. Para obter mais informações, consulte [Nível gratuito da AWS](#).

Para configurar o Amazon S3, use as etapas nas seções a seguir.

Quando você se inscreve na AWS e configura o Amazon S3, você pode, se quiser, alterar o idioma de exibição no AWS Management Console. Para obter mais informações, consulte [Alterar o idioma do AWS Management Console](#) no Guia de conceitos básicos do AWS Management Console.

Tópicos

- [Cadastro na AWS \(p. 13\)](#)

- Criar um usuário do IAM (p. 13)
- Fazer login como usuário do IAM (p. 14)

Cadastro na AWS

Se você ainda não tem uma Conta da AWS, siga as etapas a seguir para criar uma.

Para se cadastrar em uma Conta da AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga as instruções on-line.

Parte do procedimento de cadastro envolve uma chamada telefônica e a digitação de um código de verificação usando o teclado do telefone.

AWSA envia um e-mail de confirmação depois que o processo de cadastramento é concluído. A qualquer momento, é possível visualizar as atividades da conta atual e gerenciar sua conta acessando <https://aws.amazon.com/> e selecionando My Account (Minha conta).

Criar um usuário do IAM

Ao criar uma Conta da AWS, você começa com uma identidade de login que tenha acesso completo a todos os recursos e Serviços da AWS na conta. Essa identidade, denominada Conta da AWSusuário raiz da , é acessada por login com o endereço de e-mail e a senha que você usou para criar a conta.

Important

É altamente recomendável não usar o usuário raiz para tarefas diárias. Proteja as credenciais do usuário raiz e use-as para executar as tarefas que somente ele pode executar. Para obter a lista completa das tarefas que exigem login como usuário raiz, consulte [Tasks that require root user credentials](#) (Tarefas que exigem credenciais de usuário raiz) na Referência geral da AWS.

Se você se cadastrou na AWS, mas não criou um usuário do IAM próprio, siga estas etapas.

Para criar um usuário administrador para você mesmo e adicionar o usuário a um grupo de administradores (console)

1. Faça login no [console do IAM](#) como proprietário da conta escolhendo Root user (Usuário root) e inserindo o endereço de e-mail da sua Conta da AWS. Na próxima página, insira sua senha.

Note

Recomendamos seguir as práticas recomendadas para utilizar o usuário do IAM **Administrator** a seguir e armazenar as credenciais do usuário raiz com segurança. Cadastre-se como o usuário raiz apenas para executar algumas [tarefas de gerenciamento de serviços e contas](#).

2. No painel de navegação, escolha Users (Usuários) e Add users (Adicionar usuários).
3. Em User name (Nome do usuário), digite **Administrator**.
4. Marque a caixa de seleção ao lado do acesso ao AWS Management Console. Então, selecione Custom password (Senha personalizada), e insira sua nova senha na caixa de texto.
5. (Opcional) Por padrão, a AWS exige que o novo usuário crie uma senha ao fazer login pela primeira vez. Você pode desmarcar a caixa de seleção próxima de User must create a new password at next sign-in (O usuário deve criar uma senha no próximo login) para permitir que o novo usuário redefina a senha depois de fazer login.

6. Escolha Próximo: Permissões.
7. Em Set permissions (Conceder permissões), escolha Add user to group (Adicionar usuário ao grupo).
8. Escolha Create group (Criar grupo).
9. Na caixa de diálogo Create group (Criar grupo), em Group name (Nome do grupo), digite **Administrators**.
10. Escolha Filter policies (Filtrar políticas) e, depois, selecione AWS managed — job function (Função de trabalho gerenciada da AWS) para filtrar o conteúdo da tabela.
11. Na lista de políticas, marque a caixa de seleção AdministratorAccess. A seguir escolha Criar grupo.

Note

Você deve ativar o acesso de usuário do IAM e da função para Billing (Faturamento) antes de usar as permissões de AdministratorAccess para acessar o console do AWS Billing and Cost Management. Para fazer isso, siga as instruções na [etapa 1 do tutorial sobre como delegar acesso ao console de faturamento](#).

12. Superte a lista de grupos, selecione a caixa de seleção para seu novo grupo. Escolha Refresh (Atualizar) caso necessário, para ver o grupo na lista.
13. Escolha Next: Tags (Próximo: etiquetas).
14. (Opcional) Adicione metadados ao usuário anexando tags como pares de chave-valor. Para obter mais informações sobre como usar etiquetas no IAM, consulte [Marcar entidades do IAM](#) no Guia do usuário do IAM.
15. Escolha Next: Review (Próximo: Análise) para ver uma lista de associações de grupos a serem adicionadas ao novo usuário. Quando você estiver pronto para continuar, escolha Create user (Criar usuário).

Você pode usar esse mesmo processo para criar mais grupos e usuários e conceder aos seus usuários acesso aos recursos da sua Conta da AWS. Para saber como usar políticas para restringir as permissões de usuário a recursos específicos da AWS, consulte [Gerenciamento de acesso](#) e [Exemplos de políticas](#).

Fazer login como usuário do IAM

Depois de criar um usuário do IAM, você pode fazer login na AWS com seu nome de usuário e senha do IAM.

Antes de fazer login como usuário do IAM, você pode verificar o link de login para usuários do IAM no console do IAM. No painel do IAM, no IAM users sign-in link (link de login de usuários do IAM), você pode ver o link de login da sua Conta da AWS. O URL do link de login contém o ID da sua Conta da AWS sem traços (-).

Se você não quiser que o URL do link de login contenha o ID da sua Conta da AWS, crie um alias da conta. Para obter mais informações, consulte [Criação, exclusão e listagem de um alias de Conta da AWS](#) no Manual do usuário do IAM.

Faça login como usuário da AWS.

1. Saia do AWS Management Console.
2. Insira link de login.

Seu link de login inclui seu ID de Conta da AWS (sem traços) ou seu alias de Conta da AWS:

`https://aws_account_id_or_alias.signin.aws.amazon.com/console`

3. Insira o nome e a senha de usuário do IAM que você acabou de criar.

Quando você está conectado, a barra de navegação exibe "your_user_name @ your_aws_account_id".

Etapa 1: criar seu primeiro bucket do S3

Depois de se inscrever na AWS, você estará pronto para criar um bucket no Amazon S3 usando o AWS Management Console. Cada objeto no Amazon S3 é armazenado em um bucket. Antes de poder armazenar dados no Amazon S3, você deve criar um bucket.

Note

Você não é cobrado pela criação de um bucket. Você é cobrado somente pelo armazenamento de objetos no bucket e pela transferência de objetos para dentro e para fora do bucket. Você estará sujeito a uma cobrança mínima (menos de 1 USD) ao seguir os exemplos contidos neste guia. Para obter mais informações sobre os custos de armazenamento, consulte [Definição de preço do Amazon S3](#).

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Selecione Create bucket (Criar bucket).

O assistente Create bucket (Criar bucket) é aberto.

3. Em Bucket name (Nome do bucket), insira um nome compatível com o DNS para seu bucket.

O nome do bucket deve:

- Seja exclusivo em todo o Amazon S3.
- Ter entre 3 e 63 caracteres.
- Não contém caracteres maiúsculos.
- Começar com uma letra minúscula ou um número.

Depois de criado o bucket, não é possível mudar o nome. Para obter informações sobre nomeação de buckets, consulte [Regras de nomeação de bucket \(p. 127\)](#).

Important

Evite incluir informações sigilosas, como números de conta, no nome do bucket. O nome do bucket é visível nos URLs que apontam para os objetos no bucket.

4. Em Region (Região), escolha a Região da AWS onde deseja que o bucket resida.

Escolha uma região próxima de você para minimizar a latência e os custos e atender aos requisitos regulatórios. Os objetos armazenados em uma região nunca saem dessa região, a menos que você os transfira para outra região. Para obter uma lista das Regiões da AWS do Amazon S3, consulte [Endpoints de serviço da AWS](#) na Referência geral da Amazon Web Services.

5. Em Object Ownership, para desabilitar ou habilitar ACLs e controlar a propriedade de objetos carregados em seu bucket, escolha uma das seguintes configurações:

ACLs desabilitadas

- Bucket owner enforced (Aplicada pelo proprietário do bucket) (recomendado): as ACLs são desabilitadas e o proprietário do bucket automaticamente tem a propriedade e o controle total sobre todos os objetos do bucket. As ACLs não afetam mais as permissões nos dados no bucket do S3. O bucket usa políticas para definir o controle de acesso.

Para exigir que todos os novos buckets sejam criados com ACLs desabilitadas usando IAM ou políticas do AWS Organizations, consulte [Desabilitar ACLs para todos os novos buckets \(aplicada pelo proprietário do bucket\) \(p. 660\)](#).

ACLs habilitadas

- Bucket owner preferred (Preferencial do proprietário do bucket): o proprietário do bucket tem a propriedade e o controle total sobre novos objetos que outras contas gravam no bucket com a ACL `bucket-owner-full-control` pré-configurada.

Se você aplicar a configuração preferencial do proprietário do bucket para exigir que todos os carregamentos do Amazon S3 incluam a ACL `bucket-owner-full-control` pré-configurada, será possível [adicionar uma política de bucket \(p. 661\)](#) que só permita carregamentos de objetos que usam essa ACL.

- Object writer (Gravador de objetos): a Conta da AWS que carrega um objeto é a proprietária do objeto, tem controle total sobre ele e pode conceder acesso a outros usuários por meio de ACLs.

Note

Para aplicar a configuração Bucket owner enforced (Aplicada pelo proprietário do bucket) ou Bucket owner preferred (Preferencial do proprietário do bucket), é necessário ter as seguintes permissões: `s3:CreateBucket` e `s3:PutBucketOwnershipControls`.

- Em Bucket settings for Block Public Access (Configurações de bucket para o Bloqueio de acesso público), escolha as configurações de bloqueio de acesso público que deseja aplicar ao bucket.

Recomendamos que você mantenha todas as configurações ativadas, a menos que saiba que precisa desativar uma ou mais delas para seu caso de uso, como para hospedar um site público. As configurações de bloqueio de acesso público que você habilitar para o bucket também serão ativadas para todos os pontos de acesso criados no bucket. Para obter mais informações sobre como bloquear o acesso público, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

- (Opcional) Se você quiser habilitar o Bloqueio de Objetos do S3, faça o seguinte:

- Selecione Advanced settings (Configurações avançadas).

Important

Você só pode habilitar o bloqueio de objetos do S3 para um bucket ao criá-lo. Se você habilitar o bloqueio de objeto para o bucket, não poderá desabilitá-lo mais tarde. A ativação do bloqueio de objetos também permite o versionamento para o bucket. Após habilitar o bloqueio de objeto para o bucket, é necessário definir as configurações padrão de retenção e de retenção legal do bloqueio de objeto para proteger novos objetos de serem excluídos ou substituídos. Para mais informações, consulte [Configurar o bloqueio de objetos do S3 usando o console \(p. 727\)](#).

- Se quiser ativar o bloqueio de objetos, escolha **Enable**, leia o aviso que aparece e confirme-o.

Para obter mais informações sobre o recurso bloqueio de objeto do S3, consulte [Usar o bloqueio de objetos do S3 \(p. 722\)](#).

Note

Para criar um bucket habilitado para bloqueio de objetos, você deve ter as seguintes permissões: `S3:CreateBucket`, `S3:PutBucketVersioning` e `S3:PutBucketObjectLockConfiguration`.

- Selezione Create bucket (Criar bucket).

Você criou um bucket no Amazon S3.

Próxima etapa

Para adicionar um objeto ao bucket, consulte [Etapa 2: fazer upload de um objeto para o seu bucket \(p. 17\)](#).

Etapa 2: fazer upload de um objeto para o seu bucket

Depois de criar um bucket no Amazon S3, você estará pronto para fazer upload de um objeto no bucket. Um objeto pode ser qualquer tipo de arquivo: um arquivo de texto, uma foto, um vídeo etc.

Para fazer upload de um objeto em um bucket

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket no qual você deseja fazer upload do objeto.
3. Na guia Objects (Objetos) do bucket, escolha Upload.
4. Em Files and folders (Arquivos e pastas), escolha Add files (Adicionar arquivos).
5. Escolha um arquivo para carregar e, em seguida, escolha Open (Abrir).
6. Escolha Upload (Fazer upload).

O upload de um objeto no bucket foi realizado corretamente.

Próxima etapa

Para visualizar o objeto, consulte [Etapa 3: fazer download de um objeto \(p. 17\)](#).

Etapa 3: fazer download de um objeto

Depois de fazer upload de um objeto para um bucket, você já pode visualizar informações e fazer download do objeto em seu computador local.

Uso do console do S3

Esta seção explica como usar o console do Amazon S3 para fazer download de um objeto contido em um bucket do S3 usando um URL pré-assinado.

Note

- Apenas um objeto pode ser obtido por download de cada vez.
- Objetos com nomes de chave terminando com pontos “.” e baixados usando o console do Amazon S3 terão os pontos “.” removidos do nome da chave do objeto baixado. Para baixar um objeto com o nome da chave terminado com ponto “.” mantido no objeto baixado, será necessário usar a AWS Command Line Interface (AWS CLI), AWS SDKs ou a API REST. Para obter exemplos e informações sobre a AWS CLI, a API REST e o AWS SDK, consulte [Baixar um objeto](#).

Fazer download de um objeto de um bucket do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket do qual você deseja fazer download de um objeto.
3. Você pode baixar um objeto de um bucket do S3 de uma das seguintes formas:
 - No menu Actions (Ações), selecione o objeto e escolha Download (Baixar) ou Download as (Baixar como) se quiser baixar o objeto para uma pasta específica.
 - Se você quiser baixar uma versão específica do objeto, selecione o botão Show versions (Mostrar versões). No menu Actions (Ações), selecione a versão do objeto desejado e escolha Download (Baixar) ou Download as (Baixar como) se quiser baixar o objeto para uma pasta específica.

Você baixou seu objeto com êxito.

Próxima etapa

Para copiar e colar seu objeto no Amazon S3, consulte [Etapa 4: copiar seu objeto para uma pasta \(p. 18\)](#).

Etapa 4: copiar seu objeto para uma pasta

Você já adicionou um objeto a um bucket e fez download do objeto. Agora, você cria uma pasta e copia o objeto e o cola na pasta.

Como copiar um objeto em uma pasta

1. Na lista Buckets, escolha o nome do bucket.
2. Selecione Create folder (Criar pasta) e configure uma nova pasta:
 - a. Insira um nome para a pasta (por exemplo, `favorite-pics`).
 - b. Para a configuração de criptografia de pasta, escolha Disable (Desabilitar).
 - c. Escolha Save (Salvar).
3. Navegue até o bucket ou pasta do Amazon S3 que contém os objetos que você deseja copiar.
4. Marque a caixa de seleção à esquerda dos nomes dos objetos que você deseja copiar.
5. Escolha Actions (Ações) e escolha Copy (Copiar) na lista de opções exibida.

Como alternativa, escolha Copy (Copiar) nas opções no canto superior direito.

6. Escolha a pasta de destino:
 - a. Escolha Browse S3 (Navegar no S3).
 - b. Escolha o botão de opção à esquerda do nome da pasta.

Para navegar em uma pasta e escolher uma subpasta como seu destino, escolha o nome da pasta.

- c. Escolha Choose destination (Escolher destino).

O caminho para a pasta de destino aparece na caixa Destination (Destino) . Em Destination (Destino), você pode, também, inserir o caminho de destino, por exemplo, `s3://bucket-name/folder-name/`.

7. No canto inferior direito, escolha Copy (Copiar).

O Amazon S3 move seus objetos para a pasta de destino.

Próxima etapa

Para excluir um objeto e um bucket no Amazon S3, consulte [Etapa 5: excluir seus objetos e bucket \(p. 19\)](#).

Etapa 5: excluir seus objetos e bucket

Quando você não precisar mais de um objeto ou bucket, recomendamos excluí-los para evitar cobranças adicionais. Se você concluiu esta demonstração de conceitos básicos como um exercício de aprendizagem e não planeja usar o bucket ou os objetos, recomendamos que exclua os dois para não acumular cobranças.

Antes de excluir seu bucket, esvazie-o ou exclua os objetos contidos nele. Depois de excluir seus objetos e o bucket, eles não estarão mais disponíveis.

Se você quiser continuar usando o mesmo nome de bucket, recomendamos excluir os objetos ou esvaziar o bucket, mas não excluir o bucket. Depois de excluir um bucket, o nome dele fica disponível para ser reutilizado. No entanto, outra Conta da AWS pode criar um bucket com o mesmo nome antes de você ter a chance de reutilizá-lo.

Tópicos

- [Excluir um objeto \(p. 19\)](#)
- [Esvaziar o bucket \(p. 19\)](#)
- [Excluir bucket \(p. 20\)](#)

Excluir um objeto

Se quiser escolher quais objetos excluir sem esvaziar todos os objetos do bucket, você pode excluir um objeto.

1. Na lista Buckets, escolha o nome do bucket do qual deseja excluir um objeto.
2. Marque a caixa de seleção à esquerda dos nomes dos objetos que você deseja excluir.
3. Escolha Actions (Ações) e escolha Delete (Excluir) na lista de opções exibida.
Como alternativa, escolha Delete (Excluir) nas opções no canto superior direito.
4. Digite **permanently delete** se receber a solicitação para confirmar que deseja excluir esses objetos.
5. Escolha Delete objects (Excluir objetos) no canto inferior direito e o Amazon S3 excluirá os objetos especificados.

Esvaziar o bucket

Se pretender excluir seu bucket, primeiro você deve esvaziar seu bucket, o que exclui todos os objetos contidos nele.

Para esvaziar um bucket

1. Na lista Buckets, selecione o bucket que deseja esvaziar e escolha Empty (Vazio).

2. Para confirmar que deseja esvaziar o bucket e excluir todos os objetos contidos nele, em Empty bucket (Esvaziar bucket), digite **permanently delete**.

Important

Não é possível desfazer a ação de esvaziar bucket. Os objetos adicionados ao bucket enquanto a ação de esvaziamento do bucket estiver em andamento serão excluídos.

3. Para esvaziar o bucket e excluir todos os objetos contidos nele, escolha Empty (Esvaziar).

Uma página Empty bucket :Status (Esvaziar bucket: status) é aberta, para que você possa revisar um resumo de exclusões de objetos com falha e bem-sucedidas.

4. Para retornar à sua lista de buckets, escolha Exit (Sair).

Excluir bucket

Depois de esvaziar o bucket ou excluir todos os objetos dele, você poderá excluir o bucket.

1. Para excluir um bucket, selecione o bucket na lista Buckets.
2. Escolha Delete.
3. Para confirmar a exclusão, digite o nome do bucket em Delete bucket (Excluir bucket).

Important

Não é possível desfazer a ação de excluir um bucket. Nomes de bucket são exclusivos. Se você excluir seu bucket, outro usuário da AWS poderá usar o nome. Se quiser continuar usando o mesmo nome de bucket, não exclua o bucket. Em vez disso, esvazie e conserve o bucket.

4. Para excluir seu bucket, escolha Delete bucket (Excluir bucket).

Próximas etapas

Nos exemplos anteriores, você aprendeu a executar alguma tarefas básicas no Amazon S3.

Os seguintes tópicos explicam os caminhos de aprendizado que você pode usar para conhecer o Amazon S3 detalhadamente para que você possa implementá-lo em suas aplicações.

Tópicos

- [Entender casos de uso comuns \(p. 20\)](#)
- [Controlar o acesso a buckets e objetos \(p. 21\)](#)
- [Explore o treinamento e o suporte \(p. 21\)](#)
- [Gerencie e monitore seu armazenamento \(p. 22\)](#)
- [Desenvolvimento com o Amazon S3 \(p. 22\)](#)

Entender casos de uso comuns

Você pode usar o Amazon S3 para oferecer suporte ao seu caso de uso específico. A [Biblioteca de soluções da AWS](#) e o [Blog da AWS](#) fornecem informações e tutoriais específicos para casos de uso. Veja a seguir alguns casos de uso comuns do Amazon S3:

- Backup e armazenamento: use os recursos de gerenciamento de armazenamento do Amazon S3 para gerenciar custos, atender aos requisitos normativos, reduzir a latência e salvar várias cópias distintas de seus dados para requisitos de conformidade.

- Hospedagem de aplicações: implante, instale e gerencie aplicações Web confiáveis, altamente escaláveis e de baixo custo. Por exemplo, você pode configurar seu bucket do Amazon S3 para hospedar um site estático. Para obter mais informações, consulte [Hospedagem de um site estático usando o Amazon S3 \(p. 1188\)](#).
- Hospedagem de mídia: crie uma infraestrutura altamente disponível que hospede carregamentos e downloads de vídeos, fotos ou músicas.
- Entrega de software: hospede suas aplicações de software para que sejam baixadas pelos clientes.

Controlar o acesso a buckets e objetos

O Amazon S3 fornece uma variedade de recursos e ferramentas de segurança. Para obter uma visão geral, consulte [Práticas recomendadas de controle de acesso \(p. 24\)](#).

Por padrão, os buckets do S3 e os objetos deles são privados. Você tem acesso somente aos recursos do S3 criados. Você pode usar os seguintes recursos para conceder permissões de recursos detalhadas que ofereçam suporte a seu caso de uso específico ou para auditar as permissões de seus recursos do Amazon S3.

- [Bloqueio de acesso público do S3](#): bloqueie o acesso público a buckets e objetos do S3. Por padrão, as configurações de bloqueio de acesso público são ativadas no nível da conta e do bucket.
- [AWS Identity and Access Management \(IAM\)](#): crie usuários do IAM para sua Conta da AWS para gerenciar o acesso aos recursos do Amazon S3. Por exemplo, você pode usar o IAM com o Amazon S3 para controlar o tipo de acesso de um usuário ou grupo de usuários a um bucket do Amazon S3 pertencentes à sua Conta da AWS.
- [Políticas de buckets](#): use a linguagem de política baseada em IAM para configurar permissões baseadas em recursos para os buckets do S3 e os objetos neles contidos.
- [Listas de controle de acesso \(ACLs\)](#): conceda permissões de leitura e gravação para buckets e objetos individuais a usuários autorizados. Como regra geral, recomendamos o uso de políticas baseadas em recursos do S3 (políticas de bucket e políticas de ponto de acesso) ou políticas do IAM para controle de acesso, em vez de ACLs. ACLs são mecanismos de controle de acesso que antecedem políticas baseadas em recursos e IAM. Para obter mais informações sobre quando você usaria ACLs, em vez de políticas baseadas em recursos ou políticas do IAM, consulte [Diretrizes para políticas de acesso \(p. 425\)](#).
- [S3 Object Ownership](#): desabilite as ACLs e assuma a propriedade de cada objeto de seu bucket simplificando o gerenciamento de acesso para dados armazenados no Amazon S3. Como proprietário do bucket, você automaticamente tem a propriedade e o controle total sobre todos os objetos de seu bucket, e o controle de acesso a seus dados é baseado em políticas.
- [Analizador de acesso para S3](#): avalie e monitore suas políticas de acesso ao bucket do S3, garantindo que as políticas fornecem apenas o acesso pretendido aos seus recursos do S3.

Explore o treinamento e o suporte

Você pode aprender com especialistas da AWS para aprimorar suas habilidades e obter assistência especializada para alcançar seus objetivos.

- Treinamento: os recursos de treinamento oferecem uma abordagem prática para o aprendizado do Amazon S3. Para obter mais informações, consulte [Treinamentos e certificações da AWS](#) e as [AWS Online Tech Talks](#).
- Fóruns de discussão: no fórum, você pode analisar as publicações para entender o que pode e o que não pode ser feito com o Amazon S3. Você também pode publicar suas dúvidas. Para obter mais informações, consulte os [Fóruns de discussão](#).
- Suporte técnico: se você tiver mais dúvidas, entre em contato com o [Suporte técnico](#).

Gerencie e monitore seu armazenamento

- [Como gerenciar seu armazenamento \(p. 678\)](#): depois de criar buckets e carregar objetos no Amazon S3, você pode gerenciar seu armazenamento de objetos. Por exemplo, você pode usar o versionamento do S3 e a replicação do Amazon S3 para recuperação de desastres, o S3 Lifecycle para gerenciar custos de armazenamento e o bloqueio de objetos do S3 para atender aos requisitos de conformidade.
- [Monitoramento de seu armazenamento \(p. 1020\)](#): o monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e a performance do Amazon S3 e das soluções da AWS. Você pode monitorar a atividade e os custos de armazenamento. Além disso, recomendamos que você colete dados de monitoramento de todas as partes de sua solução da AWS para que possa depurar mais facilmente uma falha multiponto, caso ocorra.
- [Análises e insights \(p. 1116\)](#): você pode usar análises e insights no Amazon S3 para entender, analisar e otimizar o uso do armazenamento. Por exemplo, use o [Amazon S3 Storage Lens \(p. 1121\)](#) para entender, analisar e otimizar seu armazenamento. O S3 Storage Lens fornece mais de 29 métricas de uso e atividade e painéis interativos para agragar dados de toda a sua organização, contas específicas, regiões, buckets ou prefixos. Use a [Análise de classe de armazenamento \(p. 1116\)](#) para analisar padrões de acesso ao armazenamento para decidir quando é hora de mover seus dados para uma classe de armazenamento mais econômica.

Desenvolvimento com o Amazon S3

O Amazon S3 é um serviço REST. Você pode enviar solicitações para o Amazon S3 usando a API REST ou as bibliotecas do SDK da AWS que envolvem a API REST subjacente do Amazon S3, simplificando as tarefas de programação. Você também pode usar o AWS Command Line Interface (AWS CLI) para fazer chamadas de API do Amazon S3. Para obter mais informações, consulte [. Fazer solicitações \(p. 1211\)](#).

A API REST do Amazon S3 é uma interface HTTP para o Amazon S3. Usando a API REST, você usa solicitações HTTP padrão para criar, buscar e excluir buckets 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. Para obter mais informações, consulte [. Desenvolver com o Amazon S3 usando a API REST \(p. 1272\)](#).

Para ajudá-lo a criar aplicações usando a linguagem de sua escolha, fornecemos os seguintes recursos.

AWS CLI

Você pode acessar os recursos do Amazon S3 usando a AWS CLI. Para baixar e configurar a AWS CLI, consulte [Desenvolvimento com o Amazon S3 usando a AWS CLI \(p. 1257\)](#).

A AWS CLI oferece dois níveis de comandos para acessar o Amazon S3: comandos de alto nível ([s3](#)) e no nível de API ([s3api](#) e [s3control](#)). Os comandos de alto nível do S3 simplificam a execução de tarefas comuns, como criar, manipular e excluir objetos e buckets. Os comandos s3api e s3control expõem acesso direto a todas as operações de API do Amazon S3, que podem ser usadas para executar operações avançadas que podem não ser possíveis com os comandos de alto nível.

Para obter uma lista dos comandos da AWS CLI do Amazon S3, consulte [s3](#), [s3api](#) e [s3control](#).

AWSSDKs e Explorers da

Você pode usar os AWS SDKs ao desenvolver aplicações com o Amazon S3. Os AWS SDKs simplificam as tarefas de programação integrando a API REST subjacente. Os SDKs móveis da AWS e a biblioteca JavaScript do Amplify também estão disponíveis para a compilação de aplicações Web e aplicações para dispositivos móveis conectados usando a AWS.

Além dos SDKs da AWS, os Explorers da AWS estão disponíveis para Visual Studio e Eclipse para Java IDE. Nesse caso, os SDKs e os explorers estão empacotados como toolkits da AWS.

Para obter mais informações, consulte . [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers \(p. 1258\)](#).

Bibliotecas e código de exemplo

O [Centro do Desenvolvedor da AWS](#) e o [Catálogo de exemplos de código da AWS](#) contêm código de exemplo e bibliotecas escritos especialmente para o Amazon S3. Você pode usar esses exemplos de códigos para entender como implementar a API do Amazon S3. Você também pode visualizar a [Referência da API de Serviço do Amazon Simple Storage](#) para entender as operações da API do Amazon S3 em detalhes.

Práticas recomendadas de controle de acesso

O Amazon S3 fornece uma variedade de recursos e ferramentas de segurança. Os cenários a seguir devem servir como um guia para quais ferramentas e configurações você pode querer usar ao executar determinadas tarefas ou operar em ambientes específicos. A aplicação adequada dessas ferramentas pode ajudar a manter a integridade dos dados e ajudar a garantir que os recursos sejam acessíveis aos usuários pretendidos.

Tópicos

- [Criar um bucket \(p. 24\)](#)
- [Armazenar e compartilhar dados \(p. 26\)](#)
- [Compartilhar recursos \(p. 27\)](#)
- [Proteger dados \(p. 27\)](#)

Criar um bucket

Ao criar um bucket, é necessário aplicar as seguintes ferramentas e configurações para ajudar a garantir que os recursos do Amazon S3 estejam protegidos.

S3 Object Ownership para simplificar o controle de acesso

O S3 Object Ownership é uma configuração no nível do bucket do Amazon S3 que você pode usar para desabilitar as listas de controle de acesso (ACLs) e assumir a propriedade de cada objeto em seu bucket, simplificando o gerenciamento de acesso para dados armazenados no Amazon S3. Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. É possível usar Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Como resultado, o controle de acesso para seus dados é baseado em políticas, como políticas do IAM, políticas de bucket do S3, políticas de endpoint da Virtual Private Cloud (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations.

A Propriedade de objetos conta com três configurações que você pode usar para controlar a propriedade de objetos carregados no bucket e desabilitar ou habilitar ACLs:

ACLs desabilitadas

- Bucket owner enforced (Aplicada pelo proprietário do bucket) (recomendado): as ACLs são desabilitadas e o proprietário do bucket automaticamente tem a propriedade e o controle total sobre todos os objetos do bucket. As ACLs não afetam mais as permissões nos dados no bucket do S3. O bucket usa políticas exclusivamente para definir o controle de acesso.

ACLs habilitadas

- Bucket owner preferred (Preferencial do proprietário do bucket): o proprietário do bucket tem a propriedade e o controle total sobre novos objetos que outras contas gravam no bucket com a ACL `bucket-owner-full-control` pré-configurada.
- Object writer (Gravador de objetos) (padrão): a Conta da AWS que carrega um objeto é a proprietária do objeto, tem controle total sobre ele e pode conceder acesso a outros usuários por meio de ACLs.

Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Bloqueio de acesso público

O Bloqueio de acesso público do S3 fornece quatro configurações para ajudar a evitar expor inadvertidamente seus recursos do S3. É possível aplicar essas configurações em qualquer combinação a pontos de acesso individuais, buckets ou Contas da AWS inteiras. Caso você aplique uma configuração a uma conta, ela se aplica a todos os buckets e pontos de acesso de propriedade dessa conta. Por padrão, a configuração Block all public access (Bloquear todo o acesso público) é aplicada a novos buckets criados no console do Amazon S3.

Para obter mais informações, consulte [O significado de "público" \(p. 622\)](#).

Se as configurações do S3 Block Public Access forem muito restritivas, você poderá usar identidades do AWS Identity and Access Management (IAM) para conceder acesso a usuários específicos em vez de desabilitar todas as configurações do Block Public Access. Usar o Bloqueio de acesso público com identidades do IAM ajuda a garantir que qualquer operação bloqueada por uma configuração do Bloqueio de acesso público seja rejeitada, a menos que o usuário solicitante tenha recebido permissão específica.

Para obter mais informações, consulte [Configurações do bloqueio de acesso público \(p. 620\)](#).

Conceder acesso com identidades do IAM

Ao configurar contas para novos membros da equipe que exigem acesso ao S3, use usuários e funções do IAM para garantir privilégios mínimos. Também é possível implementar uma forma de autenticação multifator (MFA) do IAM para apoiar uma base de identidade sólida. Com as identidades do IAM, é possível conceder permissões exclusivas aos usuários e especificar quais recursos eles podem acessar e quais ações eles podem executar. As identidades do IAM fornecem mais recursos, incluindo a capacidade de exigir que os usuários insiram credenciais de login antes de acessar recursos compartilhados e aplicar hierarquias de permissão a diferentes objetos em um único bucket.

Para obter mais informações, consulte [Exemplo 1: Proprietário do bucket que concede permissões do bucket aos usuários \(p. 557\)](#).

Políticas de buckets

Com as políticas de bucket, é possível personalizar o acesso ao bucket para ajudar a garantir que somente os usuários aprovados possam acessar recursos e executar ações neles. Além das políticas de bucket, você deve usar as configurações do Bloqueio de acesso público no nível do bucket para limitar ainda mais o acesso público aos dados.

Para obter mais informações, consulte [Uso de políticas de bucket \(p. 515\)](#).

Ao criar políticas, evite o uso de caracteres curinga no elemento `Principal` porque ele permite efetivamente que qualquer pessoa acesse os recursos do Amazon S3. É melhor listar explicitamente usuários ou grupos que têm permissão para acessar o bucket. Em vez de incluir um curinga para as ações, conceda permissões específicas quando aplicável.

Para aprimorar a prática de privilégios mínimos, as declarações `Deny` no elemento `Effect` devem ser o mais amplas possíveis e as declarações `Allow` devem ser tão restritas quanto possível. O efeito de `Deny` combinado com a ação “`s3:*`” é outra boa maneira de implementar as práticas recomendadas de adesão para os usuários incluídos em declarações de condição de política.

Para obter mais informações sobre como especificar condições para quando uma política estiver em vigor, consulte [Exemplos de chave de condição do Amazon S3 \(p. 447\)](#).

Buckets em uma configuração de VPC

Ao adicionar usuários em uma configuração corporativa, você poderá usar um endpoint de Virtual Private Cloud (VPC) para permitir que todos os usuários na rede virtual acessem os recursos do Amazon S3. Os VPC endpoints permitem que os desenvolvedores concedam acesso e permissões específicos a grupos de usuários com base na rede à qual o usuário está conectado. Em vez de adicionar cada usuário a uma função ou a um grupo do IAM, você poderá usar VPC endpoints para negar acesso ao bucket se a solicitação não for originada do endpoint especificado.

Para obter mais informações, consulte [Controlar o acesso a partir de VPC endpoints com políticas de bucket \(p. 517\)](#).

Armazenar e compartilhar dados

Use as seguintes ferramentas e melhores práticas para armazenar e compartilhar os dados do Amazon S3.

Versionamento e bloqueio de objetos para integridade de dados

Se você usar o console do Amazon S3 para gerenciar buckets e objetos, você deve implementar o versionamento do S3 e o bloqueio de objeto do S3. Esses recursos ajudam a evitar alterações acidentais em dados críticos e permitem reverter ações não intencionais. Esse recurso é particularmente útil quando há vários usuários com permissões completas de gravação e execução acessando o console do Amazon S3.

Para obter informações sobre o Versionamento do S3, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#). Para obter mais informações sobre Bloqueio de objetos, consulte [Usar o bloqueio de objetos do S3 \(p. 722\)](#).

Gerenciamento do ciclo de vida do objeto para eficiência de custos

Para gerenciar os objetos para que eles sejam armazenados de forma econômica durante todo o ciclo de vida, você pode emparelhar políticas de ciclo de vida com o versionamento de objetos. As políticas de ciclo de vida definem as ações que você deseja que o S3 execute durante a vida útil de um objeto. Por exemplo, é possível criar uma política de ciclo de vida que fará a transição de objetos para outra classe de armazenamento, arquivá-los ou excluí-los após um período especificado. É possível definir uma política de ciclo de vida para todos os objetos ou para um subconjunto de objetos no bucket usando um prefixo ou uma tag compartilhados.

Para obter mais informações, consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

Replicação entre regiões para vários locais de escritório

Ao criar buckets que são acessados por diferentes locais de escritório, você deve considerar a implementação da replicação entre regiões do S3. A replicação entre regiões ajuda a garantir que todos os usuários tenham acesso aos recursos de que precisam e aumenta a eficiência operacional. A replicação entre regiões oferece maior disponibilidade mediante a cópia de objetos entre buckets do S3 em diferentes Regiões da AWS. No entanto, o uso dessa ferramenta aumenta os custos de armazenamento.

Para obter mais informações, consulte [Replicação de objetos \(p. 801\)](#).

Permissões para hospedagem segura de sites estáticos

Ao configurar um bucket para ser usado como um site estático acessado publicamente, é necessário desabilitar todas as configurações de bloqueio de acesso público. É importante fornecer apenas ações `s3:GetObject` e não permissões `ListObject` ou `PutObject` ao escrever a política de bucket para o site estático. Isso ajuda a garantir que os usuários não possam visualizar todos os objetos no bucket nem adicionar seu próprio conteúdo.

Para obter mais informações, consulte [Configuração de permissões para acesso ao site \(p. 1199\)](#).

O Amazon CloudFront fornece os recursos necessários para configurar um site estático seguro. Os sites estáticos do Amazon S3 só oferecem suporte a endpoints HTTP. O CloudFront usa o armazenamento durável do Amazon S3 ao mesmo tempo que fornece cabeçalhos de segurança adicionais, como HTTPS. O HTTPS adiciona segurança criptografando uma solicitação HTTP normal e protegendo-a contra ataques cibernéticos comuns.

Para obter mais informações, consulte [Conceitos básicos de um site estático seguro no Guia do desenvolvedor do Amazon CloudFront](#).

Compartilhar recursos

Existem várias maneiras diferentes de compartilhar recursos com um grupo específico de usuários. É possível usar as ferramentas a seguir para compartilhar um conjunto de documentos ou outros recursos com um único grupo de usuários, departamento ou escritório. Embora todos possam ser usados para atingir o mesmo objetivo, algumas ferramentas podem emparelhar melhor do que outras com as configurações existentes.

Propriedade de objetos do S3

O S3 Object Ownership é uma configuração no nível do bucket do Amazon S3 que você pode usar para desabilitar as ACLs e assumir a propriedade de cada objeto em seu bucket, simplificando o gerenciamento de acesso para dados armazenados no Amazon S3. Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. É possível usar Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Consequentemente, o controle de acesso para seus dados é baseado em políticas. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Políticas de usuário

É possível compartilhar recursos com um grupo limitado de pessoas usando políticas de grupos e de usuários do IAM. Ao criar um usuário do IAM, você será solicitado a criá-lo e adicioná-lo a um grupo. No entanto, é possível criar e adicionar usuários a grupos a qualquer momento. Se os indivíduos com os quais você pretende compartilhar esses recursos já estiverem configurados no IAM, será possível adicioná-los a um grupo comum e compartilhar o bucket com o grupo dentro da política de usuário. Também é possível usar políticas de usuário do IAM para compartilhar objetos individuais em um bucket.

Para obter mais informações, consulte [Permitir que um usuário do IAM acesse um dos seus buckets \(p. 546\)](#).

Listas de controle de acesso

Como regra geral, recomendamos que você use políticas de bucket do S3 ou políticas do IAM para controle de acesso. As ACLs do Amazon S3 são o mecanismo de controle de acesso original do Amazon S3 anterior ao IAM. Se você já usa ACLs do S3 e as considera suficientes, não há necessidade de alterar. No entanto, determinados cenários de controle de acesso exigem o uso de ACLs. Por exemplo, quando um proprietário de bucket deseja conceder permissão a objetos, mas nem todos os objetos são de propriedade dele, o proprietário do objeto deve primeiro conceder permissão ao proprietário do bucket. Isso é feito usando uma ACL de objeto.

A maioria dos casos de uso modernos no Amazon S3 não requer mais o uso de ACLs e, portanto, recomendamos desabilitar as ACLs exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com Object Ownership, é possível desabilitar ACLs e confiar em políticas para controle de acesso. Ao desabilitar ACLs, você pode facilmente manter um bucket com objetos carregados por diferentes Contas da AWS. Como proprietário do bucket, você possui todos os objetos do bucket e pode gerenciar o acesso a eles usando políticas.

Important

Se o bucket usar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Solicitações para definir ACLs ou atualizar ACLs falham e retornam o código de erro `AccessControlListNotSupported`. Ainda há suporte para solicitações de leitura de ACLs.

Para obter mais informações sobre como usar ACLs, consulte [Exemplo 3: O proprietário do bucket concede permissões para objetos que não possui \(p. 568\)](#).

Prefixos

Ao tentar compartilhar recursos específicos de um bucket, é possível replicar permissões no nível de pastas usando prefixos. O console do Amazon S3 oferece suporte ao conceito de pastas como meio de agrupar objetos utilizando um prefixo de nome compartilhado para os objetos. Você pode então especificar um prefixo de acordo com as condições da política de um usuário do IAM para conceder a ele permissão explícita para acessar os recursos associados a esse prefixo.

Para obter mais informações, consulte [Organizar objetos no console do Amazon S3 usando pastas \(p. 269\)](#).

Marcação

Se usar a marcação de objetos para categorizar o armazenamento, você poderá compartilhar objetos marcados com um valor específico com usuários especificados. A marcação de recursos permite controlar o acesso a objetos com base nas tags associadas ao recurso que um usuário está tentando acessar. Para fazer isso, use a condição `ResourceTag/key-name` dentro de uma política de usuário do IAM para permitir o acesso aos recursos marcados.

Para obter mais informações, consulte [Controle do acesso aos recursos da AWS usando tags de recursos no Guia do usuário do IAM](#).

Proteger dados

Use as seguintes ferramentas para ajudar a proteger os dados em trânsito e em repouso, ambas as quais são cruciais para manter a integridade e a acessibilidade dos dados.

Object encryption

O Amazon S3 oferece várias opções de criptografia de objetos que protegem os dados em trânsito e em repouso. A criptografia no lado do servidor criptografa o objeto antes de salvá-lo em discos em seus datacenters e os descriptografa ao fazer download dos objetos. Contanto que você autentique sua solicitação e tenha permissões de acesso, não há diferença na forma de acesso aos objetos criptografados ou não criptografados. Ao configurar a criptografia no lado do servidor, você tem três opções mutuamente exclusivas:

- Criptografia no lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3)
- Criptografia no lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS)
- Criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C)

Para obter mais informações, consulte [Proteção de dados usando criptografia no lado do servidor \(p. 359\)](#).

Criptografia no lado do cliente é o ato de criptografar os dados antes de enviá-los para o Amazon S3. Para obter mais informações, consulte [Proteger dados usando a criptografia no lado do cliente \(p. 405\)](#).

Métodos de assinatura

O Signature versão 4 é o processo para adicionar informações de autenticação às solicitações da AWS enviadas por HTTP. Por segurança, a maioria das solicitações para AWS deve ser assinada com uma chave de acesso, que consiste em um ID de chave de acesso e na chave de acesso secreta. Essas duas chaves são comumente conhecidas como suas credenciais de segurança.

Para obter mais informações, consulte [Autenticação de solicitações \(AWS Signature Version 4\) e Processo de assinatura do Signature Version 4](#).

Registro em log e monitoramento

O monitoramento é uma parte importante da manutenção da confiabilidade, da disponibilidade e da performance das soluções do Amazon S3 para que você possa depurar mais facilmente uma falha de vários pontos, caso isso ocorra. O registro em log pode fornecer insights sobre quaisquer erros que os usuários estejam recebendo e quando e quais solicitações são feitas. A AWS fornece várias ferramentas para monitorar seus recursos do Amazon S3:

- Amazon CloudWatch
- AWS CloudTrail
- Logs de acesso do Amazon S3
- AWS Trusted Advisor

Para obter mais informações, consulte [Registrar em log e monitorar no Amazon S3 \(p. 664\)](#).

O Amazon S3 é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, por uma função ou por um serviço da AWS no Amazon S3. Esse recurso pode ser emparelhado com o Amazon GuardDuty, que monitora ameaças contra seus recursos do Amazon S3 analisando eventos de gerenciamento do CloudTrail e eventos de dados do CloudTrail S3. Essas fontes de dados monitoram diferentes tipos de atividade. Por exemplo, os eventos de gerenciamento do CloudTrail relacionados ao S3 incluem operações que listam ou configuram projetos do S3. O GuardDuty analisa eventos de dados do S3 de todos os buckets do S3 e os monitora em busca de atividades maliciosas e suspeitas.

Para obter mais informações, consulte [Proteção do Amazon S3 no Amazon GuardDuty](#) no Guia do usuário do Amazon GuardDuty.

Tutoriais

Os tutoriais a seguir apresentam procedimentos de ponta a ponta para tarefas comuns do Amazon S3. Esses tutoriais foram projetados para um ambiente de tipo de laboratório e usam nomes de empresas, nomes de usuários fictícios e assim por diante. O objetivo é fornecer orientação geral. Eles não se destinam ao uso direto em um ambiente de produção sem uma revisão e adaptação cuidadosas para atender às necessidades exclusivas do ambiente da organização.

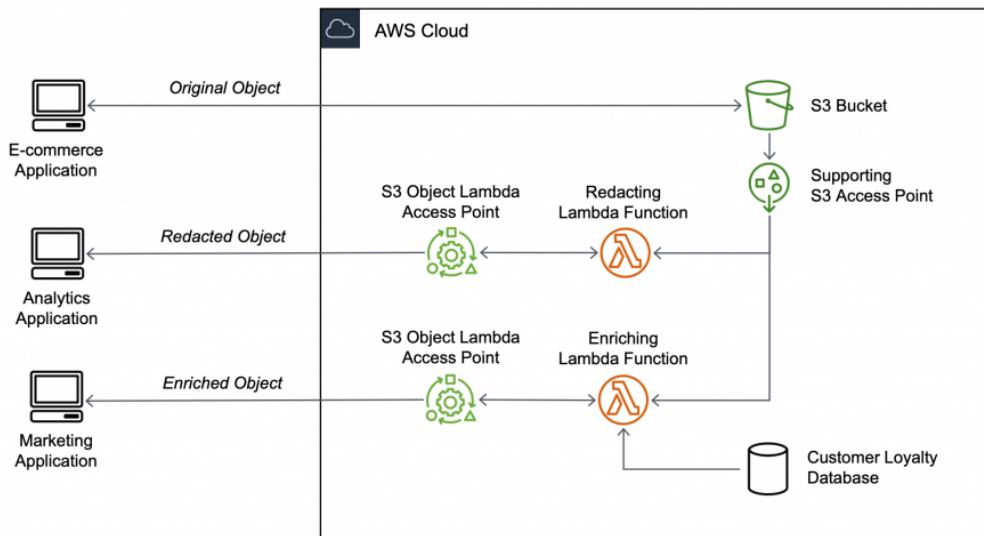
Tópicos

- [Tutorial: Como transformar dados para sua aplicação com o S3 Object Lambda \(p. 30\)](#)
- [Tutorial: Detectar e editar dados PII com o S3 Object Lambda e o Amazon Comprehend \(p. 45\)](#)
- [Tutorial: Hospedagem de transmissão sob demanda com o Amazon S3, Amazon CloudFront e Amazon Route 53 \(p. 56\)](#)
- [Tutorial: Vídeos de transcodificação em lote com operações em lote do S3, AWS Lambda e o AWS Elemental MediaConvert \(p. 70\)](#)
- [Tutorial: configurar um site estático no Amazon S3 \(p. 100\)](#)
- [Configurar um site estático usando um domínio personalizado registrado no Route 53 \(p. 105\)](#)

Tutorial: Como transformar dados para sua aplicação com o S3 Object Lambda

Quando você armazena dados no Amazon S3, você pode compartilhá-los facilmente para usar várias aplicações. No entanto, cada aplicação pode ter requisitos de formato de dados exclusivos e pode precisar de modificação ou processamento de seus dados para um caso de uso específico. Por exemplo, um conjunto de dados criado por uma aplicação de comércio eletrônico pode incluir informações de identificação pessoal (PII). Quando os mesmos dados são processados para análise, essas PII não são necessárias e devem ser editadas. No entanto, se o mesmo conjunto de dados for usado para uma campanha de marketing, talvez seja necessário enriquecer os dados com detalhes adicionais, como informações do banco de dados de fidelidade do cliente.

Com o [S3 Object Lambda](#) você pode adicionar seu próprio código para processar dados recuperados do S3 antes de retorná-los para uma aplicação. É possível, especificamente, configurar uma função do AWS Lambda e anexá-la a um ponto de acesso do S3 Object Lambda. Quando uma aplicação envia [Solicitações GET padrão do S3](#) por meio do ponto de acesso do S3 Object Lambda, a função Lambda especificada é chamada para processar quaisquer dados recuperados de um bucket do S3 por meio do ponto de acesso do S3 de suporte. Em seguida, o ponto de acesso do S3 Object Lambda retorna o resultado transformado de volta à aplicação. Você pode criar e executar suas próprias funções Lambda personalizadas, adaptando a transformação de dados do S3 Object Lambda ao seu caso de uso específico, tudo sem a necessidade de alterações em suas aplicações.



Objetivo

Neste tutorial, você aprende como adicionar código personalizado a solicitações GET padrão do S3 para modificar o objeto solicitado recuperado do S3 para que o objeto atenda às necessidades do cliente ou aplicação solicitante. Especificamente, você aprende como transformar todo o texto no objeto original armazenado no S3 para maiúsculas por meio do S3 Object Lambda.

Tópicos

- [Pré-requisitos \(p. 31\)](#)
- [Etapa 1: criar um bucket do S3 \(p. 33\)](#)
- [Etapa 2: fazer upload do arquivo para seu bucket do S3 \(p. 33\)](#)
- [Etapa 3: criar um ponto de acesso do S3 \(p. 34\)](#)
- [Etapa 4: Criar uma função Lambda \(p. 35\)](#)
- [Etapa 5: configurar uma política do IAM para a função de execução da função Lambda \(p. 39\)](#)
- [Etapa 6: criar um ponto de acesso do Object Lambda do S3 \(p. 40\)](#)
- [Etapa 7: exibir os dados transformados \(p. 41\)](#)
- [Etapa 8: limpar \(p. 43\)](#)
- [Next steps \(Próximas etapas\) \(p. 45\)](#)

Pré-requisitos

Antes de começar este tutorial, você deve ter uma conta da AWS na qual você possa fazer login como uma conta do AWS Identity and Access Management (IAM) com permissões corretas. Você também deve instalar a versão 3.8 ou posterior do Python.

Subetapas

- [Crie um usuário do IAM com permissões em sua conta da AWS \(console\) \(p. 32\)](#)
- [Instale o Python 3.8 ou posterior em sua máquina local \(p. 32\)](#)

Crie um usuário do IAM com permissões em sua conta da AWS (console)

Você pode criar um usuário do IAM para o tutorial ou pode adicionar permissões a um usuário do IAM existente. Para concluir este tutorial, o usuário do IAM deve anexar as seguintes políticas do IAM para acessar recursos da AWS e executar ações específicas.

Seu usuário do IAM requer as seguintes políticas:

- [AmazonS3FullAccess](#): concede permissões a todas as ações do Amazon S3, incluindo permissões para criar e usar um ponto de acesso do Object Lambda.
- [AWSLambda_FullAccess](#): concede permissões a todas as ações do Lambda.
- [IAMFullAccess](#): concede permissões a todas as ações do IAM.
- [IAMAccessAnalyzerReadOnlyAccess](#): concede permissões para ler todas as informações de acesso fornecidas pelo IAM Access Analyzer.

Note

Para simplificar, este tutorial usa políticas gerenciadas pela AWS de acesso total. Para uso em produção, recomendamos que você conceda apenas as permissões mínimas necessárias para seu caso de uso, de acordo com as [práticas recomendadas de segurança \(p. 672\)](#).

Para obter mais informações sobre como criar um usuário do IAM, consulte [Criar usuários do IAM \(console\)](#) no Guia do usuário do IAM.

Instale o Python 3.8 ou posterior em sua máquina local

Use o procedimento a seguir para instalar o Python 3.8 ou posterior em sua máquina local. Para obter instruções de instalação, consulte a página [Downloading Python \(Como baixar Python\)](#) no Guia do iniciante do Python.

1. Abra seu terminal local ou shell e execute o seguinte comando para determinar se o Python já está instalado e, em caso afirmativo, qual versão está instalada.

```
python --version
```

2. Se não tiver o Python 3.8 nem posterior, faça download do [instalador oficial](#) do Python 3.8 ou posterior que é adequado para sua máquina local.
3. Execute o instalador clicando duas vezes no arquivo baixado e siga as etapas para concluir a instalação.

Para os Usuários do Windows, escolha Adicionar Python 3.X ao PATH no assistente de instalação antes de escolher Instalar agora.

4. Reinicie o terminal fechando-o e reabrindo-o.
5. Execute o seguinte comando para verificar se o Python 3.8 ou posterior está instalado corretamente.

Para os usuários do macOS, execute este comando:

```
python3 --version
```

Para usuários do Windows, execute este comando:

```
python --version
```

6. Execute o comando a seguir para verificar se o gerenciador de pacotes pip3 está instalado. Se você vir um número de versão pip e python 3.8 ou posterior na resposta do comando, isso significa que o gerenciador de pacotes pip3 está instalado com sucesso.

```
pip --version
```

Etapa 1: criar um bucket do S3

Crie um bucket para armazenar os dados originais que você planeja transformar.

Para criar um bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Selecione Create bucket (Criar bucket).

A página Create bucket (Criar bucket) é aberta.

4. Para Bucket name (Nome do bucket), insira um nome para o seu bucket (por exemplo, **tutorial-bucket**).

Para obter mais informações sobre como nomear buckets no Amazon S3, consulte [Regras de nomeação de bucket \(p. 127\)](#).

5. Em Region (Região), escolha a Região da AWS onde deseja que o bucket resida.

Para obter mais informações sobre a região do bucket, consulte [Visão geral dos buckets \(p. 123\)](#).

6. Para Block Public Access settings for this bucket (Configurações de acesso de bloqueio público para este bucket), mantenha as configurações padrão (Block allpublic access (Bloquear todo acesso público) está habilitado).

Recomendamos que você mantenha todas as configurações de acesso de bloqueio público ativadas, a menos que precise desativar uma ou mais delas para seu caso de uso. Para obter mais informações sobre como bloquear o acesso público, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

7. Mantenha as configurações restantes definidas conforme os padrões.

(Opcional) Se quiser definir configurações de bucket adicionais para o caso de uso específico, consulte [Criação de um bucket \(p. 129\)](#).

8. Selecione Create bucket (Criar bucket).

Etapa 2: fazer upload do arquivo para seu bucket do S3

Carregue um arquivo de texto para o bucket do S3. Este arquivo de texto contém os dados originais que você transformará em maiúsculas posteriormente neste tutorial.

Por exemplo, você pode carregar um `tutorial.txt` que contém o seguinte texto:

```
Amazon S3 Object Lambda Tutorial:  
You can add your own code to process data retrieved from S3 before  
returning it to an application.
```

Fazer upload de um arquivo para um bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista de Buckets, escolha o nome do bucket que você criou na [Etapa 1 \(p. 33\)](#) (por exemplo, **tutorial-bucket**) para carregar seu arquivo.
4. Na guia Objects (Objetos) do bucket, escolha Upload (Fazer upload).
5. Na página Upload (Carregar), em Files and folders (Arquivos e pastas), escolha Add files (Adicionar arquivos).
6. Escolha um arquivo para carregar e, em seguida, escolha Open (Abrir). Por exemplo, você pode carregar o exemplo de arquivo `tutorial.txt` mencionado anteriormente.
7. Escolha Upload (Carregar).

Etapa 3: criar um ponto de acesso do S3

Para usar um ponto de acesso do S3 Object Lambda para acessar e transformar os dados originais, você deve criar um ponto de acesso do S3 e associá-lo ao bucket do S3 criado na [Etapa 1 \(p. 33\)](#). O ponto de acesso deve estar na mesma Região da AWS que os objetos que você deseja transformar.

Mais adiante neste tutorial, você usará esse ponto de acesso como um ponto de acesso de suporte para o ponto de acesso do Object Lambda.

Como criar um ponto de acesso

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Access Points (Pontos de acesso).
3. Na página Access Points (Pontos de acesso), escolha Create access point (Criar ponto de acesso).
4. No campo Access point name (Nome do ponto de acesso), insira o nome (por exemplo, **tutorial-access-point**) para o ponto de acesso.

Para obter mais informações sobre nomenclatura de pontos de acesso, consulte [Regras para nomear pontos de acesso do Amazon S3 \(p. 325\)](#).

5. No campo Bucket name (Nome do bucket), insira o nome do bucket criado na [Etapa 1 \(p. 33\)](#) (por exemplo, **tutorial-bucket**). O S3 anexa o ponto de acesso a este bucket.

(Opcional) Você pode escolher Browse S3 (Procurar S3) para navegar e pesquisar buckets na sua conta. Se você escolher Browse S3 (Procurar S3), selecione o bucket desejado e escolha Choose path (Escolher caminho) para preencher o campo Bucket name (Nome do bucket) com o nome do bucket.

6. Para Network origin (Origem de rede), escolha Internet.

Para obter mais informações sobre origens de rede para pontos de acesso, consulte [Criar pontos de acesso restritos a uma nuvem privada virtual \(p. 327\)](#).

7. Por padrão, todas as configurações de bloqueio de acesso público são habilitadas para seu ponto de acesso. Recomendamos manter a opção Block all public access (Bloquear todo o acesso público) ativada.

Para obter mais informações, consulte [Gerenciar o acesso público a pontos de acesso \(p. 329\)](#).

8. Para todas as outras configurações de ponto de acesso, mantenha as configurações padrão.

(Opcional) Você pode modificar as configurações do ponto de acesso para dar suporte ao caso de uso. Para este tutorial, recomendamos manter as configurações padrão.

(Opcional) Se você precisar gerenciar o acesso ao seu ponto de acesso, você pode especificar uma política de ponto de acesso. Para obter mais informações, consulte [Exemplos de política de ponto de acesso \(p. 322\)](#).

9. Selecione Create access point (Criar ponto de acesso).

Etapa 4: Criar uma função Lambda

Para transformar dados originais, crie uma função Lambda para usar com o ponto de acesso do S3 Object Lambda.

Subetapas

- [Gravar código de função Lambda e criar um pacote de implantação com um ambiente virtual \(p. 35\)](#)
- [Crie uma função Lambda com uma função de execução \(console\) \(p. 38\)](#)
- [Implante seu código de função Lambda com arquivos .zip e configure a função Lambda \(console\) \(p. 38\)](#)

Gravar código de função Lambda e criar um pacote de implantação com um ambiente virtual

1. Na sua máquina local, crie uma pasta com o nome da pasta `object-lambda` para que o ambiente virtual use posteriormente neste tutorial.
2. Na pasta `object-lambda`, crie um arquivo com uma função Lambda que altere todo o texto no objeto original para maiúsculas. Por exemplo, você pode usar a seguinte função gravada em Python. Salve esta função em um arquivo chamado `transform.py`.

```
import boto3
import requests

# This function capitalizes all text in the original object
def lambda_handler(event, context):
    object_context = event["getObjectContext"]
    # Get the presigned URL to fetch the requested original object
    # from S3
    s3_url = object_context["inputS3Url"]
    # Extract the route and request token from the input context
    request_route = object_context["outputRoute"]
    request_token = object_context["outputToken"]

    # Get the original S3 object using the presigned URL
    response = requests.get(s3_url)
    original_object = response.content.decode("utf-8")

    # Transform all text in the original object to uppercase
    # You can replace it with your custom code based on your use case
    transformed_object = original_object.upper()

    # Write object back to S3 Object Lambda
    s3 = boto3.client('s3')
    # The WriteGetObjectResponse API sends the transformed data
    # back to S3 Object Lambda and then to the user
    s3.write_get_object_response(
        Body=transformed_object,
        RequestRoute=request_route,
        RequestToken=request_token)
```

```
# Exit the Lambda function: return the status code
return {'status_code': 200}
```

Note

O exemplo anterior da função Lambda carrega todo o objeto solicitado na memória antes de transformá-lo e retorná-lo para o cliente. Como alternativa, você pode transmitir o objeto do S3 para evitar o carregamento do objeto inteiro na memória. Essa abordagem pode ser útil ao trabalhar com objetos grandes. Para obter mais informações sobre streaming de respostas com pontos de acesso do Object Lambda, consulte os exemplos de streaming em [Trabalhar com WriteGetObjectResponse \(p. 302\)](#).

Quando você está gravando uma função Lambda para uso com um ponto de acesso do S3 Object Lambda, a função é baseada no contexto de evento de entrada que o S3 Object Lambda fornece à função Lambda. O contexto do evento fornece informações sobre a solicitação que está sendo feita no evento passado do S3 Object Lambda para o Lambda. Ele contém os parâmetros que você usa para criar a função Lambda.

Os campos usados para criar a função Lambda anterior são os seguintes:

O campo `getObjectContext` significa os detalhes de entrada e saída para conexões com o Amazon S3 e S3 Object Lambda. Ele tem os seguintes campos:

- `inputs3Url`: um URL pré-assinado que a função Lambda pode usar para baixar o objeto original do ponto de acesso de suporte. Usando um URL pré-assinado, a função Lambda não precisa ter permissões de leitura do Amazon S3 para recuperar o objeto original e só pode acessar o objeto processado por cada chamada.
- `outputRoute`: um token de roteamento que é adicionado ao URL do Object Lambda do S3 quando a função Lambda chama `WriteGetObjectResponse` para enviar de volta o objeto transformado.
- `outputToken`: um token usado pelo S3 Object Lambda para corresponder à chamada `WriteGetObjectResponse` com o chamador original ao enviar de volta o objeto transformado.

Para obter mais informações sobre todos os campos no contexto de evento, consulte [Formato e uso de contexto de evento \(p. 313\)](#) e [Escrever e depurar funções do AWS Lambda para pontos de acesso do S3 Object Lambda \(p. 302\)](#).

3. No terminal local, insira o seguinte comando para instalar o comando do pacote `virtualenv`:

```
python -m pip install virtualenv
```

4. No terminal local, abra o `object-lambda` que você criou anteriormente e insira o seguinte comando para criar e inicializar um ambiente virtual chamado `venv`.

```
python -m virtualenv venv
```

5. Para ativar o ambiente virtual, insira o seguinte comando para executar o arquivo `activate` da pasta do ambiente:

Para os usuários do macOS, execute este comando:

```
source venv/bin/activate
```

Para usuários do Windows, execute este comando:

```
.\venv\Scripts\activate
```

Agora, seu prompt de comando é alterado para mostrar (venv), indicando que o ambiente virtual está ativo.

6. Para instalar as bibliotecas necessárias, execute os seguintes comandos linha por linha no ambiente virtual do venv.

Esses comandos instalam versões atualizadas das dependências de sua função Lambda `lambda_handler`. Essas dependências são o AWSSDK for Python (Boto3) e o módulo de solicitações.

```
pip3 install boto3
```

```
pip3 install requests
```

7. Para desativar o ambiente virtual, execute o seguinte comando:

```
deactivate
```

8. Para criar um pacote de implantação com as bibliotecas instaladas como um arquivo `.zip` chamado `lambda.zip` na raiz do diretório `.zip`, execute os comandos a seguir, linha por linha, em seu terminal local.

Tip

Os comandos a seguir talvez precisem ser ajustados para funcionar em seu ambiente específico. Por exemplo, uma biblioteca pode aparecer em `site-packages` ou em `dist-packages` e a primeira pasta pode ser `lib` ou `lib64`. Além disso, a pasta `python` pode ser nomeada com uma versão Python diferente. Use o comando `pip show` para localizar um pacote específico.

Para usuários do macOS, execute estes comandos:

```
cd venv/lib/python3.8/site-packages
```

```
zip -r ../../../../../../lambda.zip .
```

Para usuários do Windows, execute estes comandos:

```
cd .\venv\Lib\site-packages\
```

```
powershell Compress-Archive * ../../../../../../lambda.zip
```

O último comando salva o pacote de implantação na raiz do diretório do `object-lambda`.

9. Adicione o arquivo de código de função `transform.py` à raiz do pacote de implantação.

Para usuários do macOS, execute estes comandos:

```
cd ../../../../../../
```

```
zip -g lambda.zip transform.py
```

Para usuários do Windows, execute estes comandos:

Versão da API 2006-03-01

```
cd ..\..\..\
```

```
powershell Compress-Archive -update transform.py lambda.zip
```

Depois de concluir esta etapa, você deverá ter seguinte estrutura de diretórios:

```
lambda.zip$  
# transform.py  
# __pycache__  
| boto3/  
# certifi/  
# pip/  
# requests/  
...
```

Crie uma função Lambda com uma função de execução (console)

1. Faça login no AWS Management Console e abra o console AWS Lambda em <https://console.aws.amazon.com/lambda/>.
2. No painel de navegação à esquerda, escolha Functions (Funções).
3. Escolha Create function (Criar função).
4. Escolha Author from scratch (Criar do zero).
5. Em Basic information (Informações básicas), faça o seguinte:
 - a. Em Function name (Nome da função), insira **tutorial-object-lambda-function**.
 - b. Para Runtime (Tempo de execução), escolha Python 3.8 ou uma versão posterior.
6. Expandir a seção Alterar função de execução padrão. Em Execution role (Função de execução), selecione Create a new role with basic Lambda permissions (Criar uma função com permissões básicas do Lambda).

Na [Etapa 5 \(p. 39\)](#), posteriormente neste tutorial, você anexa o AmazonS3ObjectLambdaExecutionRolePolicy a essa função de execução do Lambda.
7. Mantenha as configurações restantes definidas conforme os padrões.
8. Escolha Create function (Criar função).

Implante seu código de função Lambda com arquivos .zip e configure a função Lambda (console)

1. No console do AWS Lambda em <https://console.aws.amazon.com/lambda/>, escolha Functions (Funções) no painel de navegação à esquerda.
2. Escolha a função Lambda que você criou anteriormente (por exemplo, **tutorial-object-lambda-function**).
3. Na página de detalhes da função Lambda, selecione a guia Code (Código). Na seção Code Source (Origem do código), escolha Upload from (Carregar de) e depois arquivo .zip.
4. Selecione Upload (Carregar) para selecionar seu arquivo .zip local.
5. Selecione o arquivo lambda.zip que você criou anteriormente e, em seguida, selecione Open (Abrir).

6. Escolha Save (Salvar).
7. Na seção Runtime settings (Configurações do tempo de execução), escolha Edit (Editar).
8. Na página Edit runtime settings (Editar configurações do tempo de execução), confirme se Runtime (Tempo de execução) foi definido como Python 3.8 ou uma versão posterior.
9. Para informar ao tempo de execução do Lambda qual método de handler em seu código de função Lambda chamar, insira `transform.lambda_handler` para Handler.

Ao configurar uma função em Python, o valor da configuração do handler é o nome do arquivo e o nome do módulo do handler exportado, separados por um ponto. Por exemplo, `transform.lambda_handler` chama o método `lambda_handler` definido no arquivo `transform.py`.

10. Escolha Save (Salvar).
11. (Opcional) Na página de detalhes da função Lambda, escolha a guia Configuration (Configuração). No painel de navegação à esquerda, selecione General configuration (Configuração geral) e, depois, escolha Edit (Editar). No campo Timeout (Tempo limite), insira 1 min 0 segundos. Mantenha as configurações restantes definidas, conforme os padrões e escolha Save (Salvar).

Timeout (Tempo limite) é a quantidade de tempo durante a qual o Lambda permite que uma função seja executada antes de interrompê-la. O padrão é 3 segundos. A duração máxima para uma função Lambda usada pelo S3 Object Lambda é de 60 segundos. O preço é baseado na quantidade de memória configurada e na quantidade de tempo em que o código é executado.

Etapa 5: configurar uma política do IAM para a função de execução da função Lambda

Para habilitar sua função Lambda para fornecer dados personalizados e cabeçalhos de resposta para o chamador de `GetObject`, a função de execução da função Lambda precisa ter permissões do IAM para chamar a API `WriteGetObjectResponse`.

Para anexar uma política do IAM à atribuição da função Lambda

1. No console do AWS Lambda em <https://console.aws.amazon.com/lambda/>, escolha Functions (Funções) no painel de navegação à esquerda.
2. Escolha a função que você criou na [Etapa 4 \(p. 35\)](#) (por exemplo, `tutorial-object-lambda-function`).
3. Na página de detalhes da função Lambda, selecione a guia Configuration (Configuração) e, depois, escolha Permission (Permissões) no painel de navegação à esquerda.
4. Em Execution role (Função de execução), escolha o link do Role name (Nome da função). O console do IAM é aberto.
5. Na página Summary (Resumo) do console do IAM para a função de execução da função Lambda, escolha a guia Permissions (Permissões) e depois escolha Attach policies (Anexar políticas).
6. Na página Attach Permissions (Anexar permissões), insira `AmazonS3ObjectLambdaExecutionRolePolicy` na caixa de pesquisa para filtrar a lista de políticas. Marque a caixa de seleção ao lado do nome da política `AmazonS3ObjectLambdaExecutionRolePolicy`.
7. Escolha Attach policy (Anexar política).

Etapa 6: criar um ponto de acesso do Object Lambda do S3

Um ponto de acesso do S3 Object Lambda fornece a flexibilidade de chamar uma função Lambda diretamente de uma solicitação GET do S3 para que a função possa processar dados recuperados de um ponto de acesso do S3. Ao criar e configurar um ponto de acesso do S3 Object Lambda, você deve especificar a função Lambda para chamar e fornecer o contexto do evento no formato JSON como parâmetros personalizados para o Lambda usar.

Criar um ponto de acesso do S3 Object Lambda

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Object Lambda access points (Pontos de acesso do Object Lambda).
3. Na página Object Lambda Access Points (Pontos de acesso do Object Lambda), escolha Create Object Lambda Access Point (Criar ponto de acesso do Object Lambda).
4. Para Object Lambda Access Point name (Nome do ponto de acesso do Object Lambda), informe o nome que você deseja usar para o ponto de acesso do Object Lambda (por exemplo, **tutorial-object-lambda-accesspoint**).
5. Para Supporting Access Point (Ponto de acesso de suporte), informe ou procure o ponto de acesso padrão criado na [Etapa 3 \(p. 34\)](#) (por exemplo, **tutorial-access-point**) e, em seguida, escolha Choose supporting Access Point (Escolher ponto de acesso de suporte).
6. Para Invoke Lambda function (Chamar função Lambda), você pode escolher qualquer uma das duas opções a seguir para este tutorial.
 - Selecione Choose from functions in your account (Escolher das funções de sua conta) e escolha a função Lambda que você criou na [Etapa 4 \(p. 35\)](#) (por exemplo, **tutorial-object-lambda-function**) da lista suspensa Lambda function (Função Lambda).
 - Escolha Enter ARN (Inserir ARN) e depois informe o Nome do recurso da Amazon (ARN) da função Lambda que você criou na [Etapa 4 \(p. 35\)](#).
7. Para Lambda function version (Versão da função Lambda), escolha \$LATEST (a versão mais recente da função Lambda que você criou na [Etapa 4 \(p. 35\)](#)).
8. (Opcional) Se precisar da função Lambda para reconhecer e processar solicitações GET com cabeçalhos de intervalo e número de peça, selecione Lambda function supports requests using range (A função Lambda suporta solicitações usando intervalo) e Lambda function supports requests using part numbers (A função Lambda suporta solicitações usando números de parte). Caso contrário, desmarque essas duas caixas de seleção.

Para obter mais informações sobre como usar números de intervalo ou de parte com o S3 Object Lambda, consulte [Trabalhar com cabeçalhos Range e partNumber \(p. 311\)](#).

9. (Opcional) Em Payload - optional (Carga útil - opcional), adicione um texto JSON para fornecer informações adicionais à sua função Lambda.

Uma carga útil é um texto JSON opcional que você pode fornecer à sua função Lambda como entrada para todas as chamadas provenientes de um ponto de acesso do S3 Object Lambda específico.

Para personalizar os comportamentos de vários pontos de acesso do Object Lambda que chamam a mesma função Lambda, você pode configurar cargas úteis com diferentes parâmetros, estendendo, assim, a flexibilidade da sua função Lambda.

Para obter mais informações sobre carga útil, consulte [Formato e uso de contexto de evento \(p. 313\)](#).

10. (Opcional) Para Request metrics - optional (Métricas de solicitação - opcional), escolha Disable (Desabilitar) ou Enable (Habilitar) para adicionar o monitoramento do Amazon S3 ao seu ponto de

acesso do Object Lambda. As métricas de solicitação são cobradas na taxa padrão do Amazon CloudWatch. Para obter mais informações, consulte [Preço do CloudWatch](#).

11. Em Object Lambda Access Point policy - optional (Política do ponto de acesso do Object Lambda - opcional), mantenha a configuração padrão.
(Opcional) Você pode definir uma política de recursos. Essa política de recursos concede permissão da API GetObject para usar o ponto de acesso do Object Lambda especificado.
12. Mantenha as configurações restantes definidas conforme os padrões e escolha Create Object Lambda Access Point (Criar ponto de acesso do Object Lambda).

Etapa 7: exibir os dados transformados

Agora, o S3 Object Lambda está pronto para transformar seus dados para seu caso de uso. Neste tutorial, o S3 Object Lambda transforma todo o texto em seu objeto em maiúsculas.

Subetapas

- [Exibir os dados transformados no ponto de acesso do S3 Object Lambda \(p. 41\)](#)
- [Execute um script Python para imprimir os dados originais e transformados \(p. 41\)](#)

Exibir os dados transformados no ponto de acesso do S3 Object Lambda

Quando você solicita para recuperar um arquivo por meio do ponto de acesso do S3 Object Lambda, você faz uma chamada de API GetObject para o S3 Object Lambda. O S3 Object Lambda chama a função Lambda para transformar seus dados e, em seguida, retorna os dados transformados como a resposta à chamada de API de GetObject do S3 padrão.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Object Lambda access points (Pontos de acesso do Object Lambda).
3. Na página Object Lambda Access Points (Pontos de acesso do Object Lambda), escolha o ponto de acesso do S3 Object Lambda que você criou na [Etapa 6 \(p. 40\)](#) (por exemplo, **tutorial-object-lambda-accesspoint**).
4. Na guia Objects (Objetos) de seu ponto de acesso do S3 Object Lambda, selecione o arquivo que tem o mesmo nome (por exemplo, **tutorial.txt**) daquele que você carregou no bucket do S3 na [Etapa 2 \(p. 33\)](#).

Esse arquivo deve conter todos os dados transformados.

5. Para exibir os dados transformados, escolha Open (Abrir) ou Download (Baixar).

Execute um script Python para imprimir os dados originais e transformados

Você pode usar o S3 Object Lambda com suas aplicações existentes. Para fazer isso, atualize a configuração da aplicação para usar o novo ARN do ponto de acesso do S3 Object Lambda criado na [Etapa 6 \(p. 40\)](#) para recuperar dados do S3.

O exemplo de script Python a seguir imprime os dados originais do bucket S3 e os dados transformados do ponto de acesso do S3 Object Lambda.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Object Lambda access points (Pontos de acesso do Object Lambda).
3. Na página Object Lambda Access Points (Pontos de acesso do Object Lambda), selecione o botão de opção à esquerda do ponto de acesso do S3 Object Lambda que você criou na [Etapa 6 \(p. 40\)](#) (por exemplo, **tutorial-object-lambda-accesspoint**).
4. Escolha Copy ARN (Copiar ARN).
5. Salve o ARN para uso mais tarde.
6. Grave um script Python em sua máquina local para imprimir os dados originais (por exemplo, **tutorial.txt**) do seu S3 Bucket e os dados transformados (por exemplo, **tutorial_transformed.txt**) do ponto de acesso do S3 Object Lambda. Você pode usar o seguinte script de exemplo:

```
import boto3

s3 = boto3.client('s3')

def getObject(bucket, key):
    objectBody = s3.get_object(Bucket = bucket, Key = key)
    print(objectBody["Body"].read().decode("utf-8"))
    print("\n")

print('Original object from the S3 bucket:')
# Replace the two input parameters of getObject() below with
# the S3 bucket name that you created in Step 1 \(p. 33\) and
# the name of the file that you uploaded to the S3 bucket in Step 2 \(p. 33\)
getObject("tutorial-bucket",
          "tutorial.txt")

print('Object transformed by S3 Object Lambda:')
# Replace the two input parameters of getObject() below with
# the ARN of your S3 Object Lambda access point that you saved earlier and
# the name of the file with the transformed data (which in this case is
# the same as the name of the file that you uploaded to the S3 bucket
# in Step 2 \(p. 33\))
getObject("arn:aws:s3-object-lambda:us-west-2:111122223333:accesspoint/tutorial-object-
lambda-accesspoint",
          "tutorial.txt")
```

7. Salve seu script Python com um nome personalizado (por exemplo, **tutorial_print.py**) na pasta (por exemplo, **object-lambda**) que você criou na [Etapa 4 \(p. 35\)](#) na sua máquina local.
8. No terminal local, execute o seguinte comando da raiz do diretório (por exemplo, **object-lambda**) que você criou na [Etapa 4 \(p. 35\)](#).

```
python3 tutorial_print.py
```

Você deve ver os dados originais e os dados transformados (todo o texto em maiúsculas) através do terminal. Por exemplo, você deve ver algo parecido com o texto a seguir.

```
Original object from the S3 bucket:
Amazon S3 Object Lambda Tutorial:
You can add your own code to process data retrieved from S3 before
returning it to an application.

Object transformed by S3 Object Lambda:
AMAZON S3 OBJECT LAMBDA TUTORIAL:
YOU CAN ADD YOUR OWN CODE TO PROCESS DATA RETRIEVED FROM S3 BEFORE
RETURNING IT TO AN APPLICATION.
```

Etapa 8: limpar

Se você transformou seus dados por meio do S3 Object Lambda apenas como um exercício de aprendizado, exclua os recursos da AWS que você alocou para que não haja mais encargos.

Subetapas

- [Excluir o ponto de acesso do Object Lambda \(p. 43\)](#)
- [Exclua o ponto de acesso do S3 \(p. 43\)](#)
- [Exclua a função de execução de sua função Lambda \(p. 43\)](#)
- [Excluir a função Lambda \(p. 44\)](#)
- [Excluir o grupo de logs do CloudWatch \(p. 44\)](#)
- [Exclua o arquivo original no bucket de origem do S3 \(p. 44\)](#)
- [Exclua o bucket de origem do S3 \(p. 44\)](#)
- [Excluir o usuário do IAM \(p. 45\)](#)

Excluir o ponto de acesso do Object Lambda

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Object Lambda access points (Pontos de acesso do Object Lambda).
3. Na página Object Lambda Access Points (Pontos de acesso do Object Lambda), selecione o botão de opção à esquerda do ponto de acesso do S3 Object Lambda que você criou na [Etapa 6 \(p. 40\)](#) (por exemplo, **tutorial-object-lambda-accesspoint**).
4. Escolha Delete (Excluir).
5. Confirme se deseja excluir o ponto de acesso do Lambda, inserindo o nome no campo de texto exibido e escolha Delete (Excluir).

Exclua o ponto de acesso do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Access Points (Pontos de acesso).
3. Navegue até o ponto de acesso que você criou na [Etapa 3 \(p. 34\)](#) (por exemplo, **tutorial-access-point**) e escolha o botão de opção ao lado do nome do ponto de acesso.
4. Escolha Delete (Excluir).
5. Confirme se deseja excluir o ponto de acesso inserindo o nome no campo de texto exibido e escolha Delete (Excluir).

Exclua a função de execução de sua função Lambda

1. Faça login no AWS Management Console e abra o console AWS Lambda em <https://console.aws.amazon.com/lambda/>.
2. No painel de navegação à esquerda, escolha Functions (Funções).
3. Escolha a função que você criou na [Etapa 4 \(p. 35\)](#) (por exemplo, **tutorial-object-lambda-function**).

4. Na página de detalhes da função Lambda, selecione a guia Configuration (Configuração) e, depois, escolha Permission (Permissões) no painel de navegação à esquerda.
5. Em Execution role (Função de execução), escolha o link do Role name (Nome da função). O console do IAM é aberto.
6. Na página Summary (Resumo) do console do IAM da função de execução da função Lambda, selecione Delete role (Excluir função).
7. Na caixa de diálogo Delete role (Excluir função), selecione Yes, delete (Sim, excluir).

Excluir a função Lambda

1. No console do AWS Lambda em <https://console.aws.amazon.com/lambda/>, escolha Functions (Funções) no painel de navegação à esquerda.
2. Marque a caixa de seleção à esquerda do nome da função que você criou na [Etapa 4 \(p. 35\)](#) (por exemplo, **tutorial-object-lambda-function**).
3. Escolha Actions (Ações) e, em seguida, escolha Delete (Excluir).
4. Na caixa de diálogo Delete function (Excluir função), escolha Delete (Excluir).

Excluir o grupo de logs do CloudWatch

1. Abra o console do CloudWatch em <https://console.aws.amazon.com/cloudwatch/>.
2. No painel de navegação esquerdo, escolha Log groups (Grupos de log).
3. Localize o grupo de logs cujo nome termina com a função Lambda que você criou na [Etapa 4 \(p. 35\)](#) (por exemplo, **tutorial-object-lambda-function**).
4. Marque a caixa de seleção à esquerda do nome do grupo de logs.
5. Escolha Actions (Ações) e Delete log group(s) (Excluir grupo(s) de log).
6. Na caixa de diálogo Delete log group(s) (Excluir grupo(s) de logs), escolha Delete (Excluir).

Exclua o arquivo original no bucket de origem do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Bucket name (Nome do bucket), escolha o nome do bucket para o qual você carregou o arquivo original na [Etapa 2 \(p. 33\)](#) (por exemplo, **tutorial-bucket**).
4. Marque a caixa de seleção à esquerda do nome do objeto que você deseja excluir (por exemplo, **tutorial.txt**).
5. Escolha Delete (Excluir).
6. Na página Delete objects (Excluir objetos) na seção Permanently delete objects? (Excluir objetos permanentemente?), confirme se deseja excluir este objeto informando **permanently delete** na caixa de texto.
7. Escolha Delete objects (Excluir objetos).

Exclua o bucket de origem do S3

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

2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista de Buckets, escolha o botão de opção ao lado do nome do bucket que você criou na [Etapa 1 \(p. 33\)](#) (por exemplo, **tutorial-bucket**).
4. Escolha Delete (Excluir).
5. Na página Delete bucket (Excluir bucket), confirme se deseja excluir o bucket inserindo o nome do bucket no campo de texto e escolha Delete bucket (Excluir bucket).

Excluir o usuário do IAM

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação esquerdo, escolha Users (Usuários) e marque a caixa de seleção ao lado do nome do usuário que você deseja excluir.
3. Na parte superior da página, escolha Delete (Excluir).
4. Na caixa de diálogo Delete **user name?** (Excluir nome de usuário?), insira o nome de usuário no campo de entrada de texto para confirmar a exclusão do usuário. Escolha Delete (Excluir).

Next steps (Próximas etapas)

Após concluir este tutorial, você pode personalizar a função Lambda para o caso de uso para modificar os dados retornados por solicitações S3 GET padrão.

Veja a seguir uma lista de casos de uso comuns para o S3 Object Lambda:

- Mascaramento de dados confidenciais para segurança e conformidade.

Para obter mais informações, consulte [Tutorial: Detectar e editar dados PII com o S3 Object Lambda e o Amazon Comprehend \(p. 45\)](#).

- Filtragem de determinadas linhas de dados para fornecer informações específicas.
- Aumento de dados com informações de outros serviços ou bancos de dados.
- Conversão entre formatos de dados, como conversão de XML em JSON para compatibilidade de aplicações.
- Compactação ou descompactação de arquivos enquanto eles estão sendo baixados.
- Redimensionamento e marcação d'água de imagens.
- Implementação de regras de autorização personalizadas para acessar dados.

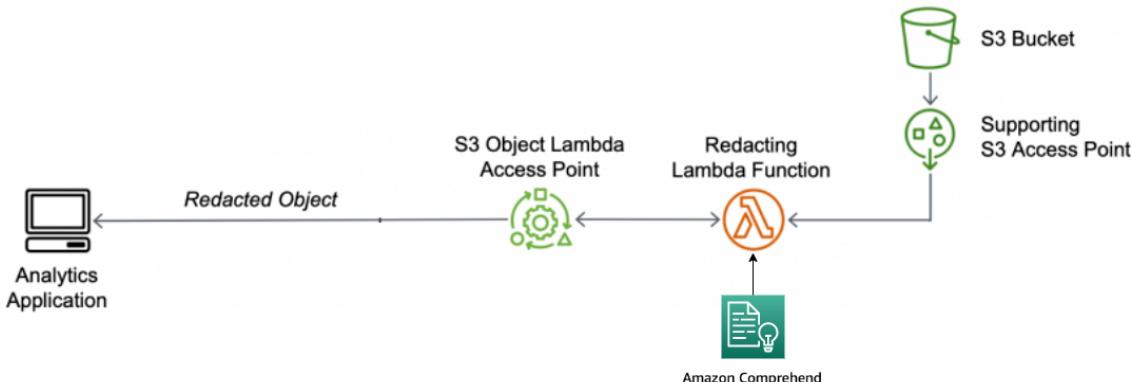
Para obter mais informações sobre o S3 Object Lambda, consulte [Transformar objetos com o S3 Object Lambda \(p. 288\)](#).

Tutorial: Detectar e editar dados PII com o S3 Object Lambda e o Amazon Comprehend

Quando você estiver usando o Amazon S3 para conjuntos de dados compartilhados para várias aplicações e os usuários acessarem, é importante restringir informações privilegiadas, como informações de identificação pessoal (PII) apenas para entidades autorizadas. Por exemplo, quando uma aplicação de marketing usa alguns dados contendo PII, ela pode precisar primeiro mascarar dados PII para atender aos requisitos de privacidade de dados. Além disso, quando um aplicação de análise usa um conjunto de

dados de inventário de ordem de produção, talvez seja necessário primeiro editar as informações do cartão de crédito do cliente para evitar perdas não intencionais de dados.

Com o [S3 Object Lambda](#) e uma função do AWS Lambda pré-construída, habilitada pelo Amazon Comprehend, você pode proteger dados de PII recuperados do S3 antes de retorná-los para uma aplicação. Especificamente, é possível usar a [função Lambda](#) pré-criada como função de edição e anexá-la a um ponto de acesso do S3 Object Lambda. Quando uma aplicação (por exemplo, uma aplicação de análise) envia [solicitações GET do S3 padrão](#), essas solicitações feitas por meio do ponto de acesso do S3 Object Lambda chamam a função Lambda de redação pré-criada para detectar e editar dados de PII recuperados de um bucket do S3, por meio de um ponto de acesso do S3 de suporte. Em seguida, o ponto de acesso do S3 Object Lambda retorna o resultado editado de volta à aplicação.



No processo, a função do Lambda pré-criada usa o [Amazon Comprehend](#), um serviço de processamento de linguagem natural (PNL) para registrar variações em como as PII são representadas, independentemente de como as PII existem no texto (por exemplo, numericamente ou uma combinação de palavras e números). O Amazon Comprehend pode até usar o contexto no texto para entender se um número de quatro dígitos é um PIN, os últimos quatro números de um número de Seguridade Social (SSN) ou um ano. O Amazon Comprehend processa qualquer arquivo de texto no formato UTF-8 e pode proteger as PII em escala sem afetar a precisão. Para obter mais informações, consulte [O que é Amazon Comprehend?](#) no Guia do desenvolvedor do Amazon Comprehend.

Objetivo

Neste tutorial, você aprenderá a usar o S3 Object Lambda com a função Lambda pré-criada `ComprehendPiRedactionS3ObjectLambda`. Essa função usa o Amazon Comprehend para detectar entidades de PII. Em seguida, ele edita essas entidades substituindo-as por asteriscos. Ao editar as PII, você oculta dados sigilosos, o que pode ajudar com segurança e conformidade.

Você também aprende a usar e configurar uma função AWS Lambda pré-criada no [AWS Serverless Application Repository](#) para trabalhar em conjunto com o S3 Object Lambda para facilitar a implantação.

Tópicos

- [Pré-requisitos: criar um usuário do IAM com permissões \(p. 47\)](#)
- [Etapa 1: criar um bucket do S3 \(p. 48\)](#)
- [Etapa 2: fazer upload do arquivo para seu bucket do S3 \(p. 49\)](#)
- [Etapa 3: criar um ponto de acesso do S3 \(p. 49\)](#)
- [Etapa 4: configurar e implantar uma função Lambda pré-construída \(p. 50\)](#)
- [Etapa 5: criar um ponto de acesso do Object Lambda do S3 \(p. 51\)](#)
- [Etapa 6: usar o ponto de acesso do S3 Object Lambda para recuperar o arquivo editado \(p. 52\)](#)
- [Etapa 7: limpeza \(p. 53\)](#)
- [Next steps \(Próximas etapas\) \(p. 55\)](#)

Pré-requisitos: criar um usuário do IAM com permissões

Antes de começar este tutorial, você deve ter uma conta da AWS na qual você possa fazer login como uma conta do AWS Identity and Access Management (IAM) com permissões corretas.

Você pode criar um usuário do IAM para o tutorial ou pode adicionar permissões a um usuário do IAM existente. Para concluir este tutorial, o usuário do IAM deve anexar as seguintes políticas do IAM para acessar recursos da AWS e executar ações específicas.

Note

Para simplificar, este tutorial usa políticas de acesso total. Para uso em produção, recomendamos que você conceda apenas as permissões mínimas necessárias para seu caso de uso, de acordo com as [práticas recomendadas de segurança \(p. 672\)](#).

Seu usuário do IAM requer as seguintes políticas gerenciadas pela AWS:

- [AmazonS3FullAccess](#): concede permissões a todas as ações do Amazon S3, incluindo permissões para criar e usar um ponto de acesso do Object Lambda.
- [AWSLambda_FullAccess](#): concede permissões a todas as ações do Lambda.
- [AWSCloudFormationFullAccess](#): concede permissões a todas as ações do AWS CloudFormation.
- [IAMFullAccess](#): concede permissões a todas as ações do IAM.
- [IAMAccessAnalyzerReadOnlyAccess](#): concede permissões para ler todas as informações de acesso fornecidas pelo IAM Access Analyzer.

Você pode anexar diretamente essas políticas existentes ao criar um usuário do IAM. Para obter mais informações sobre como criar um usuário do IAM, consulte [Criar usuários do IAM \(console\)](#) no Guia do usuário do IAM.

Além disso, seu usuário do IAM requer uma política gerenciada pelo cliente. Para conceder permissões de usuário do IAM a todos os recursos e ações do AWS Serverless Application Repository, você deve criar uma política do IAM e anexá-la ao usuário do IAM.

Para criar e anexar uma política do IAM para um usuário do IAM

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação à esquerda, escolha Policies (Políticas).
3. Escolha Create policy (Criar política).
4. Na guia Visual editor (Editor visual) de Service (Serviço), selecione Choose a service (Escolher um serviço). Em seguida, escolha Serverless Application Repository (Repositório de aplicações sem servidor).
5. Para Actions (Ações), em Manual actions (Ações manuais), selecione All Serverless Application Repository actions (serverlessrepo:*) (Todas as ações do Serverless Application Repository (serverlessrepo: *)) para este tutorial.

Como uma prática recomendada de segurança, você deve conceder permissões somente para as ações e os recursos dos quais um usuário precisa, com base no seu caso de uso. Para obter mais informações, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

6. Para Resources (Recursos), escolha All resources (Todos os recursos) para este tutorial.

Como prática recomendada, você deve definir permissões somente para recursos específicos em contas específicas. Como alternativa, você pode conceder menos privilégios usando chaves de

condição. Para obter mais informações, consulte [Grant least privilege \(Conceder privilégio mínimo\)](#) no Guia do usuário do IAM.

7. Escolha Next: Tags (Próximo: etiquetas).
8. Selecione Next: Review (Próximo: revisar).
9. Na página Review policy (Revisar política), insira um Name (Nome) (por exemplo, **tutorial-serverless-application-repository**) e uma Description (Descrição) (opcional) para a política que você está criando. Revise o resumo da política para assegurar-se de ter concedido as permissões que pretendia e, em seguida, escolha Create policy (Criar política) para salvar sua nova política.
10. No painel de navegação à esquerda, escolha Users (Usuários). Em seguida, escolha o usuário do IAM para este tutorial.
11. Na página Summary (Resumo) do usuário escolhido, escolha a guia Permissions (Permissões) e escolha Add permissions (Adicionar permissões).
12. Em Grant permissions (Conceder permissões), escolha Attach existing policies directly (Anexar políticas existentes diretamente).
13. Marque a caixa de seleção ao lado da política que você acabou de criar (por exemplo, **tutorial-serverless-application-repository**) e, depois, escolha Next: Review (Próximo: Revisar).
14. Em Permissions summary (Resumo de permissões), revise o resumo para se certificar de que anexou a política que pretendia. Em seguida, selecione Add permissions (Adicionar permissões).

Etapa 1: criar um bucket do S3

Crie um bucket para armazenar os dados originais que você planeja transformar.

Para criar um bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
 2. No painel de navegação à esquerda, escolha Buckets.
 3. Selecione Create bucket (Criar bucket).
- A página Create bucket (Criar bucket) é aberta.
4. Para Bucket name (Nome do bucket), insira um nome para o seu bucket (por exemplo, **tutorial-bucket**).

Para obter mais informações sobre como nomear buckets no Amazon S3, consulte [Regras de nomeação de bucket \(p. 127\)](#).

5. Em Region (Região), escolha a Região da AWS onde deseja que o bucket resida.

Para obter mais informações sobre a região do bucket, consulte [Visão geral dos buckets \(p. 123\)](#).

6. Para Block Public Access settings for this bucket (Configurações de acesso de bloqueio público para este bucket), mantenha as configurações padrão (Block allpublic access (Bloquear todo acesso público) está habilitado).

Recomendamos que você mantenha todas as configurações de acesso de bloqueio público ativadas, a menos que precise desativar uma ou mais delas para seu caso de uso. Para obter mais informações sobre como bloquear o acesso público, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

7. Mantenha as configurações restantes definidas conforme os padrões.

(Opcional) Se quiser definir configurações de bucket adicionais para o caso de uso específico, consulte [Criação de um bucket \(p. 129\)](#).

8. Selecione Create bucket (Criar bucket).

Etapa 2: fazer upload do arquivo para seu bucket do S3

Carregue um arquivo de texto contendo dados de PII conhecidos de vários tipos, como nomes, informações bancárias, números de telefone e SSNs, para o bucket do S3 como os dados originais dos quais você editará as PII posteriormente neste tutorial.

Por exemplo, você pode carregar seguindo o arquivo `tutorial.txt`. Este é um exemplo de arquivo de entrada do Amazon Comprehend.

```
Hello Zhang Wei, I am John. Your AnyCompany Financial Services,  
LLC credit card account 1111-0000-1111-0008 has a minimum payment  
of $24.53 that is due by July 31st. Based on your autopay settings,  
we will withdraw your payment on the due date from your  
bank account number XXXXXX1111 with the routing number XXXXX0000.  
  
Your latest statement was mailed to 100 Main Street, Any City,  
WA 98121.  
After your payment is received, you will receive a confirmation  
text message at 206-555-0100.  
If you have questions about your bill, AnyCompany Customer Service  
is available by phone at 206-555-0199 or  
email at support@anycompany.com.
```

Fazer upload de um arquivo para um bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista de Buckets, escolha o nome do bucket que você criou na [Etapa 1 \(p. 48\)](#) (por exemplo, **tutorial-bucket**) para carregar seu arquivo.
4. Na guia Objects (Objetos) do bucket, escolha Upload (Fazer upload).
5. Na página Upload (Carregar), em Files and folders (Arquivos e pastas), escolha Add files (Adicionar arquivos).
6. Escolha um arquivo para carregar e, em seguida, escolha Open (Abrir). Por exemplo, você pode carregar o exemplo de arquivo `tutorial.txt` mencionado anteriormente.
7. Escolha Upload (Carregar).

Etapa 3: criar um ponto de acesso do S3

Para usar um ponto de acesso do S3 Object Lambda para acessar e transformar os dados originais, você deve criar um ponto de acesso do S3 e associá-lo ao bucket do S3 criado na [Etapa 1 \(p. 48\)](#). O ponto de acesso deve estar na mesma Região da AWS que os objetos que você deseja transformar.

Mais adiante neste tutorial, você usará esse ponto de acesso como um ponto de acesso de suporte para o ponto de acesso do Object Lambda.

Como criar um ponto de acesso

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Access Points (Pontos de acesso).

3. Na página Access Points (Pontos de acesso), escolha Create access point (Criar ponto de acesso).
4. No campo Access point name (Nome do ponto de acesso), insira o nome (por exemplo, **tutorial-pii-access-point**) para o ponto de acesso.

Para obter mais informações sobre nomenclatura de pontos de acesso, consulte [Regras para nomear pontos de acesso do Amazon S3 \(p. 325\)](#).
5. No campo Bucket name (Nome do bucket), insira o nome do bucket criado na [Etapa 1 \(p. 48\)](#) (por exemplo, **tutorial-bucket**). O S3 anexa o ponto de acesso a este bucket.

(Opcional) Você pode escolher Browse S3 (Procurar S3) para navegar e pesquisar buckets na sua conta. Se você escolher Browse S3 (Procurar S3), selecione o bucket desejado e escolha Choose path (Escolher caminho) para preencher o campo Bucket name (Nome do bucket) com o nome do bucket.
6. Para Network origin (Origem de rede), escolha Internet.

Para obter mais informações sobre origens de rede para pontos de acesso, consulte [Criar pontos de acesso restritos a uma nuvem privada virtual \(p. 327\)](#).
7. Por padrão, todas as configurações de bloqueio de acesso público são habilitadas para seu ponto de acesso. Recomendamos manter a opção Block all public access (Bloquear todo o acesso público) ativada. Para obter mais informações, consulte [Gerenciar o acesso público a pontos de acesso \(p. 329\)](#).
8. Para todas as outras configurações de ponto de acesso, mantenha as configurações padrão.

(Opcional) Você pode modificar as configurações do ponto de acesso para dar suporte ao caso de uso. Para este tutorial, recomendamos manter as configurações padrão.

(Opcional) Se você precisar gerenciar o acesso ao seu ponto de acesso, você pode especificar uma política de ponto de acesso. Para obter mais informações, consulte [Exemplos de política de ponto de acesso \(p. 322\)](#).
9. Selecione Create access point (Criar ponto de acesso).

Etapa 4: configurar e implantar uma função Lambda pré-construída

Para editar dados de PII, configure e implante a função `ComprehendPiiRedactionS3ObjectLambda` do AWS Lambda pré-criada para uso com o ponto de acesso do S3 Object Lambda.

Para configurar e implantar a função Lambda

1. Faça login no AWS Management Console e visualize a função `ComprehendPiiRedactionS3ObjectLambda` no AWS Serverless Application Repository.
2. Para Application settings (Configurações da aplicação), em Application name (Nome da aplicação), mantenha o valor padrão (`ComprehendPiiRedactionS3ObjectLambda`) para este tutorial.

(Opcional) Você pode inserir o nome que deseja dar a esta aplicação. Talvez você queira fazer isso se pretende configurar várias funções Lambda para diferentes necessidades de acesso para o mesmo conjunto de dados compartilhado.
3. Para MaskCharacter, mantenha o valor padrão (*). O caractere de máscara substitui cada caractere na entidade PII editada.
4. Para MaskMode, mantenha o valor padrão (MASK). O valor MaskMode especifica se a entidade de PII é editada com o caractere MASK ou com o valor `PII_ENTITY_TYPE`.
5. Para editar os tipos de dados especificados, para `PiiEntityTypes`, mantenha o valor padrão ALL. O valor `PiiEntityTypes` especifica os tipos de entidade de PII a serem considerados para edição.

Para obter mais informações sobre a lista de tipos de entidade PII compatíveis, consulte [Detectar informações de identificação pessoal \(PII\)](#) no Guia do desenvolvedor do Amazon Comprehend.

6. Mantenha as configurações restantes definidas conforme os padrões.
(Opcional) Se quiser definir configurações adicionais para o caso de uso específico, consulte a seção Readme file (Arquivo Leiamer) no lado esquerdo da página.
7. Marque a caixa de seleção próxima a I acknowledge that this app creates custom IAM roles (Reconheço que esta aplicação cria funções personalizadas do IAM).
8. Escolha **Implantar**.
9. Na página da nova aplicação, em Resources (Recursos), escolha o Logical ID (ID lógico) da função Lambda que você implantou para revisar a função na página da função Lambda.

Etapa 5: criar um ponto de acesso do Object Lambda do S3

Um ponto de acesso do S3 Object Lambda fornece a flexibilidade de invocar uma função Lambda diretamente de uma solicitação do S3 GET para que a função possa editar dados PII recuperados de um ponto de acesso do S3. Ao criar e configurar um ponto de acesso do S3 Object Lambda, você deve especificar a função Lambda de redação para invocar e fornecer o contexto de evento no formato JSON como parâmetros personalizados para o Lambda usar.

O contexto do evento fornece informações sobre a solicitação que está sendo feita no evento passado do S3 Object Lambda para o Lambda. Para obter mais informações sobre todos os campos no contexto do evento, consulte [Formato e uso de contexto de evento \(p. 313\)](#).

Criar um ponto de acesso do S3 Object Lambda

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Object Lambda access points (Pontos de acesso do Object Lambda).
3. Na página Object Lambda Access Points (Pontos de acesso do Object Lambda), escolha Create Object Lambda Access Point (Criar ponto de acesso do Object Lambda).
4. Para Object Lambda Access Point name (Nome do ponto de acesso do Object Lambda), informe o nome que você deseja usar para o ponto de acesso do Object Lambda (por exemplo, **tutorial-pii-object-lambda-accesspoint**).
5. Para Supporting Access Point (Ponto de acesso de suporte), informe ou procure o ponto de acesso padrão criado na [Etapa 3 \(p. 49\)](#) (por exemplo, **tutorial-pii-access-point**) e, em seguida, escolha Choose supporting Access Point (Escolher ponto de acesso de suporte).
6. Para Invoke Lambda function (Chamar função Lambda), você pode escolher qualquer uma das duas opções a seguir para este tutorial.
 - Escolha Choose from functions in your account (Escolher das funções de sua conta) e escolha a função Lambda que você implantou na [Etapa 4 \(p. 50\)](#) (por exemplo, **serverlessrepo-ComprehendPiiRedactionS3ObjectLambda**) da lista suspensa Lambda function (Função Lambda).
 - Escolha Enter ARN (Inserir ARN) e depois informe o Nome do recurso da Amazon (ARN) da função Lambda que você criou na [Etapa 4 \(p. 50\)](#).
7. Para Lambda function version (Versão da função Lambda), escolha \$LATEST (a versão mais recente da função Lambda que você implantou na [Etapa 4 \(p. 50\)](#)).
8. (Opcional) Se precisar da função Lambda para reconhecer e processar solicitações GET com cabeçalhos de intervalo e número de peça, selecione Lambda function supports requests using range

(A função Lambda suporta solicitações usando intervalo) e Lambda function supports requests using part numbers (A função Lambda suporta solicitações usando números de parte). Caso contrário, desmarque essas duas caixas de seleção.

Para obter mais informações sobre como usar números de intervalo ou de parte com o S3 Object Lambda, consulte [Trabalhar com cabeçalhos Range e partNumber \(p. 311\)](#).

9. (Opcional) Em Payload - optional (Carga útil - opcional), adicione um texto JSON para fornecer informações adicionais à sua função Lambda.

Uma carga útil é um texto JSON opcional que você pode fornecer à sua função Lambda como entrada para todas as chamadas provenientes de um ponto de acesso do S3 Object Lambda específico.

Para personalizar os comportamentos de vários pontos de acesso do Object Lambda que chamam a mesma função Lambda, você pode configurar cargas úteis com diferentes parâmetros, estendendo, assim, a flexibilidade da sua função Lambda.

Para obter mais informações sobre carga útil, consulte [Formato e uso de contexto de evento \(p. 313\)](#).

10. (Opcional) Para Request metrics - optional (Métricas de solicitação - opcional), escolha Disable (Desabilitar) ou Enable (Habilitar) para adicionar o monitoramento do Amazon S3 ao seu ponto de acesso do Object Lambda. As métricas de solicitação são cobradas na taxa padrão do Amazon CloudWatch. Para obter mais informações, consulte [Preço do CloudWatch](#).
11. Em Object Lambda Access Point policy - optional (Política do ponto de acesso do Object Lambda - opcional), mantenha a configuração padrão.

(Opcional) Você pode definir uma política de recursos. Essa política de recursos concede permissão da API GetObject para usar o ponto de acesso do Object Lambda especificado.
12. Mantenha as configurações restantes definidas conforme os padrões e escolha Create Object Lambda Access Point (Criar ponto de acesso do Object Lambda).

Etapa 6: usar o ponto de acesso do S3 Object Lambda para recuperar o arquivo editado

Agora, o S3 Object Lambda está pronto para editar dados de PII do seu arquivo original.

Para usar o ponto de acesso do S3 Object Lambda para recuperar o arquivo editado

Quando você solicita para recuperar um arquivo por meio do ponto de acesso do S3 Object Lambda, você faz uma chamada de API GetObject para o S3 Object Lambda. O S3 Object Lambda chama a função Lambda para editar seus dados de PII e retorna os dados transformados como a resposta à chamada de API GetObject do S3 padrão.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Object Lambda access points (Pontos de acesso do Object Lambda).
3. Na página Object Lambda Access Points (Pontos de acesso do Object Lambda), escolha o ponto de acesso do S3 Object Lambda que você criou na [Etapa 5 \(p. 51\)](#) (por exemplo, **tutorial-pii-object-lambda-accesspoint**).
4. Na guia Objects (Objetos) de seu ponto de acesso do S3 Object Lambda, selecione o arquivo que tem o mesmo nome (por exemplo, **tutorial.txt**) daquele que você carregou no bucket do S3 na [Etapa 2 \(p. 49\)](#).

Esse arquivo deve conter todos os dados transformados.

5. Para exibir os dados transformados, escolha Open (Abrir) ou Download (Baixar).

Você deve ser capaz de ver o arquivo editado, conforme mostrado no exemplo a seguir.

```
Hello *****. Your AnyCompany Financial Services,  
LLC credit card account ***** has a minimum payment  
of $24.53 that is due by *****. Based on your autopay settings,  
we will withdraw your payment on the due date from your  
bank account ***** with the routing number *****.  
  
Your latest statement was mailed to *****.  
After your payment is received, you will receive a confirmation  
text message at *****.  
If you have questions about your bill, AnyCompany Customer Service  
is available by phone at ***** or  
email at *****.
```

Etapa 7: limpeza

Se você editou seus dados por meio do S3 Object Lambda apenas como um exercício de aprendizado, exclua os recursos da AWS que você alocou para que não haja mais encargos.

Subetapas

- [Excluir o ponto de acesso do Object Lambda \(p. 53\)](#)
- [Exclua o ponto de acesso do S3 \(p. 53\)](#)
- [Excluir a função Lambda \(p. 54\)](#)
- [Excluir o grupo de logs do CloudWatch \(p. 54\)](#)
- [Exclua o arquivo original no bucket de origem do S3 \(p. 54\)](#)
- [Exclua o bucket de origem do S3 \(p. 54\)](#)
- [Exclua a função do IAM para a função Lambda \(p. 55\)](#)
- [Exclua a política gerenciada pelo cliente para o usuário do IAM \(p. 55\)](#)
- [Excluir o usuário do IAM \(p. 55\)](#)

Excluir o ponto de acesso do Object Lambda

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Object Lambda access points (Pontos de acesso do Object Lambda).
3. Na página Object Lambda Access Points (Pontos de acesso do Object Lambda), escolha o botão de opção à esquerda do ponto de acesso do S3 Object Lambda que você criou na [Etapa 5 \(p. 51\)](#) (por exemplo, **tutorial-pii-object-lambda-accesspoint**).
4. Escolha Delete (Excluir).
5. Confirme se deseja excluir o ponto de acesso do Lambda, inserindo o nome no campo de texto exibido e escolha Delete (Excluir).

Exclua o ponto de acesso do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Access Points (Pontos de acesso).

3. Navegue até o ponto de acesso que você criou na [Etapa 3 \(p. 49\)](#) (por exemplo, **tutorial-pii-access-point**) e escolha o botão de opção ao lado do nome do ponto de acesso.
4. Escolha Delete (Excluir).
5. Confirme se deseja excluir o ponto de acesso inserindo o nome no campo de texto exibido e escolha Delete (Excluir).

Excluir a função Lambda

1. No console do AWS Lambda em <https://console.aws.amazon.com/lambda/>, escolha Functions (Funções) no painel de navegação à esquerda.
2. Escolha a função que você criou na [Etapa 4 \(p. 50\)](#) (por exemplo, **serverlessrepo-ComprehendPiiRedactionS3ObjectLambda**).
3. Escolha Actions (Ações) e, em seguida, escolha Delete (Excluir).
4. Na caixa de diálogo Delete function (Excluir função), escolha Delete (Excluir).

Excluir o grupo de logs do CloudWatch

1. Abra o console do CloudWatch em <https://console.aws.amazon.com/cloudwatch/>.
2. No painel de navegação esquerdo, escolha Log groups (Grupos de log).
3. Localize o grupo de logs cujo nome termina com a função Lambda que você criou na [Etapa 4 \(p. 50\)](#) (por exemplo, **serverlessrepo-ComprehendPiiRedactionS3ObjectLambda**).
4. Escolha Actions (Ações) e Delete log group(s) (Excluir grupo(s) de log).
5. Na caixa de diálogo Delete log group(s) (Excluir grupo(s) de logs), escolha Delete (Excluir).

Exclua o arquivo original no bucket de origem do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Bucket name (Nome do bucket), escolha o nome do bucket para o qual você carregou o arquivo original na [Etapa 2 \(p. 49\)](#) (por exemplo, **tutorial-bucket**).
4. Marque a caixa de seleção à esquerda do nome do objeto que você deseja excluir (por exemplo, **tutorial.txt**).
5. Escolha Delete (Excluir).
6. Na página Delete objects (Excluir objetos) na seção Permanently delete objects? (Excluir objetos permanentemente?), confirme se deseja excluir este objeto informando **permanently delete** na caixa de texto.
7. Escolha Delete objects (Excluir objetos).

Exclua o bucket de origem do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista de Buckets, escolha o botão de opção ao lado do nome do bucket que você criou na [Etapa 1 \(p. 48\)](#) (por exemplo, **tutorial-bucket**).
4. Escolha Delete (Excluir).

5. Na página Delete bucket (Excluir bucket), confirme se deseja excluir o bucket inserindo o nome do bucket no campo de texto e escolha Delete bucket (Excluir bucket).

Exclua a função do IAM para a função Lambda

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação esquerdo, escolha Roles (Funções) e marque a caixa de seleção ao lado do nome da função que você deseja excluir. O nome da função começa com o nome da função Lambda que você implantou na [Etapa 4 \(p. 50\)](#) (por exemplo, **serverlessrepo-ComprehendPiiRedactionS3ObjectLambda**).
3. Escolha Delete (Excluir).
4. Na caixa de diálogo Delete (Excluir), informe o nome da função no campo de entrada de texto para confirmar a exclusão. Em seguida, selecione Delete (Excluir).

Exclua a política gerenciada pelo cliente para o usuário do IAM

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação à esquerda, escolha Policies (Políticas).
3. Na página Policies (Políticas), insira o nome da política gerenciada pelo cliente que você criou em [Prerequisites \(p. 47\)](#) (Pré-requisitos) (por exemplo, **tutorial-serverless-application-repository**) na caixa de pesquisa para filtrar a lista de políticas. Selecione o botão de opção ao lado do nome da política que você deseja excluir.
4. Escolha Actions (Ações) e, em seguida, escolha Delete (Excluir).
5. Confirme se deseja excluir esta política, inserindo seu nome no campo de texto exibido e escolha Delete (Excluir).

Excluir o usuário do IAM

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação esquerdo, escolha Users (Usuários) e marque a caixa de seleção ao lado do nome do usuário que você deseja excluir.
3. Na parte superior da página, escolha Delete (Excluir).
4. Na caixa de diálogo Delete **user name?** (Excluir nome de usuário?), insira o nome de usuário no campo de entrada de texto para confirmar a exclusão do usuário. Escolha Delete (Excluir).

Next steps (Próximas etapas)

Depois de concluir este tutorial, você pode explorar ainda mais os seguintes casos de uso relacionados:

- Você pode criar vários pontos de acesso do S3 Object Lambda e habilitá-los com funções Lambda pré-criadas que são configuradas de forma diferente para editar tipos específicos de PII, dependendo das necessidades de negócios dos assessores de dados.

Cada tipo de usuário assume uma função do IAM e só tem acesso a um ponto de acesso do S3 Object Lambda (gerenciado por meio de políticas do IAM). Em seguida, anexe cada função Lambda **ComprehendPiiRedactionS3ObjectLambda** configurada para um caso de uso de edição diferente para um ponto de acesso do S3 Object Lambda diferente. Para cada ponto de acesso do S3 Object

Lambda, você pode ter um ponto de acesso do S3 de suporte para ler dados de um bucket do S3 que armazena o conjunto de dados compartilhado.

Para obter mais informações sobre como criar uma política de bucket do S3 que conceda aos usuários a leitura do bucket somente por meio dos pontos de acesso do S3, consulte [Configurar políticas do IAM para uso de pontos de acesso \(p. 320\)](#).

Para obter mais informações sobre como conceder permissão de usuário para acessar a função Lambda, o ponto de acesso S3 e o ponto de acesso do S3 Object Lambda, consulte [Configurando políticas do IAM para pontos de acesso do Object Lambda \(p. 299\)](#).

- Você pode criar sua própria função Lambda e usar o S3 Object Lambda com sua função Lambda personalizada para atender às suas necessidades de dados específicas.

Por exemplo, para explorar vários valores de dados, você pode usar o S3 Object Lambda e sua própria função Lambda que usa [Recursos do Amazon Comprehend](#), como reconhecimento de entidade, reconhecimento de frase-chave, análise de sentimento e classificação de documentos, para processar dados. Você também pode usar o S3 Object Lambda junto com o [Amazon Comprehend Medical](#), um serviço de PNL qualificado para HIPAA, para analisar e extrair dados com reconhecimento de contexto.

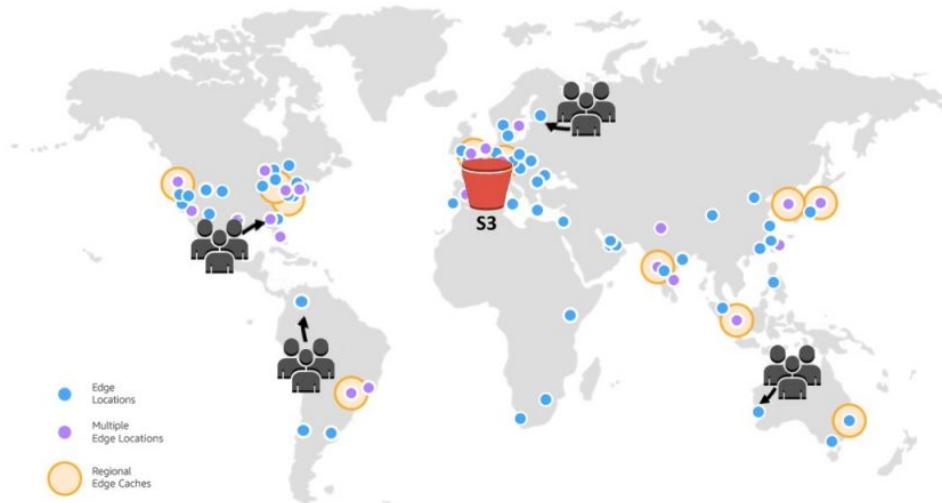
Para obter mais informações sobre como transformar dados com o S3 Object Lambda e sua própria função Lambda, consulte [Tutorial: Como transformar dados para sua aplicação com o S3 Object Lambda \(p. 30\)](#).

Tutorial: Hospedagem de transmissão sob demanda com o Amazon S3, Amazon CloudFront e Amazon Route 53

Você pode usar o Amazon S3 com o Amazon CloudFront para hospedar vídeos para visualização sob demanda, com segurança e escalabilidade. Transmissão de vídeo sob demanda (VOD - Video On Demand) significa que o conteúdo do vídeo é armazenado em um servidor e os espectadores podem assisti-lo a qualquer momento.

O CloudFront é um serviço de rede de entrega de conteúdo (CDN) rápido, altamente seguro e programável. O CloudFront pode fornecer seu conteúdo de modo seguro por HTTPS a partir de todos os locais da borda do CloudFront. Para mais informações sobre o CloudFront, consulte [O que é o Amazon CloudFront?](#) no Guia do desenvolvedor do Amazon CloudFront.

O cache do CloudFront reduz o número de solicitações às quais seu servidor de origem deve responder diretamente. Quando um espectador (usuário final) solicita um vídeo que você veicula com o CloudFront, a solicitação é encaminhada para o local da borda mais próximo de onde o visualizador está localizado. O CloudFront veicula o vídeo de seu cache, recuperando-o do bucket do S3 somente se ele ainda não estiver armazenado em cache. Esse recurso de gerenciamento de cache acelera a entrega de seu vídeo aos espectadores globalmente, com baixa latência, alto throughput e altas velocidades de transferência. Para mais informações sobre o gerenciamento do cache do CloudFront, consulte [Otimizar o armazenamento em cache e a disponibilidade](#) no Guia do desenvolvedor do Amazon CloudFront.



Objetivo

Neste tutorial, você configura um bucket do S3 para hospedar transmissão de vídeo sob demanda usando o CloudFront para entrega e o Amazon Route 53 para gerenciamento de Sistema de Nomes de Domínio (DNS) e de domínio personalizado.

Tópicos

- [Pré-requisitos: registrar e configurar um domínio personalizado com o Route 53 \(p. 57\)](#)
- [Etapa 1: criar um bucket do S3 \(p. 58\)](#)
- [Etapa 2: carregar um vídeo no bucket do S3 \(p. 59\)](#)
- [Etapa 3: Criar uma identidade do acesso de origem do CloudFront \(p. 59\)](#)
- [Etapa 4: criar uma distribuição do CloudFront \(p. 60\)](#)
- [Etapa 5: Acessar o vídeo por meio da distribuição do CloudFront \(p. 61\)](#)
- [Etapa 6: Configurar sua distribuição do CloudFront para usar o seu nome de domínio personalizado \(p. 62\)](#)
- [Etapa 7: acessar o vídeo do S3 por meio da distribuição do CloudFront com o nome de domínio personalizado \(p. 66\)](#)
- [\(Opcional\) Etapa 8: exibir dados sobre solicitações recebidas pela distribuição do CloudFront \(p. 66\)](#)
- [Etapa 9: limpeza \(p. 67\)](#)
- [Next steps \(Próximas etapas\) \(p. 70\)](#)

Pré-requisitos: registrar e configurar um domínio personalizado com o Route 53

Antes de iniciar este tutorial, você deve se inscrever e configurar um domínio personalizado (por exemplo, **example.com**) no Route 53 para poder configurar sua distribuição do CloudFront para usar um nome de domínio personalizado depois.

Sem um nome de domínio personalizado, seu vídeo do S3 é acessível ao público e hospedado por meio do CloudFront em um URL semelhante a este:

`https://CloudFront distribution domain name/Path to an S3 video`

Por exemplo, `https://d111111abcdef8.cloudfront.net/sample.mp4`.

Depois de configurar sua distribuição do CloudFront para usar um nome de domínio personalizado configurado com o Route 53, seu vídeo do S3 fica publicamente acessível e hospedado pelo CloudFront em um URL semelhante a este:

`https://CloudFront distribution alternate domain name/Path to an S3 video`

Por exemplo, `https://www.example.com/sample.mp4`. Um nome de domínio personalizado é mais simples e intuitivo para os espectadores usarem.

Para registrar um nome de domínio personalizado, consulte [Registro de um novo nome de domínio usando o Route 53](#) no Guia do desenvolvedor do Amazon Route 53.

Quando você registra um nome de domínio com o Route 53, ele cria a zona hospedada que você usará posteriormente neste tutorial. Esta zona hospedada é onde você armazena informações sobre como encaminhar o tráfego para seu domínio, por exemplo, para uma instância do Amazon EC2 ou uma distribuição do CloudFront.

Há tarifas associadas ao registro de domínios, à sua zona hospedada e às consultas de DNS recebidas pelo seu domínio. Para obter mais informações, consulte [Definição de preço do Amazon Route 53](#).

Note

Ao registrar um domínio, você gasta dinheiro imediatamente e é irreversível. Você pode escolher não renovar automaticamente o domínio, mas paga antecipadamente e adquire o domínio por um ano). Para obter mais informações, consulte [Registrar um novo domínio](#) no Guia do desenvolvedor do Amazon Route 53.

Etapa 1: criar um bucket do S3

Crie um bucket para armazenar o vídeo original que você pretende transmitir.

Para criar um bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Selecione Create bucket (Criar bucket).

A página Create bucket (Criar bucket) é aberta.

4. Em Bucket name (Nome do bucket), insira um nome para o seu bucket (por exemplo, **tutorial-bucket**).

Para obter mais informações sobre como nomear buckets no Amazon S3, consulte [Regras de nomeação de bucket \(p. 127\)](#).

5. Em Region (Região), escolha a Região da AWS onde deseja que o bucket resida.

Se possível, você deve escolher a região mais próxima da maioria dos seus espectadores. Para obter mais informações sobre a região do bucket, consulte [Visão geral dos buckets \(p. 123\)](#).

6. Para Block Public Access settings for this bucket (Configurações de acesso de bloqueio público para este bucket), mantenha as configurações padrão (Block allpublic access (Bloquear todo acesso público) está habilitado).

Mesmo com Block all public access (Bloquear todo acesso público) habilitado, os espectadores ainda podem acessar o vídeo carregado por meio do CloudFront. Esse recurso é uma grande vantagem do uso do CloudFront para hospedar um vídeo armazenado no S3.

Recomendamos que você mantenha todas as configurações ativadas, a menos que precise desabilitar uma ou mais delas para seu caso de uso. Para obter mais informações sobre como bloquear o acesso público, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

7. Mantenha as configurações restantes definidas conforme os padrões.
(Opcional) Se quiser definir configurações de bucket adicionais para o caso de uso específico, consulte [Criação de um bucket \(p. 129\)](#).
8. Selecione Create bucket (Criar bucket).

Etapa 2: carregar um vídeo no bucket do S3

O procedimento a seguir descreve como carregar um arquivo de vídeo em um bucket do S3 usando o console. Se você for carregar muitos arquivos de vídeo grandes para o S3, pode ser interessante usar o [Amazon S3 Transfer Acceleration](#) para configurar transferências de arquivos rápidas e seguras. O Transfer Acceleration pode acelerar o carregamento de vídeo no seu bucket do S3 para transferências de longa distância de vídeos maiores. Para obter mais informações, consulte [Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration \(p. 147\)](#).

Para carregar o arquivo para o bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista de Buckets, escolha o nome do bucket que você criou na [Etapa 1 \(p. 58\)](#) (por exemplo, **tutorial-bucket**) para carregar seu arquivo.
4. Na guia Objects (Objetos) do bucket, escolha Upload (Fazer upload).
5. Na página Upload (Carregar), em Files and folders (Arquivos e pastas), escolha Add files (Adicionar arquivos).
6. Escolha um arquivo para carregar e, em seguida, escolha Open (Abrir).

Por exemplo, você pode carregar um arquivo de vídeo chamado `sample.mp4`.

7. Escolha Upload (Carregar).

Etapa 3: Criar uma identidade do acesso de origem do CloudFront

Para restringir o acesso direto ao vídeo a partir do seu bucket do S3, crie um usuário especial do CloudFront denominado identidade do acesso de origem (OAI). Você vai associar a OAI à sua distribuição mais adiante neste tutorial. Usando uma OAI, você garante que os espectadores não possam ignorar o CloudFront e obter o vídeo diretamente do bucket do S3. Somente a OAI do CloudFront pode acessar o arquivo no bucket do S3. Para obter mais informações, consulte [Restrição de acesso ao conteúdo do Amazon S3 usando uma OAI](#) no Guia do desenvolvedor do Amazon CloudFront.

Para criar uma OAI do CloudFront

1. Faça login no AWS Management Console e abra o console do CloudFront em <https://console.aws.amazon.com/cloudfront/v3/home>.

2. Na seção Security (Segurança) do painel de navegação esquerdo, escolha Origin access identities (Identidades do acesso de origem).
3. Selecione Create origin access identity (Criar identidade do acesso de origem).
4. Insira um nome (por exemplo, **S3-OAI**) para a nova identidade do acesso de origem.
5. Escolha Create (Criar OpsItem).

Etapa 4: criar uma distribuição do CloudFront

Para usar o CloudFront para oferecer e distribuir o vídeo em seu bucket do S3, você deve criar uma distribuição do CloudFront.

Subetapas

- [Crie uma distribuição do CloudFront \(p. 60\)](#)
- [Revise a política de bucket \(p. 61\)](#)

Crie uma distribuição do CloudFront

1. Faça login no AWS Management Console e abra o console do CloudFront em <https://console.aws.amazon.com/cloudfront/v3/home>.
2. No painel de navegação esquerdo, escolha Distributions (Distribuições).
3. Escolha Create distribution (Criar distribuição).
4. Na seção Origin (Origem), para Origin domain (Domínio de origem), escolha o nome de domínio de sua origem do S3 que começa com o nome do bucket do S3 criado na [Etapa 1 \(p. 58\)](#) (por exemplo, **tutorial-bucket**).
5. Para S3 bucket access (Acesso ao bucket do S3), escolha Yes use OAI (bucket can restrict access to only CloudFront) (Sim, use o OAI [o bucket pode restringir o acesso apenas ao CloudFront]).
6. Em Origin access identity (Identidade do acesso de origem), escolha a identidade do acesso de origem que você criou na [Etapa 3 \(p. 59\)](#) (por exemplo, **S3-OAI**).
7. Em Bucket policy (Política de bucket), escolha Yes, update the bucket policy (Sim, atualizar a política de bucket).
8. Para Default cache behavior (Comportamento do cache padrão), em Viewer protocol policy (Política de protocolo do espectador), escolha Redirect HTTP to HTTPS (Redirecionar HTTP para HTTPS).

Quando você escolhe este recurso, as solicitações de HTTP são automaticamente redirecionadas para HTTPS para proteger seu site e proteger os dados de seus espectadores.

9. Para as outras configurações na seção Default cache behaviors (Configurações de comportamento de cache padrão), mantenha os valores padrão.

(Opcional) Você pode controlar o tempo de permanência dos arquivos em um cache do CloudFront antes que o CloudFront encaminhe outra solicitação para a origem. A diminuição da duração permite fornecer conteúdo dinâmico. Aumentar a duração significa que os espectadores obtêm uma melhor performance, pois é mais provável que seus arquivos sejam fornecidos diretamente do cache de borda. Uma duração maior também reduz a carga na origem. Para obter mais informações, consulte [Gerenciamento do tempo que o conteúdo permanece em um cache de borda \(validade\)](#) no Guia do desenvolvedor do Amazon CloudFront.

10. Para as outras seções, mantenha as demais configurações definidas como os padrões.

Para obter mais informações sobre essas opções de configuração, consulte [Valores que você especifica quando cria ou atualiza uma distribuição](#) no Guia do desenvolvedor do Amazon CloudFront.

11. Na parte inferior da página, escolha Create Distribution (Criar distribuição).

12. Na guia General (Geral) da sua distribuição do CloudFront, em Details (Detalhes), o valor da coluna Last modified (Última modificação) para sua distribuição muda de Deploying (Em implantação) para o carimbo de data e hora em que a distribuição foi modificada pela última vez. Normalmente, esse processo leva alguns minutos.

Revise a política de bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista de Buckets, escolha o nome do bucket que você usou antes, como a origem de sua distribuição do CloudFront (por exemplo, **tutorial-bucket**).
4. Escolha a guia Permissions (Permissões).
5. Na seção Bucket policy (Política de bucket), confirme que você vê uma instrução semelhante à seguinte no texto da política de bucket:

```
{  
    "Version": "2008-10-17",  
    "Id": "PolicyForCloudFrontPrivateContent",  
    "Statement": [  
        {  
            "Sid": "1",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access  
Identity EH1HDMB1FH2TC"  
            },  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws:s3:::tutorial-bucket/*"  
        }  
    ]  
}
```

Esta é a instrução que sua distribuição do CloudFront adicionou à sua política de bucket quando você escolheu Yes, update the bucket policy (Sim, atualizar a política de bucket) antes.

Esta atualização de política de bucket indica que você configurou com êxito a distribuição do CloudFront para restringir o acesso ao bucket do S3. Devido a essa restrição, os objetos no bucket só podem ser acessados por meio da distribuição do CloudFront.

Etapa 5: Acessar o vídeo por meio da distribuição do CloudFront

Agora, o CloudFront pode veicular o vídeo armazenado no seu bucket do S3. Para acessar seu vídeo por meio do CloudFront, você deve combinar seu nome de domínio de distribuição do CloudFront com o caminho para o vídeo no bucket do S3.

Para criar um URL para o vídeo do S3 usando o nome de domínio de distribuição do CloudFront

1. Faça login no AWS Management Console e abra o console do CloudFront em <https://console.aws.amazon.com/cloudfront/v3/home>.
2. No painel de navegação esquerdo, escolha Distributions (Distribuições).
3. Para obter o nome do domínio de distribuição, faça o seguinte:

- a. Na coluna Origins (Origens), encontre a distribuição correta do CloudFront localizando seu nome de origem, que começa com o bucket do S3 que você criou na [Etapa 1 \(p. 58\)](#) (por exemplo, **tutorial-bucket**).
- b. Depois de encontrar a distribuição na lista, alargue a coluna Domain name (Nome de domínio) para copiar o valor do nome do domínio para sua distribuição do CloudFront.
4. Em uma nova guia do navegador, cole o nome do domínio de distribuição que você copiou.
5. Retorne à guia anterior do navegador e abra o console do S3 em <https://console.aws.amazon.com/s3/>.
6. No painel de navegação à esquerda, escolha Buckets.
7. Na lista de Buckets, escolha o nome do bucket que você criou na [Etapa 1 \(p. 58\)](#) (por exemplo, **tutorial-bucket**).
8. Na lista Objects (Objetos), escolha o nome do vídeo que você carregou na [Etapa 2 \(p. 59\)](#) (por exemplo, **sample.mp4**).
9. Na página de detalhes do objeto, na seção Object overview (Visão geral do objeto), copie o valor da Key (Chave). Esse valor é o caminho para o objeto de vídeo carregado no bucket do S3.
10. Retorne à guia do navegador onde você colou anteriormente o nome do domínio de distribuição, insira uma barra (/) após o nome do domínio de distribuição e, em seguida, cole o caminho do vídeo que você copiou antes (por exemplo, **sample.mp4**).

Agora, seu vídeo do S3 está acessível ao público e hospedado por meio do CloudFront em um URL semelhante a este:

`https://CloudFront distribution domain name/Path to the S3 video`

Substitua *CloudFront distribution domain name* (Nome de domínio de distribuição do CloudFront) e *Path to the S3 video* (Caminho para o vídeo S3) pelos valores apropriados. O URL do exemplo é: <https://d111111abcdef8.cloudfront.net/sample.mp4>.

Etapa 6: Configurar sua distribuição do CloudFront para usar o seu nome de domínio personalizado

Para usar seu próprio nome de domínio em vez do nome de domínio do CloudFront no URL para acessar o vídeo do S3, adicione um nome de domínio alternativo à distribuição do CloudFront.

Subetapas

- [Solicitar um certificado SSL \(p. 62\)](#)
- [Adicione um nome de domínio alternativo à distribuição do CloudFront \(p. 63\)](#)
- [Crie um registro do DNS para encaminhar o tráfego do seu nome de domínio alternativo para o nome de domínio da distribuição do CloudFront \(p. 64\)](#)
- [Verifique se o IPv6 está habilitado para sua distribuição e crie outro registro do DNS, se necessário \(p. 65\)](#)

Solicitar um certificado SSL

Para permitir que seus espectadores usem HTTPS e seu nome de domínio personalizado no URL da transmissão de vídeo, use a AWS Certificate Manager (ACM) para solicitar um certificado Secure Sockets Layer (SSL). O certificado SSL estabelece uma conexão de rede criptografada com o site.

1. Faça login no AWS Management Console e abra o console do ACM em <https://console.aws.amazon.com/acm/>.

2. Se a página introdutória for exibida, em Provision certificates (Certificados de provisão), escolha Get Started (Conceitos básicos).
3. Na página Request a certificate (Solicitar um certificado), escolha Request a public certificate (Solicitar um certificado público) e depois Request a certificate (Solicitar um certificado).
4. Em Add domain names (Adicionar nomes de domínios) (insira o nome do domínio totalmente qualificado do site que você deseja proteger usando um certificado SSL/TLS. Use um asterisco (*) para solicitar um certificado curinga, que protege vários sites no mesmo domínio. Para este tutorial, insira * e o nome de domínio personalizado que você configurou em [Pré-requisitos \(p. 57\)](#)). Para este exemplo, insira *.example.com e escolha Next (Avançar).

Para obter mais informações, consulte [Solicitar um certificado público do ACM \(console\)](#) no Guia do usuário do AWS Certificate Manager.

5. Na página Select validation method (Selecionar método de validação), escolha DNS validation (Validação de DNS). Em seguida, escolha Next (Próximo).

Se você puder editar sua configuração de DNS, recomendamos usar a validação de domínio de DNS, em vez da validação de e-mail. A validação de DNS tem vários benefícios em relação à validação de e-mail. Para obter mais informações, consulte [Opção 1: Validação de DNS](#) no .Manual do usuário do AWS Certificate Manager.

6. (Opcional) Na página Add tags (Adicionar etiquetas), marque seu certificado com metadados.
7. Escolha Review.
8. Na página Review (Revisão), verifique se as informações em Domain name (Nome de domínio) e Validation method (Método de validação) estão corretas. Escolha Confirm and request (Confirmar e solicitar).

A página Validation (Validação) mostra que sua solicitação está sendo processada e que o domínio de certificado está sendo validado. O certificado que aguarda validação está no estado Pending validation (Validação pendente).

9. Na página Validation (Validação), escolha a seta para baixo à esquerda do seu nome de domínio personalizado e escolha Create record in Route 53 (Criar registro no Route 53) para validar que você é o proprietário do domínio por meio do DNS.

Isso adiciona um registro CNAME fornecido pelo AWS Certificate Manager na configuração do DNS.

10. Na caixa de diálogo Create record in Route 53 (Criar registro no Route 53), escolha Create (Criar).

A página Validation (Validação) agora deve exibir uma notificação de status de Success (Êxito) na parte inferior.

11. Selecione Continue (Continuar) para visualizar a página da lista Certificates (Certificados).

O Status de seu novo certificado muda de Pending validation (Validação pendente) para Issued (Emitida) em até 30 minutos.

Adicione um nome de domínio alternativo à distribuição do CloudFront

1. Faça login no AWS Management Console e abra o console do CloudFront em <https://console.aws.amazon.com/cloudfront/v3/home>.
2. No painel de navegação esquerdo, escolha Distributions (Distribuições).
3. Escolha o ID da distribuição que você criou na [Etapa 4 \(p. 59\)](#).
4. Na guia General (Geral), vá para a seção Settings (Configurações) e escolha Edit (Editar).
5. Na página Edit settings (Editar configurações), para Alternate domain name (CNAME) - optional (Nome de domínio alternativo (CNAME) - opcional), escolha Add item (Adicionar item) para adicionar

os nomes de domínio personalizados que você deseja usar no URL para o vídeo do S3 fornecido por essa distribuição do CloudFront.

Neste tutorial, por exemplo, se você quiser encaminhar o tráfego para um subdomínio, como www.example.com, insira o nome do subdomínio (www) com o nome do domínio (example.com). Especificamente, insira www.example.com.

Note

O nome de domínio alternativo (CNAME) que você adicionar deve ser coberto pelo certificado SSL que você anexou anteriormente à sua distribuição do CloudFront.

6. Para Custom SSL certificate optional, (Certificado SSL personalizado - opcional), escolha o certificado SSL que você solicitou antes (por exemplo, *.example.com).

Note

Se você não vir o certificado SSL imediatamente após solicitá-lo, aguarde 30 minutos, e atualize a lista até que o certificado SSL esteja disponível para você selecionar.

7. Mantenha as configurações restantes definidas conforme os padrões. Selecione Save changes.
8. Na guia General (Geral) para a distribuição, aguarde o valor de Last modified (Última modificação) mudar de Deploying (Em implantação) para o carimbo de data ehora em que a distribuição foi modificada pela última vez.

Crie um registro do DNS para encaminhar o tráfego do seu nome de domínio alternativo para o nome de domínio da distribuição do CloudFront

1. Faça login no AWS Management Console e abra o console do Route 53 em <https://console.aws.amazon.com/route53/>.
2. No painel de navegação, escolha Hosted zones (Zonas hospedadas).
3. Na página Hosted zones (Zonas hospedadas), escolha o nome da zona hospedada que o Route 53 criou para você em [Pré-requisitos \(p. 57\)](#) (por exemplo, example.com).
4. Selecione Create record (Criar registro) e, em seguida, use Quick create record (Criação rápida de registro).
5. Para Record name (Nome de registro), mantenha o valor do nome do registro igual ao nome de domínio alternativo da distribuição do CloudFront que você adicionou antes.

Neste tutorial, para encaminhar o tráfego para um subdomínio, como www.example.com, insira o nome do subdomínio sem o nome do domínio. Por exemplo, insira somente www no campo de texto antes do seu nome de domínio personalizado.

6. Em Record type (Tipo de registro), selecione A - Routes traffic to an IPv4 address and some AWS resources (A: encaminhar o tráfego para um endereço IPv4 e alguns recursos da AWS).
7. Para Value (Valor), escolha a opção Alias para habilitar o recurso de alias.
8. Em Route traffic to (Encaminhar tráfego para), escolha Alias to CloudFront distribution (Alias para distribuição do CloudFront).
9. Na caixa Choose distribution (Escolher distribuição), escolha o nome de domínio da distribuição do CloudFront que você criou na [Etapa 4 \(p. 60\)](#).

Para localizar o nome de domínio da sua distribuição do CloudFront, faça o seguinte:

- a. Em uma nova guia do navegador, entre no AWS Management Console e abra o console do CloudFront em <https://console.aws.amazon.com/cloudfront/v3/home>.
- b. No painel de navegação esquerdo, escolha Distributions (Distribuições).

- c. Na coluna Origins (Origens), encontre a distribuição correta do CloudFront localizando seu nome de origem, que começa com o bucket do S3 que você criou na [Etapa 1 \(p. 58\)](#) (por exemplo, **tutorial-bucket**).
 - d. Depois de encontrar a distribuição na lista, amplie a coluna Domain name (Nome de domínio) para ver o valor do nome do domínio para sua distribuição do CloudFront.
10. Na página Create record (Criar registro) no console do Route 53, para as demais configurações, mantenha os padrões.
11. Escolha Create records (Criar registros).

Verifique se o IPv6 está habilitado para sua distribuição e crie outro registro do DNS, se necessário

Se o IPv6 estiver habilitado para sua distribuição, você deve criar outro registro do DNS.

1. Para verificar se o IPv6 está habilitado para sua distribuição, faça o seguinte:
 - a. Faça login no AWS Management Console e abra o console do CloudFront em <https://console.aws.amazon.com/cloudfront/v3/home>.
 - b. No painel de navegação esquerdo, escolha Distributions (Distribuições).
 - c. Escolha o ID da distribuição do CloudFront que você criou na [Etapa 4 \(p. 60\)](#).
 - d. Na guia General (Geral), em Settings (Configurações), verifique se IPv6 está definido como Enabled (Habilitado).

Se o IPv6 estiver habilitado para sua distribuição, você deve criar outro registro do DNS.
2. Se o IPv6 estiver habilitado para sua distribuição, faça o seguinte para criar um registro DNS:
 - a. Faça login no AWS Management Console e abra o console do Route 53 em <https://console.aws.amazon.com/route53/>.
 - b. No painel de navegação, escolha Hosted zones (Zonas hospedadas).
 - c. Na página Hosted zones (Zonas hospedadas), escolha o nome da zona hospedada que o Route 53 criou para você em [Pré-requisitos \(p. 57\)](#) (por exemplo, **example.com**).
 - d. Selecione Create record (Criar registro) e, em seguida, use Quick create record (Criação rápida de registro).
 - e. Para Record name (Nome de registro), no campo de texto antes do nome de domínio personalizado, digite o mesmo valor que você digitou quando criou antes o registro do DNS do IPv4. Por exemplo, neste tutorial, para encaminhar o tráfego para o subdomínio **www.example.com**, insira apenas **www**.
 - f. Para Record type (Tipo de registro), escolha AAAA - Routes traffic to an IPv6 address and some AWS resources (AAAA - Encaminha o tráfego para um endereço IPv6 e alguns recursos da AWS).
 - g. Para Value (Valor), escolha a opção Alias para habilitar o recurso de alias.
 - h. Em Route traffic to (Encaminhar tráfego para), escolha Alias to CloudFront distribution (Alias para distribuição do CloudFront).
 - i. Na caixa Choose distribution (Escolher distribuição), escolha o nome de domínio da distribuição do CloudFront que você criou na [Etapa 4 \(p. 60\)](#).
 - j. Mantenha as configurações restantes definidas conforme os padrões.
 - k. Escolha Create records (Criar registros).

Etapa 7: acessar o vídeo do S3 por meio da distribuição do CloudFront com o nome de domínio personalizado

Para acessar o vídeo do S3 usando o URL personalizado, você deve combinar seu nome de domínio alternativo com o caminho para o vídeo no bucket do S3.

Criar um URL personalizado para acessar o vídeo do S3 por meio da distribuição do CloudFront

1. Faça login no AWS Management Console e abra o console do CloudFront em <https://console.aws.amazon.com/cloudfront/v3/home>.
2. No painel de navegação esquerdo, escolha Distributions (Distribuições).
3. Para obter o nome de domínio alternativo da sua distribuição do CloudFront, faça o seguinte:
 - a. Na coluna Origins (Origens), encontre a distribuição correta do CloudFront procurando seu nome de origem, que começa com o nome do bucket do S3 que você criou na [Etapa 1 \(p. 58\)](#) (por exemplo, **tutorial-bucket**).
 - b. Depois de encontrar a distribuição na lista, alargue a coluna Alternate domain names (Nomes de domínio alternativos) para copiar o valor do nome de domínio alternativo de sua distribuição do CloudFront.
4. Em uma nova guia do navegador, cole o nome de domínio alternativo da distribuição do CloudFront.
5. Retorne à guia anterior do navegador e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
6. Encontre o caminho para o seu vídeo S3, como explicado na [Etapa 5 \(p. 61\)](#).
7. Retorne à guia do navegador onde você colou anteriormente o nome de domínio alternativo, insira uma barra (/) e cole o caminho para o vídeo do S3 (por exemplo, **sample.mp4**).

Agora, seu vídeo do S3 está acessível ao público e hospedado por meio do CloudFront em um URL personalizado semelhante a este:

https://CloudFront distribution alternate domain name/Path to the S3 video

Substitua **CloudFront distribution alternate domain name** (Nome alternativo de domínio de distribuição do CloudFront) e **Path to the S3 video** (Caminho para o vídeo S3) pelos valores apropriados. O URL do exemplo é: <https://www.example.com/sample.mp4>.

(Opcional) Etapa 8: exibir dados sobre solicitações recebidas pela distribuição do CloudFront

Para exibir dados sobre solicitações recebidas pela sua distribuição do CloudFront

1. Faça login no AWS Management Console e abra o console do CloudFront em <https://console.aws.amazon.com/cloudfront/v3/home>.
2. No painel de navegação esquerdo, em Reports & analytics (Relatórios e análise), escolha os relatórios no console, que incluem Cache statistics (Estatísticas de cache), Popular Objects (Objetos populares), Top Referrers (Principais indicadores), Usage (Uso) e Viewers (Espectadores).

Você pode filtrar cada painel de relatório. Para obter mais informações, consulte [Relatórios do CloudFront no console](#) no Guia do desenvolvedor do Amazon CloudFront.
3. Para filtrar dados, escolha o ID da distribuição do CloudFront que você criou na [Etapa 4 \(p. 60\)](#).

Etapa 9: limpeza

Se você hospedou um vídeo de transmissão do S3 usando o CloudFront e o Route 53 apenas como um exercício de aprendizado, exclua os recursos da AWS alocados para que não continuar a acumular encargos.

Note

Ao registrar um domínio, você gasta dinheiro imediatamente e é irreversível. Você pode escolher não renovar automaticamente o domínio, mas paga antecipadamente e adquire o domínio por um ano). Para obter mais informações, consulte [Registrar um novo domínio](#) no Guia do desenvolvedor do Amazon Route 53.

Subetapas

- [Exclua a distribuição do CloudFront \(p. 67\)](#)
- [Excluir o registro DNS \(p. 68\)](#)
- [Exclua a zona hospedada de seu domínio personalizado \(p. 68\)](#)
- [Exclua o nome de domínio personalizado do Route 53 \(p. 69\)](#)
- [Exclua o vídeo original no bucket de origem do S3 \(p. 69\)](#)
- [Exclua o bucket de origem do S3 \(p. 70\)](#)

Exclua a distribuição do CloudFront

1. Faça login no AWS Management Console e abra o console do CloudFront em <https://console.aws.amazon.com/cloudfront/v3/home>.
 2. No painel de navegação esquerdo, escolha Distributions (Distribuições).
 3. Na coluna Origins (Origens), encontre a distribuição correta do CloudFront procurando seu nome de origem, que começa com o nome do bucket do S3 que você criou na [Etapa 1 \(p. 58\)](#) (por exemplo, **tutorial-bucket**).
 4. Para excluir a distribuição do CloudFront, você deve primeiro desabilitá-la.
 - Se o valor da coluna Status for Enabled (Habilitado) e o valor de Last modified (Última modificação) for o carimbo de data e hora em que a distribuição foi modificada pela última vez, continue para desabilitar a distribuição antes de excluí-la.
 - Se o valor de Status for Enabled (Habilitado) e o valor de Last modified (Última modificação) for Deploying (Em implantação), aguarde até que Status mude para o carimbo de data/hora em que a distribuição foi modificada pela última vez. Em seguida, continue para desabilitar a distribuição antes de excluí-la.
 5. Para desabilitar a distribuição do CloudFront, faça o seguinte:
 - a. Na lista Distributions (Distribuições), marque a caixa de seleção ao lado do ID da distribuição que você deseja excluir.
 - b. Para desabilitar a distribuição, escolha Disable (Desabilitar) e depois escolha Disable (Desabilitar) para confirmar.
- Se você desabilitar uma distribuição que tenha um nome de domínio alternativo associado a ela, o CloudFront deixará de aceitar o tráfego para esse nome de domínio (como `www.example.com`), mesmo que outra distribuição tenha um nome de domínio alternativo com um caractere curinga (`*`) correspondente ao mesmo domínio (como `*.example.com`).
- c. O valor da coluna Status é imediatamente alterado para Disabled (Desabilitado). Aguarde até que o valor de Last modified (Última modificação) mude de Deploying (Implantação) para o carimbo de data e hora em que a distribuição foi modificada pela última vez.

Como o CloudFront deve propagar essa alteração para todos os locais da borda, pode levar alguns minutos para que a atualização seja concluída e a opção Delete (Excluir) esteja disponível para você excluir a distribuição.

6. Para excluir a distribuição desabilitada, faça o seguinte:
 - a. Marque a caixa de seleção ao lado do ID da distribuição que você deseja excluir.
 - b. Selecione Delete (Excluir) e depois escolha Delete (Excluir) para confirmar.

Excluir o registro DNS

Se você quiser excluir a zona hospedada pública para o domínio (incluindo o registro do DNS), consulte [Exclua a zona hospedada de seu domínio personalizado \(p. 68\)](#) no Guia do desenvolvedor do Amazon Route 53. Se você só deseja excluir o registro do DNS criado na [Etapa 6 \(p. 62\)](#), faça o seguinte:

1. Faça login no AWS Management Console e abra o console do Route 53 em <https://console.aws.amazon.com/route53/>.
2. No painel de navegação, escolha Hosted zones (Zonas hospedadas).
3. Na página Hosted zones (Zonas hospedadas), escolha o nome da zona hospedada que o Route 53 criou para você em [Pré-requisitos \(p. 57\)](#) (por exemplo, `example.com`).
4. Na lista de registros, marque a caixa de seleção ao lado dos registros que você deseja excluir (os registros que você criou na [Etapa 6 \(p. 62\)](#)).

Note

Você não pode excluir registros que têm um valor Type (Tipo) de NS ou SOA.

5. Escolha Delete records (Excluir registros).
6. Para confirmar a exclusão, selecione Delete (Excluir).

As alterações nos registros demoram para serem propagadas até os servidores de DNS do Route 53. Atualmente, a única maneira de verificar se as alterações foram propagadas é usar a [ação da API GetChange](#). As alterações geralmente são propagadas para todos os servidores de nome do Route 53 em até 60 segundos.

Exclua a zona hospedada de seu domínio personalizado

Warning

Se quiser manter o registro do domínio, mas interromper o encaminhamento do tráfego da Internet para seu site ou sua aplicação Web, recomendamos excluir os registros na zona hospedada (conforme descrito na seção anterior), em vez de excluir a zona hospedada.

Se você excluir uma zona hospedada, alguém pode usar o domínio e encaminhar o tráfego para seus próprios recursos usando seu nome de domínio.

Além disso, se você excluir uma zona hospedada, não será possível cancelar a exclusão. É necessário criar uma nova zona hospedada e atualizar os servidores de nome para o registro do domínio, o que pode levar até 48 horas para entrar em vigor.

Se desejar tornar o domínio indisponível na Internet, recomendamos que você transfira seu serviço de DNS para um serviço de DNS gratuito e, em seguida, elimine a zona hospedada do Route 53. Isso impede que futuras consultas DNS sejam incorretamente encaminhadas.

1. Se o domínio estiver registrado no Route 53, consulte [Adição ou alteração de servidores de nome e registros cola de um domínio](#) no Guia do desenvolvedor do Amazon Route 53 para obter informações sobre como substituir servidores de nome do Route 53 por servidores de nome do novo serviço de DNS.

2. Se o domínio estiver registrado com outro registrador, use o método fornecido pelo registrador para alterar os servidores de nome do domínio.

Note

Se você estiver excluindo uma zona hospedada de um subdomínio (www.example.com), não será necessário alterar os servidores de nome do domínio (example.com).

1. Faça login no AWS Management Console e abra o console do Route 53 em <https://console.aws.amazon.com/route53/>.
2. No painel de navegação, escolha Hosted zones (Zonas hospedadas).
3. Na página Hosted zones (Zonas hospedadas), escolha o nome da zona hospedada que você deseja excluir.
4. Na guia Records (Registros) de sua zona hospedada, verifique se a zona hospedada que você deseja excluir contém apenas um registro DNS e um SOA.

Se ele contiver registros adicionais, exclua-os.

Se você criou registros de NS para subdomínios na zona hospedada, exclua esses registros também.

5. Na guia DNSSEC signing (Assinatura de DNSSEC) para sua zona hospedada, desabilite a assinatura DNSSSEC, se ela estiver habilitada. Para obter mais informações, consulte [Desabilitação de assinatura de DNSSEC](#) no Guia do desenvolvedor do Amazon Route 53.
6. Na parte superior da página de detalhes da zona hospedada, selecione Delete zone (Excluir zona).
7. Insira **delete** para confirmar a exclusão e depois escolha Delete (Excluir).

Exclua o nome de domínio personalizado do Route 53

A maioria dos domínios de nível superior (TLDs) permite a exclusão do registro quando ele não é mais necessário. Se você excluir um registro de nome de domínio do Route 53 antes de o registro ser programado para expirar, a AWS não reembolsa a taxa de registro. Para obter mais informações, consulte [Exclusão de um registro de nome de domínio](#) no Guia do desenvolvedor do Amazon Route 53.

Important

Se desejar transferir o domínio entre Contas da AWS ou transferir o domínio para outro registrador, não exclua um domínio esperando poder registrá-lo de novo imediatamente. Em vez disso, consulte a documentação aplicável no Guia do desenvolvedor do Amazon Route 53:

- [Transferência de um domínio para uma Conta da AWS diferente](#)
- [Transferência de um domínio do Amazon Route 53 para outro registrador](#)

Exclua o vídeo original no bucket de origem do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Bucket name (Nome do bucket), escolha o nome do bucket para o qual carregou o vídeo na [Etapa 2 \(p. 59\)](#) (por exemplo, **tutorial-bucket**).
4. Na guia Objects (Objetos), marque a caixa de seleção ao lado do nome do objeto que você deseja excluir (por exemplo, **sample.mp4**).

5. Escolha Delete (Excluir).
6. Em Permanently delete objects?, (Excluir objetos permanentemente?), insira **permanently delete** para confirmar que deseja excluir esse objeto.
7. Escolha Delete objects (Excluir objetos).

Exclua o bucket de origem do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, escolha o botão de opção ao lado do nome do bucket que você criou na [Etapa 1 \(p. 58\)](#) (por exemplo, **tutorial-bucket**).
4. Escolha Delete (Excluir).
5. Na página Delete bucket (Excluir bucket), confirme se deseja excluir o bucket inserindo o nome do bucket no campo de texto e escolha Delete bucket (Excluir bucket).

Next steps (Próximas etapas)

Depois de concluir este tutorial, você pode continuar a explorar os seguintes casos de uso relacionados:

- Transcodifique vídeos do S3 em formatos de streaming necessários para uma televisão ou dispositivo conectado específico antes de hospedar esses vídeos com uma distribuição do CloudFront.

Para usar as operações em lote do Amazon S3, o AWS Lambda e o AWS Elemental MediaConvert para transcodificar em lote uma coleção de vídeos para vários formatos de mídia de saída, consulte [Tutorial: Vídeos de transcodificação em lote com operações em lote do S3, AWS Lambda e o AWS Elemental MediaConvert \(p. 70\)](#).

- Hospede outros objetos armazenados no S3, como imagens, áudio, animações, folhas de estilo, HTML, JavaScript, aplicações React etc., usando o CloudFront e o Route 53.

Por exemplo, consulte [Configurar um site estático usando um domínio personalizado registrado no Route 53 \(p. 105\)](#) e [Acelerar seu site com o Amazon CloudFront \(p. 118\)](#).

- Use [Amazon S3 Transfer Acceleration](#) para configurar transferências de arquivos rápidas e seguras. O Transfer Acceleration pode acelerar o carregamento de vídeo no seu bucket do S3 para transferências de longa distância de vídeos maiores. O Transfer Acceleration melhora a performance da transferência roteando o tráfego pelos locais da borda distribuídos globalmente do CloudFront e pelas redes de estrutura da AWS. Ele também usa otimizações de protocolo de rede. Para obter mais informações, consulte [Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration \(p. 147\)](#).

Tutorial: Vídeos de transcodificação em lote com operações em lote do S3, AWS Lambda e o AWS Elemental MediaConvert

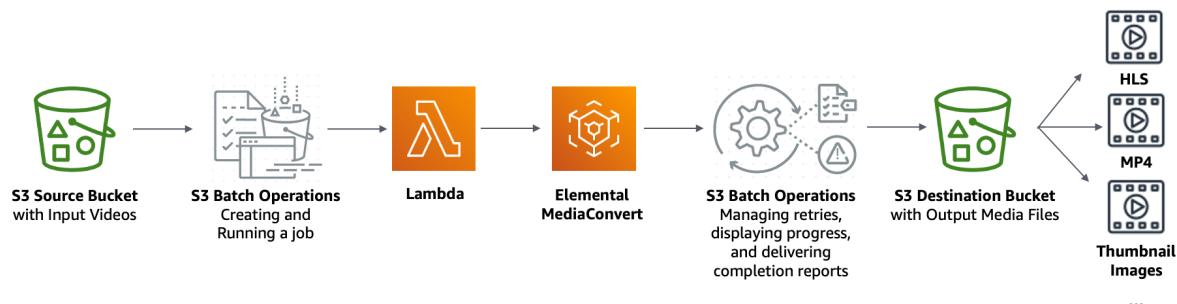
Os consumidores de vídeo usam dispositivos de todas as formas, tamanhos e épocas para desfrutar de conteúdo de mídia. Essa grande variedade de dispositivos representa um desafio para criadores e distribuidores de conteúdo. Em vez um formato de tamanho único, os vídeos precisam abranger uma

ampla variedade de tamanhos, formatos e bitrates. Essa tarefa de conversão é ainda mais desafiadora quando você tem um grande número de vídeos a serem convertidos.

A AWS oferece um método para construir uma arquitetura escalável e distribuída que faz o seguinte:

- Ingere vídeos de entrada
- Processa os vídeos para reprodução em uma ampla variedade de dispositivos
- Armazena os arquivos de mídia transcodificados
- Fornece os arquivos de mídia de saída para atender à demanda

Quando você tem extensos repositórios de vídeo armazenados no Amazon S3, pode transcodificar esses vídeos de seus formatos na fonte para vários tipos de arquivo, no tamanho, resolução e formato necessários para um determinado player ou dispositivo de vídeo. Mais especificamente, as [operações em lote do S3](#) fornecem uma solução para chamar as funções do AWS Lambda para vídeos de entrada existentes em um bucket de origem do S3. Em seguida, as funções Lambda chamam o [AWS Elemental MediaConvert](#) para executar tarefas de transcodificação de vídeo em grande escala. Os arquivos de mídia de saída convertidos são armazenados em um bucket de destino do S3.



Objetivo

Neste tutorial, você aprenderá como configurar as operações em lote do S3 para chamar uma função Lambda para transcodificação em lote de vídeos armazenados em um bucket de origem do S3. A função Lambda chama o MediaConvert para transcodificar os vídeos. As saídas para cada vídeo no bucket de origem do S3 são as seguintes:

- Um fluxo de bitrate adaptativa [HTTP Live Streaming \(HLS\)](#) para reprodução em dispositivos de vários tamanhos e larguras de banda variáveis
- Um arquivo de vídeo MP4
- Imagens em miniatura coletadas em intervalos

Tópicos

- [Pré-requisitos \(p. 72\)](#)
- [Etapa 1: Criar um bucket do S3 para os arquivos de mídia de saída \(p. 72\)](#)
- [Etapa 2: criar uma função do IAM para MediaConvert \(p. 74\)](#)
- [Etapa 3: criar uma função do IAM para a função Lambda \(p. 74\)](#)
- [Etapa 4: Criar uma função do Lambda para transcodificação de vídeo \(p. 76\)](#)
- [Etapa 5: Configurar o inventário do Amazon S3 para seu bucket de origem do S3 \(p. 88\)](#)
- [Etapa 6: criar uma função do IAM para operações em lote do S3 \(p. 91\)](#)
- [Etapa 7: criar e executar um trabalho de operações em lote do S3 \(p. 93\)](#)
- [Etapa 8: Conferir os arquivos de mídia de saída do bucket de destino do S3 \(p. 97\)](#)
- [Etapa 9: limpeza \(p. 97\)](#)
- [Next steps \(Próximas etapas\) \(p. 99\)](#)

Pré-requisitos

Antes de iniciar este tutorial, você deve ter um bucket de origem do Amazon S3 (por exemplo, **tutorial-bucket-1**) com vídeos a serem transcodificados já armazenados nele.

Você pode dar ao bucket outro nome, se desejar. Para obter mais informações sobre os nomes de bucket do Amazon S3, consulte [Regras de nomeação de bucket \(p. 127\)](#).

Para o bucket de origem do S3, mantenha as configurações relacionadas a Block Public Access settings for this bucket (Configurações de bloquear acesso público a este bucket) definidas com os padrões (Block all public access (Bloquear todo o acesso público) está habilitado). Para obter mais informações, consulte [Criação de um bucket \(p. 129\)](#).

Para obter mais informações sobre como carregar vídeos para o bucket de origem do S3, consulte [Fazer upload de objetos \(p. 171\)](#). Se você for carregar muitos arquivos de vídeo grandes para o S3, pode ser interessante usar o [Amazon S3 Transfer Acceleration](#) para configurar transferências de arquivos rápidas e seguras. O Transfer Acceleration pode acelerar o carregamento de vídeo no seu bucket do S3 para transferências de longa distância de vídeos maiores. Para obter mais informações, consulte [Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration \(p. 147\)](#).

Etapa 1: Criar um bucket do S3 para os arquivos de mídia de saída

Nesta etapa, você cria um bucket de destino do S3 para armazenar os arquivos de mídia de saída convertidos. Você também cria uma configuração CORS (Cross Origin Resource Sharing) para permitir acesso de origem cruzada aos arquivos de mídia transcodificados armazenados no bucket de destino do S3.

Subetapas

- [Criar um bucket para os arquivos de mídia de saída \(p. 72\)](#)
- [Para adicionar uma configuração CORS a um bucket de saída do S3 \(p. 73\)](#)

Criar um bucket para os arquivos de mídia de saída

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Selecione Create bucket (Criar bucket).
4. Em Bucket name (Nome do bucket), insira um nome para o seu bucket (por exemplo, **tutorial-bucket-2**).
5. Em Region (Região), escolha a Região da AWS onde deseja que o bucket resida.
6. Para garantir o acesso público aos arquivos de mídia de saída, em Block Public Access settings for this bucket (Configurações de bloqueio de acesso público para este bucket), desmarque Block all public access (Bloquear todo acesso público).

Warning

Antes de concluir esta etapa, revise [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#) para garantir que você entenda e aceite os riscos envolvidos em permitir o acesso público. Quando você desativa as configurações Block Public Access (Bloquear acesso público) para tornar seu bucket público, qualquer pessoa na Internet pode acessá-lo. Recomendamos que você bloquee todo o acesso público aos buckets.

Se você não quiser limpar as configurações de Block Public Access (Bloquear acesso público), pode usar o Amazon CloudFront para entregar os arquivos de mídia transcodificados aos visualizadores (usuários finais). Para obter mais informações, consulte [Tutorial: Hospedagem de transmissão sob demanda com o Amazon S3, Amazon CloudFront e Amazon Route 53 \(p. 56\)](#).

7. Marque a caixa de seleção ao lado de I acknowledge that the current settings might result in this bucket and the objects within becoming public (Eu reconheço que as configurações atuais podem fazer com que o bucket e os objetos fiquem públicos.)
8. Mantenha as configurações restantes definidas conforme os padrões.
9. Selecione Create bucket (Criar bucket).

Para adicionar uma configuração CORS a um bucket de saída do S3

Uma configuração de CORS JSON define uma maneira para as aplicações Web clientes (players de vídeo neste contexto) que são carregadas em um domínio reproduzirem os arquivos de mídia de saída transcodificados em um domínio diferente.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, escolha o nome do bucket que você criou antes (por exemplo, **tutorial-bucket-2**).
4. Escolha a guia Permissions (Permissões).
5. Na seção Cross-origin resource sharing (CORS) (Compartilhamento de recursos de origem cruzada (CORS)) escolha Edit (Editar).
6. Na caixa de texto de configuração de CORS, copie e cole a seguinte configuração de CORS.

A configuração do CORS deve estar no formato JSON. Neste exemplo, o atributo AllowedOrigins usa o caractere curinga (*) para especificar todas as origens. Se você souber qual é a sua origem específica, pode restringir o atributo AllowedOrigins ao URL do seu player específico. Para obter mais informações sobre esse e outros atributos, consulte [Configuração de CORS \(p. 608\)](#).

```
[  
  {  
    "AllowedOrigins": [  
      "*"  
    ],  
    "AllowedMethods": [  
      "GET"  
    ],  
    "AllowedHeaders": [  
      "*"  
    ],  
    "ExposeHeaders": []  
  }  
]
```

7. Selecione Save changes.

Etapa 2: criar uma função do IAM para MediaConvert

Para usar o AWS Elemental MediaConvert para transcodificar vídeos de entrada armazenados em seu bucket do S3, você deve ter um perfil de serviço do AWS Identity and Access Management (IAM) para conceder ao serviço MediaConvert permissões para ler e gravar arquivos de vídeo do bucket entre os buckets de origem e de destino do S3. Quando você executa trabalhos de transcodificação, o console do MediaConvert usa essa função.

Para criar uma função do IAM para MediaConvert

1. Crie uma função do IAM com um nome de função que você escolhe (por exemplo, **tutorial-mediaconvert-role**). Para criar essa função, siga as etapas em [Criar sua função do MediaConvert no IAM \(console\)](#) no Manual do usuário do AWS Elemental MediaConvert.
2. Depois de criar a função do IAM para o MediaConvert, na lista Roles (Funções), escolha o nome da função para o MediaConvert que você criou (por exemplo, **tutorial-mediaconvert-role**).
3. Na página Summary (Resumo), copie o Role ARN (ARN da função) (que começa com `arn:aws:iam::`) e salve o ARN para uso posterior.

Para obter mais informações sobre os ARNs, consulte [Nomes de recurso da Amazon \(ARNs\)](#) na Referência geral da AWS.

Etapa 3: criar uma função do IAM para a função Lambda

Para transcodificar vídeos em lote com o MediaConvert e operações em lote do S3, você usa uma função do Lambda para conectar esses dois serviços para converter vídeos. Essa função do Lambda precisa ter uma função do IAM que conceda permissões à função do Lambda para acessar o MediaConvert e as operações em lote do S3.

Subetapas

- [Criar uma função do IAM para sua função Lambda \(p. 74\)](#)
- [Incorpore uma política em linha para a função do IAM da sua função Lambda \(p. 75\)](#)

Criar uma função do IAM para sua função Lambda

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação à esquerda, escolha Roles (Funções) e Create role (Criar função).
3. Escolha o tipo de função AWS service (serviço da WS) e em Common use cases (Casos de uso comuns), escolha Lambda.
4. Escolha Next: Permissions (Próximo: permissões).
5. Na página Attach Permissions policies (Anexar políticas de permissões), insira **AWSLambdaBasicExecutionRole** na caixa Filter policies (Filtrar políticas). Para anexar a política gerenciada AWSLambdaBasicExecutionRole a esta função para conceder permissões de gravação a Amazon CloudWatch Logs, marque a caixa de seleção ao lado de AWSLambdaBasicExecutionRole.
6. Escolha Next: Tags (Próximo: etiquetas).
7. (Opcional) Adicione etiquetas à política gerenciada.
8. Selecione Next: Review (Próximo: revisar).
9. Em Role name (Nome da função), insira **tutorial-lambda-transcode-role**.

10. Escolha Create role (Criar função).

Incorpore uma política em linha para a função do IAM da sua função Lambda

Para conceder permissões para o recurso do MediaConvert que é necessário para que a função do Lambda seja executada, você deve usar uma política em linha.

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação à esquerda, escolha Roles.
3. Na lista Roles (Funções), escolha o nome da função do IAM que você criou antes para a sua função do Lambda (por exemplo, **tutorial-lambda-transcode-role**).
4. Escolha a guia Permissions (Permissões).
5. Escolha Add inline policy (Adicionar política em linha).
6. Escolha a guia JSON e copie e cole a política JSON a seguir.

Na política JSON, substitua o valor de Resource do ARN do exemplo pelo ARN da função do IAM para o MediaConvert que você criou na [Etapa 2 \(p. 74\)](#) (por exemplo, **tutorial-mediacconvert-role**).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs>CreateLogGroup",
                "logs>CreateLogStream",
                "logs>PutLogEvents"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "Logging"
        },
        {
            "Action": [
                "iam>PassRole"
            ],
            "Resource": [
                "arn:aws:iam::111122223333:role/tutorial-mediacconvert-role"
            ],
            "Effect": "Allow",
            "Sid": "PassRole"
        },
        {
            "Action": [
                "mediacconvert:)"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "MediaConvertService"
        },
        {
            "Action": [
                "s3:)"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "S3"
        }
    ]
}
```

```
        "*"
    ],
    "Effect": "Allow",
    "Sid": "S3Service"
}
]
```

7. Escolha Review Policy (Revisar política).
8. Em Name (Nome), insira **tutorial-lambda-policy**.
9. Escolha Create Policy (Criar política).

Após a criação de uma política em linha, ela é automaticamente incorporada à função do IAM de sua função Lambda.

Etapa 4: Criar uma função do Lambda para transcodificação de vídeo

Nesta seção do tutorial, você constrói uma função do Lambda usando o SDK for Python para integrar com as operações em lote do S3 e o MediaConvert. Para iniciar a transcodificação dos vídeos já armazenados no bucket de origem do S3, você executa um trabalho de operações em lote do S3 que invoca diretamente a função do Lambda para cada vídeo no bucket de origem do S3. Em seguida, a função Lambda envia um trabalho de transcodificação de cada vídeo para MediaConvert.

Subetapas

- [Grave o código de função Lambda e crie um pacote de implantação \(p. 76\)](#)
- [Crie uma função Lambda com uma função de execução \(console\) \(p. 86\)](#)
- [Implante sua função Lambda com arquivos .zip e configure a função Lambda \(console\) \(p. 87\)](#)

Grave o código de função Lambda e crie um pacote de implantação

1. Na sua máquina local, crie uma pasta denominada `batch-transcode`.
2. Na pasta `batch-transcode`, crie um arquivo com as configurações de trabalho de JSON. Por exemplo, use as configurações fornecidas nesta seção e dê ao arquivo o nome de `job.json`.

Um arquivo `job.json` especifica o seguinte:

- Quais arquivos devem ser transcodificados
- Como você deseja transcodificar seus vídeos de entrada
- Quais arquivos de mídia de saída você deseja criar
- Como dar nomes aos arquivos transcodificados
- Onde salvar os arquivos transcodificados
- Quais recursos avançados aplicar e assim por diante

Neste tutorial, usamos o arquivo `job.json` a seguir para criar as seguintes saídas para cada vídeo no bucket de origem do S3:

- Um fluxo de bitrate adaptativa HTTP Live Streaming (HLS) para reprodução em dispositivos de diversos tamanhos e diferentes larguras de banda
- Um arquivo de vídeo MP4

- Imagens em miniatura coletadas em intervalos

Este exemplo de arquivo job.json usa a variável definida por qualidade (QVCR - Quality-Defined Variable Bitrate) para otimizar a qualidade do vídeo. A saída HTTP Live Streaming (HLS) é compatível com Apple (áudio de vídeo não mixado, duração de segmento de 6 segundos e qualidade de vídeo otimizada por QVBR automática).

Se você não quiser usar as configurações de exemplo fornecidas aqui, pode gerar uma especificação de job.json baseada em seu caso de uso. Para garantir a consistência em todas as saídas, certifique-se de que os arquivos de entrada tenham configurações de vídeo e áudio semelhantes. Para arquivos de entrada com diferentes configurações de vídeo e áudio, crie automações separadas (configurações de job.json exclusivas). Para obter mais informações, consulte [Exemplo de configurações de trabalho do AWS Elemental MediaConvert em JSON](#) no Manual do usuário do AWS Elemental MediaConvert.

```
{  
    "OutputGroups": [  
        {  
            "CustomName": "HLS",  
            "Name": "Apple HLS",  
            "Outputs": [  
                {  
                    "ContainerSettings": {  
                        "Container": "M3U8",  
                        "M3u8Settings": {  
                            "AudioFramesPerPes": 4,  
                            "PcrControl": "PCR_EVERY_PES_PACKET",  
                            "PmtPid": 480,  
                            "PrivateMetadataPid": 503,  
                            "ProgramNumber": 1,  
                            "PatInterval": 0,  
                            "PmtInterval": 0,  
                            "TimedMetadata": "NONE",  
                            "VideoPid": 481,  
                            "AudioPids": [  
                                482,  
                                483,  
                                484,  
                                485,  
                                486,  
                                487,  
                                488,  
                                489,  
                                490,  
                                491,  
                                492  
                            ]  
                        }  
                    },  
                    "VideoDescription": {  
                        "Width": 640,  
                        "ScalingBehavior": "DEFAULT",  
                        "Height": 360,  
                        "TimecodeInsertion": "DISABLED",  
                        "AntiAlias": "ENABLED",  
                        "Sharpness": 50,  
                        "CodecSettings": {  
                            "Codec": "H_264",  
                            "H264Settings": {  
                                "InterlaceMode": "PROGRESSIVE",  
                                "NumberReferenceFrames": 3,  
                                "Syntax": "DEFAULT",  
                            }  
                        }  
                    }  
                }  
            ]  
        }  
    ]  
}
```

```
        "Softness": 0,
        "GopClosedCadence": 1,
        "GopSize": 2,
        "Slices": 1,
        "GopBReference": "DISABLED",
        "MaxBitrate": 1200000,
        "SlowPal": "DISABLED",
        "SpatialAdaptiveQuantization": "ENABLED",
        "TemporalAdaptiveQuantization": "ENABLED",
        "FlickerAdaptiveQuantization": "DISABLED",
        "EntropyEncoding": "CABAC",
        "FramerateControl": "INITIALIZE_FROM_SOURCE",
        "RateControlMode": "QVBR",
        "CodecProfile": "MAIN",
        "Telecine": "NONE",
        "MinIInterval": 0,
        "AdaptiveQuantization": "HIGH",
        "CodecLevel": "AUTO",
        "FieldEncoding": "PAFF",
        "SceneChangeDetect": "TRANSITION_DETECTION",
        "QualityTuningLevel": "SINGLE_PASS_HQ",
        "FramerateConversionAlgorithm": "DUPLICATE_DROP",
        "UnregisteredSeiTimecode": "DISABLED",
        "GopSizeUnits": "SECONDS",
        "ParControl": "INITIALIZE_FROM_SOURCE",
        "NumberBFramesBetweenReferenceFrames": 2,
        "RepeatPps": "DISABLED"
    }
},
"AfdSignaling": "NONE",
"DropFrameTimecode": "ENABLED",
"RespondToAfd": "NONE",
"ColorMetadata": "INSERT"
},
"OutputSettings": {
    "HlsSettings": {
        "AudioGroupId": "program_audio",
        "AudioRenditionSets": "program_audio",
        "SegmentModifier": "$dt$",
        "IFrameOnlyManifest": "EXCLUDE"
    }
},
"NameModifier": "_360"
},
{
    "ContainerSettings": {
        "Container": "M3U8",
        "M3u8Settings": {
            "AudioFramesPerPes": 4,
            "PcrControl": "PCR_EVERY_PES_PACKET",
            "PmtPid": 480,
            "PrivateMetadataPid": 503,
            "ProgramNumber": 1,
            "PatInterval": 0,
            "PmtInterval": 0,
            "TimedMetadata": "NONE",
            "TimedMetadataPid": 502,
            "VideoPid": 481,
            "AudioPids": [
                482,
                483,
                484,
                485,
                486,
                487,
                488,
```

```
        489,
        490,
        491,
        492
    ],
},
"VideoDescription": {
    "Width": 960,
    "ScalingBehavior": "DEFAULT",
    "Height": 540,
    "TimecodeInsertion": "DISABLED",
    "AntiAlias": "ENABLED",
    "Sharpness": 50,
    "CodecSettings": {
        "Codec": "H_264",
        "H264Settings": {
            "InterlaceMode": "PROGRESSIVE",
            "NumberReferenceFrames": 3,
            "Syntax": "DEFAULT",
            "Softness": 0,
            "GopClosedCadence": 1,
            "GopSize": 2,
            "Slices": 1,
            "GopBReference": "DISABLED",
            "MaxBitrate": 3500000,
            "SlowPal": "DISABLED",
            "SpatialAdaptiveQuantization": "ENABLED",
            "TemporalAdaptiveQuantization": "ENABLED",
            "FlickerAdaptiveQuantization": "DISABLED",
            "EntropyEncoding": "CABAC",
            "FramerateControl": "INITIALIZE_FROM_SOURCE",
            "RateControlMode": "QVBR",
            "CodecProfile": "MAIN",
            "Telecine": "NONE",
            "MinIInterval": 0,
            "AdaptiveQuantization": "HIGH",
            "CodecLevel": "AUTO",
            "FieldEncoding": "PAFF",
            "SceneChangeDetect": "TRANSITION_DETECTION",
            "QualityTuningLevel": "SINGLE_PASS_HQ",
            "FramerateConversionAlgorithm": "DUPLICATE_DROP",
            "UnregisteredSeiTimecode": "DISABLED",
            "GopSizeUnits": "SECONDS",
            "ParControl": "INITIALIZE_FROM_SOURCE",
            "NumberBFramesBetweenReferenceFrames": 2,
            "RepeatPps": "DISABLED"
        }
    },
    "AfdSignaling": "NONE",
    "DropFrameTimecode": "ENABLED",
    "RespondToAfd": "NONE",
    "ColorMetadata": "INSERT"
},
"OutputSettings": {
    "HlsSettings": {
        "AudioGroupId": "program_audio",
        "AudioRenditionSets": "program_audio",
        "SegmentModifier": "$dt$",
        "IFrameOnlyManifest": "EXCLUDE"
    }
},
"NameModifier": "_540"
},
{
    "ContainerSettings": {
```

```
"Container": "M3U8",
"M3u8Settings": {
    "AudioFramesPerPes": 4,
    "PcrControl": "PCR_EVERY_PES_PACKET",
    "PmtPid": 480,
    "PrivateMetadataPid": 503,
    "ProgramNumber": 1,
    "PatInterval": 0,
    "PmtInterval": 0,
    "TimedMetadata": "NONE",
    "VideoPid": 481,
    "AudioPids": [
        482,
        483,
        484,
        485,
        486,
        487,
        488,
        489,
        490,
        491,
        492
    ]
},
"VideoDescription": {
    "Width": 1280,
    "ScalingBehavior": "DEFAULT",
    "Height": 720,
    "TimecodeInsertion": "DISABLED",
    "AntiAlias": "ENABLED",
    "Sharpness": 50,
    "CodecSettings": {
        "Codec": "H_264",
        "H264Settings": {
            "InterlaceMode": "PROGRESSIVE",
            "NumberReferenceFrames": 3,
            "Syntax": "DEFAULT",
            "Softness": 0,
            "GopClosedCadence": 1,
            "GopSize": 2,
            "Slices": 1,
            "GopBReference": "DISABLED",
            "MaxBitrate": 5000000,
            "SlowPal": "DISABLED",
            "SpatialAdaptiveQuantization": "ENABLED",
            "TemporalAdaptiveQuantization": "ENABLED",
            "FlickerAdaptiveQuantization": "DISABLED",
            "EntropyEncoding": "CABAC",
            "FramerateControl": "INITIALIZE_FROM_SOURCE",
            "RateControlMode": "QVBR",
            "CodecProfile": "MAIN",
            "Telecine": "NONE",
            "MinIInterval": 0,
            "AdaptiveQuantization": "HIGH",
            "CodecLevel": "AUTO",
            "FieldEncoding": "PAFF",
            "SceneChangeDetect": "TRANSITION_DETECTION",
            "QualityTuningLevel": "SINGLE_PASS_HQ",
            "FramerateConversionAlgorithm": "DUPLICATE_DROP",
            "UnregisteredSeiTimecode": "DISABLED",
            "GopSizeUnits": "SECONDS",
            "ParControl": "INITIALIZE_FROM_SOURCE",
            "NumberBFramesBetweenReferenceFrames": 2,
            "RepeatPps": "DISABLED"
        }
    }
}
```

```
        },
        "AfdSignaling": "NONE",
        "DropFrameTimecode": "ENABLED",
        "RespondToAfd": "NONE",
        "ColorMetadata": "INSERT"
    },
    "OutputSettings": {
        "HlsSettings": {
            "AudioGroupId": "program_audio",
            "AudioRenditionSets": "program_audio",
            "SegmentModifier": "$dt$",
            "IFrameOnlyManifest": "EXCLUDE"
        }
    },
    "NameModifier": "_720"
},
{
    "ContainerSettings": {
        "Container": "M3U8",
        "M3u8Settings": {}
    },
    "AudioDescriptions": [
        {
            " AudioSourceName": "Audio Selector 1",
            " CodecSettings": {
                "Codec": "AAC",
                "AacSettings": {
                    "Bitrate": 96000,
                    "CodingMode": "CODING_MODE_2_0",
                    "SampleRate": 48000
                }
            }
        }
    ],
    "OutputSettings": {
        "HlsSettings": {
            "AudioGroupId": "program_audio",
            "AudioTrackType": "ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT"
        }
    },
    "NameModifier": "_audio"
},
],
"OutputGroupSettings": {
    "Type": "HLS_GROUP_SETTINGS",
    "HlsGroupSettings": {
        "ManifestDurationFormat": "INTEGER",
        "SegmentLength": 6,
        "TimedMetadataId3Period": 10,
        "CaptionLanguageSetting": "OMIT",
        "Destination": "s3://EXAMPLE-BUCKET/HLS/",
        "DestinationSettings": {
            "S3Settings": {
                "AccessControl": {
                    "CannedAcl": "PUBLIC_READ"
                }
            }
        },
        "TimedMetadataId3Frame": "PRIV",
        "CodecSpecification": "RFC_4281",
        "OutputSelection": "MANIFESTS_AND_SEGMENTS",
        "ProgramDateTimePeriod": 600,
        "MinSegmentLength": 0,
        "DirectoryStructure": "SINGLE_DIRECTORY",
        "ProgramDateTime": "EXCLUDE",
        "ManifestCompression": "NONE"
    }
}
```

```
"SegmentControl": "SEGMENTED_FILES",
"ManifestCompression": "NONE",
"ClientCache": "ENABLED",
"StreamInfResolution": "INCLUDE"
}
},
{
"CustomName": "MP4",
"Name": "File Group",
"Outputs": [
{
"ContainerSettings": {
"Container": "MP4",
"Mp4Settings": {
"CslgAtom": "INCLUDE",
"FreeSpaceBox": "EXCLUDE",
"MoovPlacement": "PROGRESSIVE_DOWNLOAD"
}
},
"VideoDescription": {
"Width": 1280,
"ScalingBehavior": "DEFAULT",
"Height": 720,
"TimecodeInsertion": "DISABLED",
"AntiAlias": "ENABLED",
"Sharpness": 100,
"CodecSettings": {
"Codec": "H_264",
"H264Settings": {
"InterlaceMode": "PROGRESSIVE",
"ParNumerator": 1,
"NumberReferenceFrames": 3,
"Syntax": "DEFAULT",
"Softness": 0,
"GopClosedCadence": 1,
"HRdBUFFERInitialFillPercentage": 90,
"GopSize": 2,
"Slices": 2,
"GopBReference": "ENABLED",
"HRdBUFFERSize": 10000000,
"MaxBitrate": 5000000,
"ParDenominator": 1,
"EntropyEncoding": "CABAC",
"RateControlMode": "QVBR",
"CodecProfile": "HIGH",
"MinIInterval": 0,
"AdaptiveQuantization": "AUTO",
"CodecLevel": "AUTO",
"FieldEncoding": "PAFF",
"SceneChangeDetect": "ENABLED",
"QualityTuningLevel": "SINGLE_PASS_HQ",
"UnregisteredSeiTImecode": "DISABLED",
"GopSizeUnits": "SECONDS",
"ParControl": "SPECIFIED",
"NumberBFramesBetweenReferenceFrames": 3,
"RepeatPps": "DISABLED",
"DynamicSubGop": "ADAPTIVE"
}
}
},
"AfdSignaling": "NONE",
"DropFrameTimecode": "ENABLED",
"RespondToAfd": "NONE",
"ColorMetadata": "INSERT"
},
"AudioDescriptions": [
```

```
{  
    "AudioTypeControl": "FOLLOW_INPUT",  
    "AudioSourceName": "Audio Selector 1",  
    "CodecSettings": {  
        "Codec": "AAC",  
        "AacSettings": {  
            "AudioDescriptionBroadcasterMix": "NORMAL",  
            "Bitrate": 160000,  
            "RateControlMode": "CBR",  
            "CodecProfile": "LC",  
            "CodingMode": "CODING_MODE_2_0",  
            "RawFormat": "NONE",  
            "SampleRate": 48000,  
            "Specification": "MPEG4"  
        }  
    },  
    "LanguageCodeControl": "FOLLOW_INPUT",  
    "AudioType": 0  
},  
]  
],  
],  
"OutputGroupSettings": {  
    "Type": "FILE_GROUP_SETTINGS",  
    "FileGroupSettings": {  
        "Destination": "s3://EXAMPLE-BUCKET/MP4/",  
        "DestinationSettings": {  
            "S3Settings": {  
                "AccessControl": {  
                    "CannedAcl": "PUBLIC_READ"  
                }  
            }  
        }  
    }  
},  
,  
{  
    "CustomName": "Thumbnails",  
    "Name": "File Group",  
    "Outputs": [  
        {  
            "ContainerSettings": {  
                "Container": "RAW"  
            },  
            "VideoDescription": {  
                "Width": 1280,  
                "ScalingBehavior": "DEFAULT",  
                "Height": 720,  
                "TimecodeInsertion": "DISABLED",  
                "AntiAlias": "ENABLED",  
                "Sharpness": 50,  
                "CodecSettings": {  
                    "Codec": "FRAME_CAPTURE",  
                    "FrameCaptureSettings": {  
                        "FramerateNumerator": 1,  
                        "FramerateDenominator": 5,  
                        "MaxCaptures": 500,  
                        "Quality": 80  
                    }  
                },  
                "AfdSignaling": "NONE",  
                "DropFrameTimecode": "ENABLED",  
                "RespondToAfd": "NONE",  
                "ColorMetadata": "INSERT"  
            }  
        }  
    ]  
}
```

```
        ],
        "OutputGroupSettings": {
            "Type": "FILE_GROUP_SETTINGS",
            "FileGroupSettings": {
                "Destination": "s3://EXAMPLE-BUCKET/Thumbnails/",
                "DestinationSettings": {
                    "S3Settings": {
                        "AccessControl": {
                            "CannedAcl": "PUBLIC_READ"
                        }
                    }
                }
            }
        }
    ],
    "AdAvailOffset": 0,
    "Inputs": [
        {
            "AudioSelectors": {
                "Audio Selector 1": {
                    "Offset": 0,
                    "DefaultSelection": "DEFAULT",
                    "ProgramSelection": 1
                }
            },
            "VideoSelector": {
                "ColorSpace": "FOLLOW"
            },
            "FilterEnable": "AUTO",
            "PsiControl": "USE_PSI",
            "FilterStrength": 0,
            "DeblockFilter": "DISABLED",
            "DenoiseFilter": "DISABLED",
            "TimecodeSource": "EMBEDDED",
            "FileInput": "s3://EXAMPLE-INPUT-BUCKET/input.mp4"
        }
    ]
}
```

3. Na pasta `batch-transcode`, crie um arquivo com uma função Lambda. Você pode usar o seguinte exemplo Python e nomear o arquivo `convert.py`.

As operações em lote do S3 enviam dados de tarefa específicos para uma função do Lambda e requerem os dados do resultado de volta. Para obter exemplos de solicitação e resposta para a função do Lambda, informações sobre códigos de resposta e de resultados, e exemplos de funções do Lambda para operações em lote do S3, consulte [Invocar função do AWS Lambda \(p. 966\)](#).

```
import json
import os
from urllib.parse import urlparse
import uuid
import boto3

"""

When you run an S3 Batch Operations job, your job
invokes this Lambda function. Specifically, the Lambda function is
invoked on each video object listed in the manifest that you specify
for the S3 Batch Operations job in Step 5 (p. 88).

Input parameter "event": The S3 Batch Operations event as a request
for the Lambda function.

Input parameter "context": Context about the event.
```

```

Output: A result structure that Amazon S3 uses to interpret the result
       of the operation. It is a job response returned back to S3 Batch Operations.
"""
def handler(event, context):

    invocation_schema_version = event['invocationSchemaVersion']
    invocation_id = event['invocationId']
    task_id = event['tasks'][0]['taskId']

    source_s3_key = event['tasks'][0]['s3Key']
    source_s3_bucket = event['tasks'][0]['s3BucketArn'].split(':::')[ -1]
    source_s3 = 's3://' + source_s3_bucket + '/' + source_s3_key

    result_list = []
    result_code = 'Succeeded'
    result_string = 'The input video object was converted successfully.'

    # The type of output group determines which media players can play
    # the files transcoded by MediaConvert.
    # For more information, see Creating outputs with AWS Elemental MediaConvert.
    output_group_type_dict = {
        'HLS_GROUP_SETTINGS': 'HlsGroupSettings',
        'FILE_GROUP_SETTINGS': 'FileGroupSettings',
        'CMAF_GROUP_SETTINGS': 'CmafGroupSettings',
        'DASH_ISO_GROUP_SETTINGS': 'DashIsoGroupSettings',
        'MS_SMOOTH_GROUP_SETTINGS': 'MsSmoothGroupSettings'
    }

    try:
        job_name = 'Default'
        with open('job.json') as file:
            job_settings = json.load(file)

        job_settings['Inputs'][0]['FileInput'] = source_s3

        # The path of each ouput video is constructed based on the values of
        # the attributes in each object of OutputGroups in the job.json file.
        destination_s3 = 's3://{0}/{1}/{2}' \
            .format(os.environ['DestinationBucket'],
                    os.path.splitext(os.path.basename(source_s3_key))[0],
                    os.path.splitext(os.path.basename(job_name))[0])

        for output_group in job_settings['OutputGroups']:
            output_group_type = output_group['OutputGroupSettings']['Type']
            if output_group_type in output_group_type_dict.keys():
                output_group_type = output_group_type_dict[output_group_type]
                output_group['OutputGroupSettings'][output_group_type]['Destination'] =
                    \
                    "{0}{1}" .format(destination_s3,
                                      urlparse(output_group['OutputGroupSettings'][
                                              output_group_type]['Destination']).path)
            else:
                raise ValueError("Exception: Unknown Output Group Type {}."
                                .format(output_group_type))

        job_metadata_dict = {
            'assetID': str(uuid.uuid4()),
            'application': os.environ['Application'],
            'input': source_s3,
            'settings': job_name
        }

        region = os.environ['AWS_DEFAULT_REGION']
        endpoints = boto3.client('mediaconvert', region_name=region) \
            .describe_endpoints()
    
```

```

client = boto3.client('mediaconvert', region_name=region,
                      endpoint_url=endpoints['Endpoints'][0]['Url'],
                      verify=False)

try:
    client.create_job(Role=os.environ['MediaConvertRole'],
                       UserMetadata=job_metadata_dict,
                       Settings=job_settings)
    # You can customize error handling based on different error codes that
    # MediaConvert can return.
    # For more information, see MediaConvert error codes.
    # When the result_code is TemporaryFailure, S3 Batch Operations retries
    # the task before the job is completed. If this is the final retry,
    # the error message is included in the final report.
except Exception as error:
    result_code = 'TemporaryFailure'
    raise

except Exception as error:
    if result_code != 'TemporaryFailure':
        result_code = 'PermanentFailure'
    result_string = str(error)

finally:
    result_list.append({
        'taskId': task_id,
        'resultCode': result_code,
        'resultString': result_string,
    })

return {
    'invocationSchemaVersion': invocation_schema_version,
    'treatMissingKeyAs': 'PermanentFailure',
    'invocationId': invocation_id,
    'results': result_list
}

```

- Para criar um pacote de implantação com `convert.py` e `job.json` como um arquivo `.zip`, denominado `lambda.zip` em seu terminal local, abra a pasta `batch-transcode` que você criou antes e execute o comando a seguir.

Para usuários do macOS, execute o seguinte comando:

```
zip -r lambda.zip convert.py job.json
```

Para usuários do Windows, execute os seguintes comandos:

```
powershell Compress-Archive convert.py lambda.zip
```

```
powershell Compress-Archive -update job.json lambda.zip
```

Crie uma função Lambda com uma função de execução (console)

- Abra o console do AWS Lambda em <https://console.aws.amazon.com/lambda/>.
- No painel de navegação à esquerda, escolha Functions (Funções).
- Escolha Create function (Criar função).

4. Escolha Author from scratch (Criar do zero).
5. Em Basic information (Informações básicas), faça o seguinte:
 - a. Em Function name (Nome da função), insira **tutorial-lambda-convert**.
 - b. Para Runtime (Tempo de execução), escolha Python 3.8 ou uma versão posterior do Python.
6. Escolha Change default execution role (Alterar a função de execução padrão) e, em Execution role (Função de execução), escolha Use an existing role (Usar uma função existente).
7. Em Existing role (Função de saída), escolha o nome da função do IAM que você criou para sua função do Lambda na [Etapa 3 \(p. 74\)](#) (por exemplo, **tutorial-lambda-transcode-role**).
8. Mantenha as configurações restantes definidas conforme os padrões.
9. Escolha Create function (Criar função).

Implante sua função Lambda com arquivos .zip e configure a função Lambda (console)

1. Na seção Code Source (Código-fonte) da página da função do Lambda que você criou (por exemplo, **tutorial-lambda-convert**), escolha Upload from (Carregar de) e depois .zip file (arquivo .zip).
2. Selecione Upload (Carregar) para selecionar seu arquivo .zip local.
3. Selecione o arquivo lambda.zip que você criou antes e escolha Open (Abrir).
4. Escolha Save (Salvar).
5. Na seção Runtime settings (Configurações do tempo de execução), escolha Edit (Editar).
6. Para informar ao tempo de execução do Lambda qual método de handler em seu código de função do Lambda deve invocar, insira no campo **convert.handler** Handler.

Ao configurar uma função em Python, o valor da configuração do handler é o nome do arquivo e o nome do módulo do handler exportado, separados por um ponto (.). Por exemplo, `convert.handler` chama o método `handler` definido no arquivo `convert.py`.

7. Escolha Save (Salvar).
8. Em sua página da função Lambda, escolha a guia Configuration (Configuração). No painel de navegação esquerdo na guia Configuration (Configuração), escolha Environment variables (Variáveis de ambiente) e, depois, escolha Edit (Editar).
9. Escolha Add environment variable (Adicionar variável de ambiente). Em seguida, insira a Key (Chave) e o Value (Valor) especificados para cada uma das seguintes variáveis de ambiente:

- Key (Chave): **DestinationBucket** Value (Valor): **tutorial-bucket-2**

Este valor é o bucket do S3 para os arquivos de mídia de saída que você criou na [Etapa 1 \(p. 72\)](#).

- Key (Chave): **MediaConvertRole** Value (Valor): **arn:aws:iam::111122223333:role/tutorial-mediaconvert-role**

Este valor é o ARN da função do IAM para MediaConvert que você criou na [Etapa 2 \(p. 74\)](#). Certifique-se de substituir esse ARN pelo ARN real da sua função do IAM.

- Key (Chave): **Application** Value (Valor): **Batch-Transcoding**

Esse valor é o nome da sua aplicação.

10. Escolha Save (Salvar).
11. (Opcional) Na guia Configuration (Configuração), na seção General configuration (Configuração geral) do painel de navegação à esquerda, escolha Edit (Editar). No campo Timeout (Tempo limite), insira **2 min 0 segundos**. Escolha Save (Salvar).

Timeout (Tempo limite) é a quantidade de tempo durante a qual o Lambda permite que uma função seja executada antes de interrompê-la. O padrão é 3 segundos. O preço é baseado na quantidade de memória configurada e na quantidade de tempo em que o código é executado. Para obter mais informações, consulte [Definição de preço do AWS Lambda](#).

Etapa 5: Configurar o inventário do Amazon S3 para seu bucket de origem do S3

Depois de configurar a função de transcodificação do Lambda, crie um trabalho de operações em lote do S3 para transcodificar um conjunto de vídeos. Primeiro, você precisa de uma lista de objetos de vídeo de entrada nos quais deseja que as operações em lote do S3 executem a ação de transcodificação especificada. Para obter uma lista de objetos de vídeo de entrada, você pode gerar um relatório de inventário do S3 para o bucket de origem do S3 (por exemplo, **tutorial-bucket-1**).

Subetapas

- [Crie e configure um bucket para relatórios de inventário do S3 para vídeos de entrada \(p. 88\)](#)
- [Configure o inventário do Amazon S3 para o bucket de origem de vídeo do S3 \(p. 89\)](#)
- [Confira o relatório de inventário para o seu bucket de origem de vídeo S3 \(p. 90\)](#)

Crie e configure um bucket para relatórios de inventário do S3 para vídeos de entrada

Para armazenar relatórios de inventário do S3 que listam os objetos do bucket de origem do S3, você precisa criar um bucket de destino de inventário do S3 e depois configurar uma política de bucket para que o bucket grave os arquivos de inventário no bucket de origem do S3.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Selecione Create bucket (Criar bucket).
4. Em Bucket name (Nome do bucket), insira um nome para o seu bucket (por exemplo, **tutorial-bucket-3**).
5. Em Região da AWS, escolha a Região da AWS na qual você deseja que o bucket resida.

O bucket de destino do inventário deve estar na mesma Região da AWS que o bucket de origem para o qual você está configurando o inventário do S3. O bucket de destino do inventário pode estar em uma Conta da AWS diferente.

6. Em Block Public Access settings for this bucket (Configurações de bloquear acesso público para este bucket), mantenha as configurações padrão (Block all public access (Bloquear todo acesso público) está habilitado).
7. Mantenha as configurações restantes definidas conforme os padrões.
8. Selecione Create bucket (Criar bucket).
9. Na lista de Buckets, escolha o nome do bucket que você acabou de criar (por exemplo, **tutorial-bucket-3**).
10. Para conceder permissão ao Amazon S3 para gravar dados para os relatórios de inventário no bucket de destino de inventário do S3, escolha a guia Permissions (Permissões).
11. Role para baixo até a seção Bucket policy (Política de bucket) e escolha Edit (Editar). A página Bucket policy (Política de bucket) se abre.

12. Para conceder permissões para o inventário do S3, no campo Policy (Política), cole a seguinte política de bucket.

Substitua os três valores do exemplo pelos seguintes valores:

- O nome do bucket criado para armazenar os relatórios de inventário (por exemplo, **tutorial-bucket-3**).
- O nome do bucket de origem que armazena os vídeos de entrada (por exemplo, **tutorial-bucket-1**).
- O ID da Conta da AWS que você usou para criar o bucket de origem do vídeo S3 (por exemplo, **111122223333**).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "InventoryAndAnalyticsExamplePolicy",  
            "Effect": "Allow",  
            "Principal": {"Service": "s3.amazonaws.com"},  
            "Action": "s3:PutObject",  
            "Resource": ["arn:aws:s3:::tutorial-bucket-3/*"],  
            "Condition": {  
                "ArnLike": {  
                    "aws:SourceArn": "arn:aws:s3:::tutorial-bucket-1"  
                },  
                "StringEquals": {  
                    "aws:SourceAccount": "111122223333",  
                    "s3:x-amz-acl": "bucket-owner-full-control"  
                }  
            }  
        }  
    ]  
}
```

13. Selecione Save changes.

Configure o inventário do Amazon S3 para o bucket de origem de vídeo do S3

Para gerar uma lista de arquivos simples de objetos de vídeo e metadados, você deve configurar o inventário do S3 para o seu bucket de origem de vídeo do S3. Esses relatórios programados podem incluir todos os objetos no bucket ou objetos agrupados por um prefixo compartilhado. Neste tutorial, o relatório de inventário do S3 inclui todos os objetos de vídeo em seu bucket de origem do S3.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Para configurar um relatório de inventário do S3 dos vídeos de entrada no bucket de origem do S3, na seção Buckets, escolha o nome do bucket de origem do S3 (por exemplo, **tutorial-bucket-1**).
4. Escolha a guia Management.
5. Role para baixo até a seção Inventory configurations (Configurações de inventário), e escolha Create Inventory configuration (Criar configuração de inventário).
6. Para Inventory configuration name (Nome da configuração de inventário), insira um nome (por exemplo, **tutorial-inventory-config**).

7. Em Inventory scope (Escopo do inventário), escolha Current version only (Somente versão atual) para Object versions (Versões do objeto) e mantenha as outras configurações de Inventory scope(Escopo do inventário) definidas com os padrões para este tutorial.
8. Na seção Report details (Detalhes do relatório), para Destination bucket (Bucket de destino), escolha This account (Esta conta).
9. Para Destination (Destino), escolha Browse S3 (Procurar no S3) e escolha o bucket de destino que você criou antes para salvar os relatórios de inventário (por exemplo, **tutorial-bucket-3**). Em seguida, escolha Choose path (Escolher caminho).

O bucket de destino do inventário deve estar na mesma Região da AWS que o bucket de origem para o qual você está configurando o inventário do S3. O bucket de destino do inventário pode estar em uma Conta da AWS diferente.

No campo de bucket de Destination (Destino), a Destination bucket permission (Permissão do bucket de destino) é adicionada à política de bucket de destino para permitir que o Amazon S3 coloque dados nesse bucket de destino do inventário. Para obter mais informações, consulte [Criação de uma política de bucket de destino \(p. 790\)](#).

10. Em Frequency (Frequência), escolha Daily (Diária).
11. Em Output format (Formato de saída), escolha CSV.
12. Em Status, escolha Enabled (Habilitado).
13. Em Server-side encryption (Criptografia do lado do servidor), escolha Disable (Desabilitar) para este tutorial.

Para obter mais informações, consulte [Configuração de um inventário usando o console do S3 \(p. 792\)](#) e [Conceder permissão para que o Amazon S3 use a AWS KMS key para criptografia \(p. 791\)](#).

14. Na seção Additional fields - optional (Campos adicionais - opcionais), escolha Size (Tamanho), Last modified (Última modificação) e Storage class (Classe de armazenamento).
15. Escolha Create (Criar OpsItem).

Para obter mais informações, consulte [Configuração de um inventário usando o console do S3 \(p. 792\)](#).

Confira o relatório de inventário para o seu bucket de origem de vídeo S3

Quando um relatório de inventário é publicado, os arquivos manifestos são enviados para o bucket de destino do S3.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista de Buckets, escolha o nome do bucket de origem de vídeo (por exemplo, **tutorial-bucket-1**).
4. Escolha Management (Gerenciamento).
5. Para ver se o relatório de inventário do S3 está pronto para você poder criar um trabalho de operações em lote do S3 na [Etapa 7 \(p. 93\)](#), em Inventory configurations (Configurações do inventário), confira se o botão Create job from manifest (Criar trabalho a partir do manifesto) está habilitado.

Note

O primeiro relatório de inventário pode demorar até 48 horas para ser entregue. Se o botão Create job from manifest (Criar trabalho do manifesto) estiver desativado, o primeiro relatório de inventário não foi entregue. Espere até que o primeiro relatório de inventário seja entregue

e o botão Create job from manifest (Criar trabalho a partir do manifesto) esteja habilitado para criar um trabalho de operações em lote do S3 na [Etapa 7 \(p. 93\)](#).

6. Para conferir um relatório de inventário do S3 (`manifest.json`), na coluna Destination (Destino), escolha o nome do bucket de destino do inventário que você criou antes para armazenar relatórios de inventário (por exemplo, `tutorial-bucket-3`).
 7. Na guia Objects (Objetos), escolha a pasta existente com o nome do seu bucket de origem do S3 (por exemplo, `tutorial-bucket-1`). Em seguida, escolha o nome que você inseriu em Inventory configuration name (Nome da configuração do inventário) quando criou a configuração de inventário antes (por exemplo, `tutorial-inventory-config`).
- Você pode ver uma lista de pastas com as datas de geração dos relatórios como seus nomes.
8. Para conferir o relatório de inventário diário do S3 para uma determinada data, escolha a pasta com um nome de data de geração e depois escolha `manifest.json`.
 9. Para verificar os detalhes do relatório de inventário em uma data específica, na página `manifest.json`, escolha Download (Baixar) ou Open (Abrir).

Etapa 6: criar uma função do IAM para operações em lote do S3

Para usar as operações em lote do S3 para fazer transcodificação em lote, você deve primeiro criar uma função do IAM para conceder permissões ao Amazon S3 para executar as operações em lote do S3.

Subetapas

- [Criar uma política do IAM para operações em lote do S3 \(p. 91\)](#)
- [Crie uma função do IAM de operações em lote do S3 e anexe políticas de permissão \(p. 92\)](#)

Criar uma política do IAM para operações em lote do S3

Você deve criar uma política do IAM que conceda permissão às operações em lote do S3 para ler o manifesto de entrada, invocar a função do Lambda e gravar o relatório de conclusão do trabalho de operações em lote do S3.

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação à esquerda, escolha Policies (Políticas).
3. Escolha Create policy (Criar política).
4. Escolha a guia JSON.
5. No campo de texto JSON, cole a política JSON a seguir.

Na política JSON, substitua os quatro valores do exemplo pelos seguintes valores:

- O nome do bucket de origem que armazena vídeos de entrada (por exemplo, `tutorial-bucket-1`).
- O nome do bucket de destino de inventário que você criou na [Etapa 5 \(p. 88\)](#) para armazenar os arquivos `manifest.json` (por exemplo, `tutorial-bucket-3`).
- O nome do bucket que você criou na [Etapa 1 \(p. 72\)](#) para armazenar arquivos de mídia de saída (por exemplo, `tutorial-bucket-2`). Neste tutorial, colocamos relatórios de conclusão de trabalho no bucket de destino para arquivos de mídia de saída.
- A função ARN da função Lambda que você criou na [Etapa 4 \(p. 76\)](#). Para localizar e copiar o ARN da função do Lambda, faça o seguinte:

- Em uma nova guia do navegador, abra a página Functions (Funções) no console do Lambda em <https://console.aws.amazon.com/lambda/home#/functions>.
- Na lista de Functions (Funções), escolha o nome da função do Lambda que você criou na [Etapa 4 \(p. 76\)](#) (por exemplo, **tutorial-lambda-convert**).
- Escolha Copy ARN (Copiar ARN).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3GetObject",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::tutorial-bucket-1/*",
                "arn:aws:s3:::tutorial-bucket-3/*"
            ]
        },
        {
            "Sid": "S3PutJobCompletionReport",
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::tutorial-bucket-2/*"
        },
        {
            "Sid": "S3BatchOperationsInvokeLambda",
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:us-west-2:111122223333:function:tutorial-lambda-
convert"
            ]
        }
    ]
}
```

6. Escolha Next: Tags (Próximo: etiquetas).
7. Selecione Next: Review (Próximo: revisar).
8. No campo Name (Nome), insira **tutorial-s3batch-policy**.
9. Escolha Create policy (Criar política).

Crie uma função do IAM de operações em lote do S3 e anexe políticas de permissão

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação à esquerda, escolha Roles (Funções) e Create role (Criar função).
3. Escolha o tipo de função AWS service (Serviço da AWS) e depois escolha o serviço S3.
4. Em Select your use case (Selecionar seu caso de uso), escolha S3 Batch Operations (Operações em lote do S3).
5. Escolha Next: Permissions (Próximo: permissões).

6. Em Attach permissions policies (Anexar políticas de permissões), insira o nome da política do IAM que você criou antes (por exemplo, **tutorial-s3batch-policy**) na caixa de pesquisa para filtrar a lista de políticas. Marque a caixa de seleção ao lado do nome da política (por exemplo, **tutorial-s3batch-policy**).
7. Escolha Next: Tags (Próximo: etiquetas).
8. Escolha Next: Review (Próximo: revisar).
9. Em Role name (Nome da função), insira **tutorial-s3batch-role**.
10. Escolha Create role (Criar função).

Depois que você cria a função do IAM para as operações em lote do S3, a política de confiança a seguir é anexada automaticamente à função. Essa política de confiança permite que o principal das operações em lote do S3 assuma a função do IAM.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "batchoperations.s3.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

Etapa 7: criar e executar um trabalho de operações em lote do S3

Para criar um trabalho de operações em lote do S3 para processar os vídeos de entrada no bucket de origem do S3, você deve especificar os parâmetros para esse trabalho específico.

Note

Antes de começar a criar um trabalho de operações em lote do S3, certifique-se de que o botão Create job from manifest (Criar trabalho a partir do manifesto) esteja habilitado. Para obter mais informações, consulte [Confira o relatório de inventário para o seu bucket de origem de vídeo S3 \(p. 90\)](#). Se o botão Create job from manifest (Criar trabalho a partir do manifesto) estiver desabilitado, o primeiro relatório de inventário não foi entregue e você deve esperar até que o botão esteja habilitado. Depois de configurar o inventário do Amazon S3 para o bucket de origem do S3 na [Etapa 5 \(p. 88\)](#), a entrega do primeiro relatório de inventário pode levar até 48 horas.

Subetapas

- [Criar um trabalho de operações em lote do S3 \(p. 94\)](#)
- [Execute o trabalho de operações em lote do S3 para chamar a função Lambda \(p. 95\)](#)
- [\(Opcional\) Verificar seu relatório de conclusão \(p. 95\)](#)
- [\(Opcional\) Monitore cada chamada do Lambda no console do Lambda \(p. 96\)](#)
- [\(Opcional\) Monitore cada trabalho de transcodificação de vídeo MediaConvert no console do MediaConvert \(p. 96\)](#)

Criar um trabalho de operações em lote do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação esquerdo, escolha Batch Operations (Operações em lote).
3. Escolha Create job (Criar trabalho).
4. Para Região da AWS, escolha a região onde você deseja criar o trabalho.

Neste tutorial, para usar o trabalho de operações em lote do S3 para invocar uma função do Lambda, você deve criar o trabalho na mesma região que o bucket de origem de vídeo do S3 onde os objetos referenciados no manifesto estão localizados.

5. Na seção Manifest (Manifesto), faça o seguinte:
 - a. Para Manifest format (Formato do manifesto), escolha S3 Inventory report (manifest.json) (Relatório do S3 Inventory [manifest.json]).
 - b. Para Manifest object (Objeto de manifesto), escolha Browse S3 (Procurar no S3) para encontrar o bucket que você criou na [Etapa 5 \(p. 88\)](#) para armazenar relatórios de inventário (por exemplo, **tutorial-bucket-3**). Na página Manifest object, (Objeto de manifesto) navegue pelos nomes de objetos até encontrar um arquivo `manifest.json` para uma data específica. Este arquivo lista as informações sobre todos os vídeos que você deseja transcodificar em lote. Quando encontrar o arquivo `manifest.json` que você deseja usar, escolha o botão de opção ao lado dele. Em seguida, escolha Choose path (Escolher caminho).
 - c. (Opcional) Para Manifest object version ID - optional (ID da versão do objeto de manifesto - opcional) insira o ID de versão do objeto de manifesto se quiser usar uma versão que não seja a mais recente.
6. Escolha Next (Próximo).
7. Para usar a função do Lambda para transcodificar todos os objetos listados no arquivo `manifest.json` selecionado, em Operation type (Tipo de operação), escolha Invoke AWS Lambda function (Invocar função do IAM).
8. Na seção Invoke Lambda Function (Invocar função do Lambda), faça o seguinte:
 - a. Escolha Choose from functions in your account (Escolher das funções de sua conta).
 - b. Para Lambda function (Função do Lambda), escolha a função do Lambda que você criou na [Etapa 4 \(p. 76\)](#) (por exemplo, **tutorial-lambda-convert**).
 - c. Para Lambda function version (Versão da função do Lambda), mantenha o valor padrão \$LATER.
9. Escolha Next (Próximo). A página Configure additional options (Configurar opções adicionais) se abre.
10. Na seção Additional options (Opções adicionais), mantenha as configurações padrão.

Para obter mais informações sobre essas opções, consulte [Elementos da solicitação de trabalho de Operações em lote \(p. 946\)](#).

11. Na seção Completion report (Relatório de conclusão) para Path to completion report destination (Caminho para o destino do relatório de conclusão), escolha Browse S3 (Procurar no S3). Encontre o nome do bucket para armazenar os arquivos de mídia de saída que você criou na [Etapa 1 \(p. 72\)](#) (por exemplo, **tutorial-bucket-2**). Escolha o botão de opção ao lado do nome desse bucket. Em seguida, escolha Choose path (Escolher caminho).

Para as demais configurações de Completion report (Relatório de conclusão), mantenha os padrões. Para obter mais informações sobre conclusão de configurações do relatório, consulte [Elementos da solicitação de trabalho de Operações em lote \(p. 946\)](#). Um relatório de conclusão mantém um registro dos detalhes do trabalho e das operações executadas.

12. Em Permissions (Permissões), escolha Choose from existing IAM roles (Escolher entre as funções do IAM existentes). Para IAM role (Função do IAM), escolha a função do IAM para seu trabalho de operações em lote do S3 que você criou na [Etapa 6 \(p. 91\)](#) (por exemplo, **tutorial-s3batch-role**).

13. Escolha Next (Próximo).
14. Na página Review (Revisão), revise as configurações. Depois, escolha Create Job (Criar trabalho).

Depois que o S3 termina de ler o manifesto do trabalho de operações em lote do S3, ele define o Status do trabalho como Awaiting your confirmation to run (Aguardando sua confirmação para ser executado). Para ver as atualizações do status do trabalho, atualize a página. Você não pode executar seu trabalho até que o status dele seja Awaiting your confirmation to run (Aguardando sua confirmação para ser executado).

Execute o trabalho de operações em lote do S3 para chamar a função Lambda

Execute seu trabalho de operações em lote para chamar sua função Lambda para transcodificação de vídeo. Se o trabalho falhar, você pode conferir o relatório da conclusão para identificar a causa.

Para executar o trabalho de operações em lote do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação esquerdo, escolha Batch Operations (Operações em lote).
3. Na lista Jobs (Trabalhos), escolha a opção Job ID (ID do trabalho) na primeira linha, que é o trabalho das operações em lote do S3 que você criou antes.
4. Escolha Run job (Executar trabalho).
5. Revise os parâmetros do job novamente e confirme se o valor para Total objects listed in manifest (Total de objeto listados no manifesto) é o mesmo do número de objetos no manifesto. Em seguida, selecione Run job (Executar trabalho).

A página de trabalho de operações em lote do S3 é aberta.

6. Depois que o trabalho começar a execução, em sua página de trabalho em Status, verifique o andamento de seu trabalho de operações em lote do S3, como Status, % Complete (% de conclusão), Total succeeded (rate) (Total com êxito [taxa]), Total failed (rate) (Total com falha [taxa]), Date terminated (Data de término) e Reason for termination (Motivo do término).

Quando o trabalho de operações em lote do S3 for concluído, visualize os dados na página do trabalho para confirmar que o trabalho foi concluído conforme esperado.

Se mais de 50% das operações de objeto do trabalho das operações em lote do S3 falharem após mais de 1.000 tentativas, o trabalho automaticamente falha. Para conferir o relatório de conclusão para identificar a causa das falhas, consulte o procedimento opcional abaixo.

(Opcional) Verificar seu relatório de conclusão

Você pode usar o relatório de conclusão para determinar quais objetos falharam e a causa das falhas.

Para conferir o relatório de conclusão para obter detalhes sobre os objetos que falharam

1. Na página do trabalho de operações em lote do S3, role para baixo até a seção Completion report (Relatório de conclusão) e escolha o link em Completion report destination (Destino do relatório de conclusão).

A página do bucket de destino de saída do S3 se abre.

2. Na guia Objects (Objetos), escolha a pasta que tem um nome que termina com o ID do trabalho de operações em lote do S3 que você criou antes.

3. Escolha results/ (resultados/).
4. Marque a caixa de seleção próxima ao arquivo .csv.
5. Para visualizar o relatório de trabalhos, escolha Open (Abrir) ou Download (Baixar).

(Opcional) Monitore cada chamada do Lambda no console do Lambda

Depois que o trabalho de operações em lote do S3 começa a ser executado, o trabalho invoca a função do Lambda para cada objeto de vídeo de entrada. O S3 grava logs de cada chamada do Lambda no CloudWatch Logs. Você pode usar o painel de monitoramento do console do Lambda para monitorar sua função Lambda.

1. Abra o console do AWS Lambda em <https://console.aws.amazon.com/lambda/>.
2. No painel de navegação à esquerda, escolha Functions (Funções).
3. Na lista Functions (Funções), escolha o nome da função do Lambda que você criou na [Etapa 4 \(p. 76\)](#) (por exemplo, **tutorial-lambda-convert**).
4. Escolha a guia Monitor (Monitorar).
5. Em Metrics (Métricas), consulte as métricas de tempo de execução para sua função Lambda.
6. Em Logs, visualize dados de log para cada chamada do Lambda por meio do CloudWatch Logs Insights

Note

Quando você usa as operações em lote do S3 com uma função Lambda, essa função Lambda é chamada em cada objeto. Se o trabalho de operações em lote do S3 for grande, ele poderá chamar várias funções Lambda ao mesmo tempo, causando um pico na simultaneidade do Lambda.

Cada Conta da AWS tem uma cota de simultaneidade do Lambda por região. Para obter mais informações, consulte [Escalabilidade de função do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda. Uma prática recomendada para usar funções Lambda com operações em lote do S3 é definir um limite de simultaneidade na própria função Lambda. Definir um limite de simultaneidade evita que seu trabalho consuma a maior parte de sua simultaneidade do Lambda e potencialmente restrinja a utilização de outras funções em sua conta. Para obter mais informações, consulte [Gerenciamento de simultaneidade reservada do Lambda](#) no Guia do desenvolvedor do AWS Lambda.

(Opcional) Monitore cada trabalho de transcodificação de vídeo MediaConvert no console do MediaConvert

Um trabalho do MediaConvert se encarrega da transcodificação de um arquivo de mídia. Quando o trabalho de operações em lote do S3 invoca sua função do Lambda para cada vídeo, cada invocação da função do Lambda cria um trabalho de transcodificação do MediaConvert para cada vídeo de entrada.

1. Faça login no AWS Management Console e abra o console do MediaConvert em <https://console.aws.amazon.com/mediaconvert/>.
2. Se a página introdutória MediaConvert for exibida, escolha Get started (Conceitos básicos).
3. Na lista de Jobs (Trabalhos), visualize cada linha para monitorar a tarefa de transcodificação para cada vídeo de entrada.
4. Identifique a linha de um trabalho que você deseja conferir e escolha o link Job ID (ID do trabalho) para abrir a página de detalhes do trabalho.

5. Na página Job summary (Resumo do trabalho) em Outputs (Saídas), escolha o link para a saída HLS, MP4 ou miniaturas, dependendo do que é suportado pelo navegador, para ir para o bucket de destino do S3 para os arquivos de mídia de saída.
6. Na pasta correspondente (HLS, MP4 ou Thumbnails [Miniaturas]) do bucket de destino de saída do S3, escolha o nome do objeto de arquivo de mídia de saída.
A página de details do objeto se abre.
7. Na página de detalhes do objeto, em Object overview (Visão geral do objeto), escolha o link em Object URL (URL do objeto) para assistir ao arquivo de mídia de saída transcodificado.

Etapa 8: Conferir os arquivos de mídia de saída do bucket de destino do S3

Para conferir os arquivos de mídia de saída do bucket de destino do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, escolha o nome do bucket de destino do S3 para os arquivos de mídia de saída que você criou na [Etapa 1 \(p. 72\)](#) (por exemplo, **tutorial-bucket-2**).
4. Na guia Objects (Objetos), cada vídeo de entrada tem uma pasta com o nome do vídeo de entrada. Cada pasta contém os arquivos de mídia de saída transcodificados para um vídeo de entrada.

Para verificar os arquivos de mídia de saída para um vídeo de entrada, faça o seguinte:

- a. Escolha a pasta com o nome do vídeo de entrada que você deseja conferir.
- b. Escolha a pasta Default/ (Padrão).
- c. Escolha a pasta para um formato transcodificado (HLS, MP4 ou miniaturas neste tutorial).
- d. Escolha o nome do arquivo de mídia de saída.
- e. Para ver o arquivo transcodificado, na página de detalhes do objeto, escolha o link em Object URL (URL do objeto).

Os arquivos de mídia de saída no formato HLS são divididos em segmentos curtos. Para reproduzir esses vídeos, incorpore o URL do objeto do arquivo .m3u8 em um player compatível.

Etapa 9: limpeza

Se você transcodificou vídeos usando as operações em lote do S3, o Lambda e o MediaConvert somente como um exercício de aprendizado, exclua os recursos da AWS alocados para que não haja mais encargos.

Subetapas

- [Exclua a configuração do inventário do S3 para o bucket de origem do S3 \(p. 98\)](#)
- [Excluir a função Lambda \(p. 98\)](#)
- [Excluir o grupo de logs do CloudWatch \(p. 98\)](#)
- [Exclua as funções do IAM junto com as políticas em linha das funções do IAM \(p. 98\)](#)
- [Exclua a política do IAM gerenciada pelo cliente \(p. 99\)](#)
- [Esvaziar os buckets do S3 \(p. 99\)](#)
- [Excluir os buckets do S3 \(p. 99\)](#)

Exclua a configuração do inventário do S3 para o bucket de origem do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista de Buckets, selecione o nome do bucket de origem (por exemplo, **tutorial-bucket-1**).
4. Escolha a guia Management.
5. Em Inventory configurations (Configurações do inventário), escolha o botão de opção ao lado da configuração de inventário que você criou na [Etapa 5 \(p. 88\)](#) (por exemplo, **tutorial-inventory-config**).
6. Escolha Delete (Excluir) e, em seguida, escolha Confirm (Confirmar).

Excluir a função Lambda

1. Abra o console do AWS Lambda em <https://console.aws.amazon.com/lambda/>.
2. No painel de navegação à esquerda, escolha Functions (Funções).
3. Marque a caixa de seleção ao lado da função que você criou na [Etapa 4 \(p. 76\)](#) (por exemplo, **tutorial-lambda-convert**).
4. Escolha Actions (Ações) e, em seguida, escolha Delete (Excluir).
5. Na caixa de diálogo Delete function (Excluir função), escolha Delete (Excluir).

Excluir o grupo de logs do CloudWatch

1. Abra o console do CloudWatch em <https://console.aws.amazon.com/cloudwatch/>.
2. No painel de navegação à esquerda, escolha Logs e, em seguida, escolha Log groups (Grupos de log).
3. Marque a caixa de seleção ao lado do grupo de logs que tem um nome que termina com a função do Lambda que você criou na [Etapa 4 \(p. 76\)](#) (por exemplo, **tutorial-lambda-convert**).
4. Escolha Actions (Ações) e Delete log group(s) (Excluir grupo(s) de log).
5. Na caixa de diálogo Delete log group(s) (Excluir grupo(s) de logs), escolha Delete (Excluir).

Exclua as funções do IAM junto com as políticas em linha das funções do IAM

Para excluir as funções do IAM criadas na [Etapa 2 \(p. 74\)](#), [Etapa 3 \(p. 74\)](#) e [Etapa 6 \(p. 91\)](#), faça o seguinte:

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação da esquerda, escolha Roles (Funções) e marque as caixas de seleção ao lado do nome da função que você deseja excluir.
3. Na parte superior da página, escolha Delete (Excluir).
4. Na caixa de diálogo de confirmação, insira a resposta necessária no campo de entrada de texto com base no prompt e escolha Delete (Excluir).

Exclua a política do IAM gerenciada pelo cliente

Para excluir a política do IAM gerenciada pelo cliente que você criou na [Etapa 6 \(p. 91\)](#), faça o seguinte:

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação à esquerda, escolha Policies (Políticas).
3. Escolha o botão de opção ao lado da política que você criou na [Etapa 6 \(p. 91\)](#) (por exemplo, **tutorial-s3batch-policy**). Você pode usar a caixa de pesquisa para filtrar a lista de políticas.
4. Escolha Actions (Ações) e, em seguida, escolha Delete (Excluir).
5. Confirme que deseja excluir esta política, inserindo o nome da política no campo de texto, e escolha Delete (Excluir).

Esvaziar os buckets do S3

Para esvaziar os buckets do S3 criados em [Pré-requisitos \(p. 72\)](#), [Etapa 1 \(p. 72\)](#) e [Etapa 5 \(p. 88\)](#), faça o seguinte:

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, escolha o botão de opção ao lado do nome do bucket que você deseja esvaziar e depois escolha Empty (Esvaziar).
4. Na página Empty bucket (Esvaziar bucket), confirme se deseja esvaziar o bucket inserindo **permanently delete** no campo de texto e depois escolha Empty (Esvaziar).

Excluir os buckets do S3

Para excluir os buckets do S3 criados em [Pré-requisitos \(p. 72\)](#), [Etapa 1 \(p. 72\)](#) e [Etapa 5 \(p. 88\)](#), faça o seguinte:

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, escolha o botão de opção ao lado do nome do bucket que você deseja excluir.
4. Escolha Delete (Excluir).
5. Na página Delete bucket (Excluir bucket), confirme se deseja excluir o bucket inserindo o nome do bucket no campo de texto e escolha Delete bucket (Excluir bucket).

Next steps (Próximas etapas)

Depois de concluir este tutorial, você poderá explorar outros casos de uso relevantes:

- Você pode usar o Amazon CloudFront para transmitir os arquivos de mídia transcodificados para espectadores em todo o mundo. Para obter mais informações, consulte [Tutorial: Hospedagem de transmissão sob demanda com o Amazon S3, Amazon CloudFront e Amazon Route 53 \(p. 56\)](#).
- Você pode transcodificar vídeos no momento em que os carrega para o bucket de origem do S3. Para fazer isso, você pode configurar um acionador de evento do Amazon S3 que invoca automaticamente a função do Lambda para transcodificar novos objetos no S3 com o MediaConvert. Para obter mais informações, consulte o [Tutorial: Uso de um acionador do Amazon S3 para invocar uma função do Lambda](#) no Guia do desenvolvedor do AWS Lambda.

Tutorial: configurar um site estático no Amazon S3

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: Criar um bucket \(p. 100\)](#)
- [Etapa 2: Habilitar hospedagem de site estático \(p. 100\)](#)
- [Etapa 3: editar as configurações do Bloqueio de acesso público \(p. 101\)](#)
- [Etapa 4: Adicionar política de bucket que torna o conteúdo do bucket publicamente disponível \(p. 102\)](#)
- [Etapa 5: Configurar um documento de índice \(p. 103\)](#)
- [Etapa 6: configurar um documento de erros \(p. 104\)](#)
- [Etapa 7: testar o endpoint do site \(p. 105\)](#)
- [Etapa 8: Limpar \(p. 105\)](#)

Etapa 1: Criar um bucket

As instruções a seguir fornecem uma visão geral de como criar seus buckets para hospedagem de sites. Para obter instruções detalhadas passo a passo sobre como criar um bucket, consulte [Criação de um bucket \(p. 129\)](#).

Para criar um bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Selecione Create bucket (Criar bucket).
3. Insira o Bucket name (Nome do bucket) (por exemplo, `example.com`).
4. Selecione a região onde você deseja criar o bucket.

Escolha uma região próxima de você para minimizar a latência e os custos ou atender a requisitos normativos. A região escolhida determina o endpoint de site do Amazon S3. Para obter mais informações, consulte [Endpoints de site \(p. 1188\)](#).

5. Para aceitar as configurações padrão e criar o bucket, escolha Create (Criar).

Etapa 2: Habilitar hospedagem de site estático

Depois de criar um bucket, você pode habilitar a hospedagem de site estático para seu bucket. Você pode criar um bucket novo ou usar um existente.

Como habilitar a hospedagem de sites estáticos

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket para o qual você deseja habilitar a hospedagem de site estático.
3. Escolha Properties (Propriedades).
4. Em Static website hosting (Hospedagem estática de sites), escolha Edit (Editar).

5. Escolha Use this bucket to host a website (Usar este bucket para hospedar um site).
6. Em Static website hosting (Hospedagem estática de sites), escolha Enable (Ativar).
7. Em Index Document (Documento de índice), insira o nome do arquivo do documento de índice, que geralmente é `index.html`.

O nome do documento de índice diferencia letras maiúsculas de minúsculas e deve corresponder exatamente ao nome do arquivo do documento de índice HTML do qual você planeja fazer upload para o bucket do S3. Quando você configura um bucket para hospedagem de site, deve especificar um documento de índice. O Amazon S3 retorna esse documento de índice quando as solicitações são feitas para o domínio raiz ou alguma subpasta. Para obter mais informações, consulte [Configurar um documento de índice \(p. 1194\)](#).

8. Para fornecer seu próprio documento de erros personalizado para erros da classe 4XX, em Error document (Documento de erros), insira o nome de arquivo do documento de erros personalizado.

O nome do documento de erro diferencia letras maiúsculas de minúsculas e deve corresponder exatamente ao nome do arquivo do documento de erro HTML do qual você planeja fazer upload para o bucket do S3. Se você não especificar um documento de erro personalizado e ocorrer um erro, o Amazon S3 retornará um documento de erro HTML padrão. Para obter mais informações, consulte [Configurar um documento de erro personalizado \(p. 1196\)](#).

9. (Opcional) Se você quiser especificar regras avançadas de redirecionamento em Redirection rules (Regras de redirecionamento), use JSON para descrevê-las.

Por exemplo, você pode encaminhar solicitações condicionalmente de acordo com nomes de chave de objeto ou prefixos específicos na solicitação. Para obter mais informações, consulte [Configurar regras de redirecionamento para usar redirecionamentos condicionais avançados \(p. 1204\)](#).

10. Selecione Save changes.

O Amazon S3 permite a hospedagem estática de sites para seu bucket. Na parte inferior da página, em Static website hosting (Hospedagem estática de sites), você verá o endpoint do site do seu bucket.

11. Em Static website hosting (Hospedagem de sites estáticos), anote o Endpoint.

O Endpoint é o endpoint do site do Amazon S3 para o bucket. Depois de concluir a configuração do bucket como um site estático, é possível usar esse endpoint para testar o site.

Etapa 3: editar as configurações do Bloqueio de acesso público

Por padrão, o Amazon S3 bloqueia o acesso público à sua conta e aos seus buckets. Se quiser usar um bucket para hospedar um site estático, use estas etapas para editar as configurações de bloqueio de acesso público.

Warning

Antes de concluir esta etapa, revise [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#) para garantir que você entenda e aceite os riscos envolvidos em permitir o acesso público. Ao desativar as configurações de bloqueio de acesso público para tornar seu bucket público, qualquer pessoa na Internet pode acessá-lo. Recomendamos que você bloquee todo o acesso público aos buckets.

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Escolha o nome do bucket configurado como um site estático.
3. Escolha Permissions (Permissões).
4. Em Block public access (bucket settings) (Bloqueio de acesso público (configurações de bucket)), escolha Edit (Editar).

- Desmarque Block all public access (Bloquear todo acesso público) e escolha Save changes (Salvar alterações).

Warning

Antes de concluir esta etapa, revise [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#) para garantir que você entenda e aceite os riscos envolvidos em permitir o acesso público. Ao desativar as configurações de bloqueio de acesso público para tornar seu bucket público, qualquer pessoa na Internet pode acessá-lo. Recomendamos que você bloquee todo o acesso público aos buckets.

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)



Account settings for Block Public Access are currently turned on

Account settings for Block Public Access that are enabled apply even if they are disabled for this bucket.

Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

Block public access to buckets and objects granted through new access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

Block public access to buckets and objects granted through any access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

Block public access to buckets and objects granted through new public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

Block public and cross-account access to buckets and objects through any public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

O Amazon S3 desativa as configurações do Bloqueio de acesso público para seu bucket. Para criar um site público e estático, você também pode ter que [editar as configurações de Bloqueio de acesso público](#) para sua conta antes de adicionar uma política de bucket. Se as configurações da conta para bloquear acesso público estiverem ativadas no momento, você verá uma observação em Block public access (bucket settings) (Bloqueio de acesso público (configurações de bucket)).

Etapa 4: Adicionar política de bucket que torna o conteúdo do bucket publicamente disponível

Depois de editar as configurações do Bloqueio de acesso público do S3, é possível adicionar uma política de bucket para conceder acesso público de leitura ao bucket. Ao conceder um acesso público de leitura, qualquer pessoa na Internet pode acessar seu bucket.

Important

A política a seguir é somente um exemplo e concede acesso total aos conteúdos do bucket. Antes de prosseguir com esta etapa, revise [Como posso proteger os arquivos no meu bucket do Amazon S3?](#) para garantir que você entende as práticas recomendadas a fim de proteger os arquivos no bucket do S3 e os riscos envolvidos na concessão de acesso público.

1. Em Buckets, escolha o nome do seu bucket.
2. Escolha Permissions (Permissões).
3. Em Bucket Policy (Política de bucket), escolha Edit (Editar).
4. Para conceder acesso público de leitura ao site, copie a política de bucket a seguir e cole-a no Bucket policy editor (Editor de política de bucket).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "PublicReadGetObject",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::Bucket-Name/*"  
            ]  
        }  
    ]  
}
```

5. Atualize o Resource para o nome do bucket.

Na política de bucket do exemplo anterior, **Bucket-Name** é um espaço reservado para o nome do bucket. Para usar essa política de bucket com seu próprio bucket, você deve atualizar esse nome para corresponder ao nome do seu bucket.

6. Selecione Save changes.

Uma mensagem é exibida indicando que a política de bucket foi adicionada com sucesso.

Se você vir um erro que diz `Policy has invalid resource`, confirme se o nome do bucket na política de bucket corresponde ao nome do seu bucket. Para obter informações sobre como adicionar uma política de bucket, consulte [Como adicionar uma política de bucket do S3?](#)

Se você receber uma mensagem de erro e não puder salvar a política do bucket, verifique suas configurações de acesso público para confirmar que você permite acesso público ao bucket.

Etapa 5: Configurar um documento de índice

Quando você habilita a hospedagem de sites estáticos para seu bucket, insere o nome do documento de índice (por exemplo, `index.html`). Depois de habilitar a hospedagem de sites estáticos para seu bucket, faça upload de um arquivo HTML com esse nome de documento de índice para o bucket.

Como configurar o documento de índice

1. Criar um arquivo `index.html`

Se você não tiver um arquivo `index.html`, poderá usar o HTML a seguir para criar um:

```
<html xmlns="http://www.w3.org/1999/xhtml" >  
<head>  
    <title>My Website Home Page</title>  
</head>  
<body>  
    <h1>Welcome to my website</h1>  
    <p>Now hosted on Amazon S3!</p>
```

```
</body>  
</html>
```

2. Salve o arquivo de índice localmente.

O nome do documento de índice deve corresponder exatamente ao nome do documento de índice que você inseriu na caixa de diálogo Hospedagem de site estático. O nome do documento de índice diferencia maiúsculas de minúsculas. Por exemplo, se você inserir `index.html` no nome do Documento de índice na caixa de diálogo Hospedagem de site estático, o nome do arquivo do documento de índice também deverá ser `index.html` e não `Index.html`.

3. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
4. Na lista Buckets, selecione o nome do bucket que você deseja usar para hospedar um site estático.
5. Habilite a hospedagem de sites estáticos para seu bucket e insira o nome exato do documento de índice (por exemplo, `index.html`). Para obter mais informações, consulte [Habilitar a hospedagem de sites \(p. 1190\)](#).

Depois de habilitar a hospedagem estática do site, vá para a etapa 6.

6. Para fazer upload do documento de índice para o bucket, siga um destes procedimentos:

- Arraste e solte o arquivo de índice na listagem de buckets do console.
- Escolha Upload (Fazer upload) e siga as instruções para escolher e fazer upload do arquivo de índice.

Para obter instruções detalhadas, consulte [Fazer upload de objetos \(p. 171\)](#).

7. (Opcional) Faça upload do conteúdo de outro site para o seu bucket.

Etapa 6: configurar um documento de erros

Ao habilitar a hospedagem de site estático para o bucket, insira o nome do documento de erro (por exemplo, `404.html`). Depois de habilitar a hospedagem de sites estáticos para seu bucket, faça upload de um arquivo HTML com esse nome de documento de erros para o bucket.

Para configurar um documento de erros

1. Crie um documento de erro, por exemplo `404.html`.
2. Salve o arquivo de documento de erros localmente.

O nome do documento de erros diferencia maiúsculas e minúsculas e deve corresponder exatamente ao nome que você insere ao habilitar a hospedagem estática do site. Por exemplo, se você inserir `404.html` como o nome do Error document (Documento de erro) na caixa de diálogo Static website hosting (Hospedagem de site estático), o nome de arquivo do documento de erro também deve ser `404.html`.

3. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
4. Na lista Buckets, selecione o nome do bucket que você deseja usar para hospedar um site estático.
5. Habilite a hospedagem de site estático para seu bucket e insira o nome exato do documento de erro (por exemplo, `404.html`). Para obter mais informações, consulte [Habilitar a hospedagem de sites \(p. 1190\)](#).

Depois de habilitar a hospedagem estática do site, vá para a etapa 6.

6. Para fazer upload do documento de erros para o bucket, siga um destes procedimentos:

- Arraste e solte o arquivo de documento de erros na lista de buckets do console.

- Escolha Upload (Fazer upload) e siga as instruções para escolher e fazer upload do arquivo de índice.

Para obter instruções detalhadas, consulte [Fazer upload de objetos \(p. 171\)](#).

Etapa 7: testar o endpoint do site

Depois de configurar a hospedagem de site estático para seu bucket, você pode testar o endpoint do site.

Note

O Amazon S3 não oferece suporte para o acesso HTTPS ao site. Se quiser usar HTTPS, você poderá usar o Amazon CloudFront para servir um site estático hospedado no Amazon S3.

Para obter mais informações, consulte [Como uso o CloudFront para veicular um site estático hospedado no Amazon S3?](#) e [Exigir HTTPS para comunicação entre visualizadores e CloudFront](#).

1. Em Buckets, escolha o nome do seu bucket.
2. Escolha Properties (Propriedades).
3. Na parte inferior da página, em Static website hosting (Hospedagem estática de sites), escolha seu Bucket website endpoint (Endpoint de site do Bucket).

Seu documento de índice é aberto em uma janela separada do navegador.

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 [Configurar um site estático usando um domínio personalizado registrado no Route 53 \(p. 105\)](#).

Etapa 8: Limpar

Se tiver criado o site apenas como exercício de aprendizado, você poderá excluir os recursos da AWS alocaados, de maneira que deixe de acumular cobranças. Depois que você excluir os recursos da AWS, o site deixará de estar disponível. Para obter mais informações, consulte [Excluir um bucket \(p. 139\)](#).

Configurar um site estático usando um domínio personalizado registrado no Route 53

Suponha que você queira hospedar um site estático no Amazon S3. Você registrou um domínio no Amazon Route 53 (por exemplo, `example.com`) e deseja que solicitações `http://www.example.com` e `http://example.com` sejam veiculadas a partir do conteúdo do Amazon S3. É possível usar essa demonstração para saber como hospedar um site estático e criar redirecionamentos no Amazon S3 para um site com um nome de domínio personalizado que é registrado com o Route 53. É possível trabalhar com um site existente que você deseja hospedar no Amazon S3 ou usar esta demonstração para começar do zero.

Depois de concluir esta demonstração, você tem a opção de usar o Amazon CloudFront para melhorar a performance do seu site. Para obter mais informações, consulte [Acelerar seu site com o Amazon CloudFront \(p. 118\)](#).

Note

Os endpoints de site do Amazon S3 não oferecem suporte a HTTPS ou pontos de acesso. Se quiser usar HTTPS, você poderá usar o Amazon CloudFront para servir um site estático hospedado no Amazon S3.

Para obter mais informações, consulte [Como uso o CloudFront para veicular um site estático hospedado no Amazon S3?](#) e [Exigir HTTPS para comunicação entre visualizadores e CloudFront.](#)

Automatização da configuração de site estático com um modelo do AWS CloudFormation

Você pode usar um modelo do AWS CloudFormation para automatizar a configuração do site estático. O modelo do AWS CloudFormation configura os componentes que você precisa para hospedar um site estático seguro e se concentrar mais no conteúdo do seu site e menos na configuração de componentes.

O modelo do AWS CloudFormation inclui os seguintes componentes:

- Amazon S3: cria um bucket do Amazon S3 para hospedar seu site estático.
- CloudFront: cria uma distribuição do CloudFront para acelerar seu site estático.
- Lambda@Edge: usa o [Lambda@Edge](#) para adicionar cabeçalhos de segurança a cada resposta do servidor. Os cabeçalhos de segurança são um grupo de cabeçalhos na resposta do servidor web que dizem aos navegadores da web para tomarem precauções de segurança extras. Para obter mais informações, consulte a publicação do blog: [Adding HTTP security headers using Lambda@Edge and Amazon CloudFront.](#)

Este modelo do AWS CloudFormation está disponível para download e uso. Para obter informações e instruções, consulte [Conceitos básicos de um site estático seguro](#) no Guia do desenvolvedor do Amazon CloudFront.

Tópicos

- [Antes de começar \(p. 106\)](#)
- [Etapa 1: Registrar um domínio personalizado no Route 53 \(p. 107\)](#)
- [Etapa 2: Criar dois buckets \(p. 107\)](#)
- [Etapa 3: Configurar o bucket de domínio raiz para hospedagem de sites \(p. 108\)](#)
- [Etapa 4: Configurar o bucket de subdomínio para redirecionamento de sites \(p. 109\)](#)
- [Etapa 5: Configurar o registro em log para o tráfego do site \(p. 109\)](#)
- [Etapa 6: Fazer upload do conteúdo do site e do índice \(p. 110\)](#)
- [Etapa 7: carregar um documento de erros \(p. 111\)](#)
- [Etapa 8: Editar configurações do S3 Block Public Access \(p. 111\)](#)
- [Etapa 9: Anexar uma política de bucket \(p. 113\)](#)
- [Etapa 10: Testar o endpoint de domínio \(p. 114\)](#)
- [Etapa 11: Adicionar registros de alias para seu domínio e subdomínio \(p. 114\)](#)
- [Etapa 12: Testar o site \(p. 117\)](#)
- [Acelerar seu site com o Amazon CloudFront \(p. 118\)](#)
- [Limpar seus recursos de exemplo \(p. 121\)](#)

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. O exemplo mostra como criar registros de alias do Route 53 que

roteiam o tráfego para do domínio (`example.com`) e do 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 personalizado no Route 53

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 seu nome de domínio, é possível criar e configurar seus buckets do Amazon S3 para hospedagem de sites.

Etapa 2: Criar dois buckets

Para oferecer suporte a solicitações no domínio raiz e no subdomínio, crie dois buckets.

- Bucket de domínio – `example.com`
- Bucket de subdomínio – `www.example.com`

Esses nomes de bucket devem corresponder exatamente ao seu nome de domínio. Neste exemplo, o nome de domínio é `example.com`. Você hospeda seu conteúdo fora do bucket de domínio raiz (`example.com`). Crie uma solicitação de redirecionamento para o bucket de subdomínio (`www.example.com`). Em outras palavras, se uma pessoa insere `www.example.com` no navegador, ela é redirecionada para `example.com` e vê o conteúdo hospedado no bucket do Amazon S3 com esse nome.

Como criar buckets para hospedagem de sites

As instruções a seguir fornecem uma visão geral de como criar seus buckets para hospedagem de sites. Para obter instruções detalhadas passo a passo sobre como criar um bucket, consulte [Criação de um bucket \(p. 129\)](#).

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Crie o bucket do domínio raiz:

- a. Selecione Create bucket (Criar bucket).
- b. Insira o Bucket name (Nome do bucket) (por exemplo, `example.com`).
- c. Selecione a região onde você deseja criar o bucket.

Escolha uma região próxima de você para minimizar a latência e os custos ou atender a requisitos normativos. A região escolhida determina o endpoint de site do Amazon S3. Para obter mais informações, consulte [Endpoints de site \(p. 1188\)](#).

- d. Para aceitar as configurações padrão e criar o bucket, escolha Create (Criar).
3. Crie o bucket de subdomínio:

- a. Selecione Create bucket (Criar bucket).
- b. Insira o Bucket name (Nome do bucket) (por exemplo, `www.example.com`).
- c. Selecione a região onde você deseja criar o bucket.

Escolha uma região próxima de você para minimizar a latência e os custos ou atender a requisitos normativos. A região escolhida determina o endpoint de site do Amazon S3. Para obter mais informações, consulte [Endpoints de site \(p. 1188\)](#).

- d. Para aceitar as configurações padrão e criar o bucket, escolha Create (Criar).

Na próxima etapa, configure `example.com` para a hospedagem do site.

Etapa 3: Configurar o bucket de domínio raiz para hospedagem de sites

Nesta etapa, você configura o bucket de domínio raiz (`example.com`) como um site. Esse bucket terá o conteúdo do site. Ao configurar um bucket para hospedagem de sites, é possível acessar o site usando o [Endpoints de site \(p. 1188\)](#).

Como habilitar a hospedagem de sites estáticos

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket para o qual você deseja habilitar a hospedagem de site estático.
3. Escolha Properties (Propriedades).
4. Em Static website hosting (Hospedagem estática de sites), escolha Edit (Editar).
5. Escolha Use this bucket to host a website (Usar este bucket para hospedar um site).
6. Em Static website hosting (Hospedagem estática de sites), escolha Enable (Ativar).
7. Em Index Document (Documento de índice), insira o nome do arquivo do documento de índice, que geralmente é `index.html`.

O nome do documento de índice diferencia letras maiúsculas de minúsculas e deve corresponder exatamente ao nome do arquivo do documento de índice HTML do qual você planeja fazer upload para o bucket do S3. Quando você configura um bucket para hospedagem de site, deve especificar um documento de índice. O Amazon S3 retorna esse documento de índice quando as solicitações são feitas para o domínio raiz ou alguma subpasta. Para obter mais informações, consulte [Configurar um documento de índice \(p. 1194\)](#).

8. Para fornecer seu próprio documento de erros personalizado para erros da classe 4XX, em Error document (Documento de erros), insira o nome de arquivo do documento de erros personalizado.

O nome do documento de erro diferencia letras maiúsculas de minúsculas e deve corresponder exatamente ao nome do arquivo do documento de erro HTML do qual você planeja fazer upload para o bucket do S3. Se você não especificar um documento de erro personalizado e ocorrer um erro, o Amazon S3 retornará um documento de erro HTML padrão. Para obter mais informações, consulte [Configurar um documento de erro personalizado \(p. 1196\)](#).

9. (Opcional) Se você quiser especificar regras avançadas de redirecionamento em Redirection rules (Regras de redirecionamento), use JSON para descrevê-las.

Por exemplo, você pode encaminhar solicitações condicionalmente de acordo com nomes de chave de objeto ou prefixos específicos na solicitação. Para obter mais informações, consulte [Configurar regras de redirecionamento para usar redirecionamentos condicionais avançados \(p. 1204\)](#).

10. Selecione Save changes.

O Amazon S3 permite a hospedagem estática de sites para seu bucket. Na parte inferior da página, em Static website hosting (Hospedagem estática de sites), você verá o endpoint do site do seu bucket.

11. Em Static website hosting (Hospedagem de sites estáticos), anote o Endpoint.

O Endpoint é o endpoint do site do Amazon S3 para o bucket. Depois de concluir a configuração do bucket como um site estático, é possível usar esse endpoint para testar o site.

Depois de [editar as configurações de acesso público de bloqueio](#) e [adicionar uma política de bucket](#) que permita acesso público de leitura, você pode usar o endpoint do site para acessar seu site.

Na próxima etapa, configure o subdomínio (www.example.com) para redirecionar solicitações para o domínio (example.com).

Etapa 4: Configurar o bucket de subdomínio para redirecionamento de sites

Depois de configurar o bucket de domínio raiz para a hospedagem de sites, é possível configurar o bucket do subdomínio para redirecionar todas as solicitações para o domínio. Neste exemplo, todas as solicitações para www.example.com são redirecionadas para example.com.

Para configurar uma solicitação de redirecionamento

1. No console do Amazon S3, na lista Buckets, escolha o nome do bucket do subdomínio (www.example.com neste exemplo).
2. Escolha Properties (Propriedades).
3. Em Static website hosting (Hospedagem estática de sites), escolha Edit (Editar).
4. Selecione Redirect requests for an object (Redirecionar solicitações de um objeto).
5. Na caixa Target bucket (Bucket de destino), insira o domínio raiz (por exemplo, example.com).
6. Em Protocol (Protocolo), selecione http.
7. Selecione Save changes.

Etapa 5: Configurar o registro em log para o tráfego do site

Se quiser rastrear o número de visitantes que acessam seu site, opcionalmente você pode habilitar o log para seu bucket de domínio raiz. Para obter mais informações, consulte [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#). Se planeja usar o Amazon CloudFront para acelerar seu site, você também pode usar o registro em log do CloudFront.

Como habilitar o registro em log do acesso ao servidor para o bucket de domínio raiz

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na mesma região onde você criou o bucket configurado como um site estático, crie um bucket para registro em log, por exemplo logs.example.com.
3. Crie uma pasta para os arquivos de log do registro em log de acesso ao servidor (por exemplo, [logs](http://logs.example.com/logs)).
4. (Opcional) Se você quiser usar o CloudFront para melhorar a performance do seu site, crie uma pasta para os arquivos de log do CloudFront (por exemplo, [cdn](http://logs.example.com/cdn)).

Important

Ao criar ou atualizar uma distribuição e habilitar o registro do CloudFront, o CloudFront atualiza a lista de controle de acesso (ACL) do bucket para fornecer à conta `awslogsdelivery` permissões `FULL_CONTROL` para registrar logs em seu bucket. Para obter mais informações, consulte [Permissões necessárias para configurar o registro padrão e acessar seus arquivos de log](#) no Guia do desenvolvedor do Amazon CloudFront. Se o bucket que armazena os logs usar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership para desabilitar ACLs, o CloudFront não poderá gravar logs no bucket. Para obter

mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

5. Na lista Buckets, escolha seu bucket de domínio raiz.
6. Escolha Properties (Propriedades).
7. Em Server access logging (Registro de acesso ao servidor), selecione Edit (Editar).
8. Escolha Habilitar.
9. No Target bucket (Bucket de destino), escolha o destino do bucket e da pasta para os logs de acesso ao servidor:
 - Navegue até o local da pasta e do bucket:
 1. Escolha Browse S3 (Navegar no S3).
 2. Escolha o nome do bucket e, depois, escolha a pasta de logs.
 3. Selecione Choose path (Escolher caminho).
 - Insira o caminho do bucket do S3, por exemplo, s3://logs.example.com/logs/.
10. Selecione Save changes.

No bucket de log, agora você pode acessar seus logs. O Amazon S3 grava os logs de acesso ao site no bucket de log a cada duas horas.

Etapa 6: Fazer upload do conteúdo do site e do índice

Nesta etapa, faça upload do documento de índice e do conteúdo opcional do site no bucket de domínio raiz.

Quando você habilita a hospedagem de sites estáticos para seu bucket, insere o nome do documento de índice (por exemplo, **index.html**). Depois de habilitar a hospedagem de sites estáticos para seu bucket, faça upload de um arquivo HTML com esse nome de documento de índice para o bucket.

Como configurar o documento de índice

1. Criar um arquivo **index.html**

Se você não tiver um arquivo **index.html**, poderá usar o HTML a seguir para criar um:

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <title>My Website Home Page</title>
</head>
<body>
    <h1>Welcome to my website</h1>
    <p>Now hosted on Amazon S3!</p>
</body>
</html>
```

2. Salve o arquivo de índice localmente.

O nome do documento de índice deve corresponder exatamente ao nome do documento de índice que você inseriu na caixa de diálogo Hospedagem de site estático. O nome do documento de índice diferencia maiúsculas de minúsculas. Por exemplo, se você inserir **index.html** no nome do Documento de índice na caixa de diálogo Hospedagem de site estático, o nome do arquivo do documento de índice também deverá ser **index.html** e não **Index.html**.

3. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
4. Na lista Buckets, selecione o nome do bucket que você deseja usar para hospedar um site estático.

5. Habilite a hospedagem de sites estáticos para seu bucket e insira o nome exato do documento de índice (por exemplo, `index.html`). Para obter mais informações, consulte [Habilitar a hospedagem de sites \(p. 1190\)](#).

Depois de habilitar a hospedagem estática do site, vá para a etapa 6.

6. Para fazer upload do documento de índice para o bucket, siga um destes procedimentos:

- Arraste e solte o arquivo de índice na listagem de buckets do console.
- Escolha Upload (Fazer upload) e siga as instruções para escolher e fazer upload do arquivo de índice.

Para obter instruções detalhadas, consulte [Fazer upload de objetos \(p. 171\)](#).

7. (Opcional) Faça upload do conteúdo de outro site para o seu bucket.

Etapa 7: carregar um documento de erros

Ao habilitar a hospedagem de site estático para o bucket, insira o nome do documento de erro (por exemplo, `404.html`). Depois de habilitar a hospedagem de sites estáticos para seu bucket, faça upload de um arquivo HTML com esse nome de documento de erros para o bucket.

Para configurar um documento de erros

1. Crie um documento de erro, por exemplo `404.html`.
2. Salve o arquivo de documento de erros localmente.

O nome do documento de erros diferencia maiúsculas e minúsculas e deve corresponder exatamente ao nome que você insere ao habilitar a hospedagem estática do site. Por exemplo, se você inserir `404.html` como o nome do Error document (Documento de erro) na caixa de diálogo Static website hosting (Hospedagem de site estático), o nome de arquivo do documento de erro também deve ser `404.html`.

3. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
4. Na lista Buckets, selecione o nome do bucket que você deseja usar para hospedar um site estático.
5. Habilite a hospedagem de site estático para seu bucket e insira o nome exato do documento de erro (por exemplo, `404.html`). Para obter mais informações, consulte [Habilitar a hospedagem de sites \(p. 1190\)](#).

Depois de habilitar a hospedagem estática do site, vá para a etapa 6.

6. Para fazer upload do documento de erros para o bucket, siga um destes procedimentos:

- Arraste e solte o arquivo de documento de erros na lista de buckets do console.
- Escolha Upload (Fazer upload) e siga as instruções para escolher e fazer upload do arquivo de índice.

Para obter instruções detalhadas, consulte [Fazer upload de objetos \(p. 171\)](#).

Etapa 8: Editar configurações do S3 Block Public Access

Neste exemplo, você edita configurações de bloqueio de acesso público para o bucket de domínio (`example.com`) para permitir acesso público.

Por padrão, o Amazon S3 bloqueia o acesso público à sua conta e aos seus buckets. Se quiser usar um bucket para hospedar um site estático, use estas etapas para editar as configurações de bloqueio de acesso público.

Warning

Antes de concluir esta etapa, revise [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#) para garantir que você entenda e aceite os riscos envolvidos em permitir o acesso público. Ao desativar as configurações de bloqueio de acesso público para tornar seu bucket público, qualquer pessoa na Internet pode acessá-lo. Recomendamos que você bloquee todo o acesso público aos buckets.

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Escolha o nome do bucket configurado como um site estático.
3. Escolha Permissions (Permissões).
4. Em Block public access (bucket settings) (Bloqueio de acesso público (configurações de bucket)), escolha Edit (Editar).
5. Desmarque Block all public access (Bloquear todo acesso público) e escolha Save changes (Salvar alterações).

Warning

Antes de concluir esta etapa, revise [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#) para garantir que você entenda e aceite os riscos envolvidos em permitir o acesso público. Ao desativar as configurações de bloqueio de acesso público para tornar seu bucket público, qualquer pessoa na Internet pode acessá-lo. Recomendamos que você bloquee todo o acesso público aos buckets.

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)



Account settings for Block Public Access are currently turned on

Account settings for Block Public Access that are enabled apply even if they are disabled for this bucket.

Block **all** public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

Block public access to buckets and objects granted through **new** access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

Block public access to buckets and objects granted through **any** access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

Block public access to buckets and objects granted through **new** public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

Block public and cross-account access to buckets and objects through **any** public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

O Amazon S3 desativa as configurações do Bloqueio de acesso público para seu bucket. Para criar um site público e estático, você também pode ter que [editar as configurações de Bloqueio de acesso público](#) para sua conta antes de adicionar uma política de bucket. Se as configurações da conta para

bloquear acesso público estiverem ativadas no momento, você verá uma observação em Block public access (bucket settings) (Bloqueio de acesso público (configurações de bucket)).

Etapa 9: Anexar uma política de bucket

Neste exemplo, você anexa uma política de bucket ao bucket de domínio (`example.com`) para permitir acesso de leitura pública. Você substitui o **Bucket-Name** na política de bucket de exemplo pelo nome do bucket de domínio, por exemplo, `example.com`.

Depois de editar as configurações do Bloqueio de acesso público do S3, é possível adicionar uma política de bucket para conceder acesso público de leitura ao bucket. Ao conceder um acesso público de leitura, qualquer pessoa na Internet pode acessar seu bucket.

Important

A política a seguir é somente um exemplo e concede acesso total aos conteúdos do bucket. Antes de prosseguir com esta etapa, revise [Como posso proteger os arquivos no meu bucket do Amazon S3?](#) para garantir que você entende as práticas recomendadas a fim de proteger os arquivos no bucket do S3 e os riscos envolvidos na concessão de acesso público.

1. Em Buckets, escolha o nome do seu bucket.
2. Escolha Permissions (Permissões).
3. Em Bucket Policy (Política de bucket), escolha Edit (Editar).
4. Para conceder acesso público de leitura ao site, copie a política de bucket a seguir e cole-a no Bucket policy editor (Editor de política de bucket).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "PublicReadGetObject",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::Bucket-Name/*"  
            ]  
        }  
    ]  
}
```

5. Atualize o Resource para o nome do bucket.

Na política de bucket do exemplo anterior, **Bucket-Name** é um espaço reservado para o nome do bucket. Para usar essa política de bucket com seu próprio bucket, você deve atualizar esse nome para corresponder ao nome do seu bucket.

6. Selecione Save changes.

Uma mensagem é exibida indicando que a política de bucket foi adicionada com sucesso.

Se você vir um erro que diz `Policy has invalid resource`, confirme se o nome do bucket na política de bucket corresponde ao nome do seu bucket. Para obter informações sobre como adicionar uma política de bucket, consulte [Como adicionar uma política de bucket do S3?](#)

Se você receber uma mensagem de erro e não puder salvar a política do bucket, verifique suas configurações de acesso público para confirmar que você permite acesso público ao bucket.

Na próxima etapa, é possível descobrir os endpoints do site e testar o endpoint do domínio.

Etapa 10: Testar o endpoint de domínio

Depois de configurar seu bucket de domínio para hospedar um site público, você pode testar seu endpoint. Para obter mais informações, consulte [Endpoints de site \(p. 1188\)](#). Somente será possível testar o endpoint de seu bucket de domínio porque ele está configurado para redirecionamento de site e não para hospedagem de site estático.

Note

O Amazon S3 não oferece suporte para o acesso HTTPS ao site. Se quiser usar HTTPS, você poderá usar o Amazon CloudFront para servir um site estático hospedado no Amazon S3. Para obter mais informações, consulte [Como uso o CloudFront para veicular um site estático hospedado no Amazon S3?](#) e [Exigir HTTPS para comunicação entre visualizadores e CloudFront](#).

1. Em Buckets, escolha o nome do seu bucket.
2. Escolha Properties (Propriedades).
3. Na parte inferior da página, em Static website hosting (Hospedagem estática de sites), escolha seu Bucket website endpoint (Endpoint de site do Bucket).

Seu documento de índice é aberto em uma janela separada do navegador.

Na próxima etapa, você usa o Amazon Route 53 para permitir que os clientes usem ambos os URLs personalizados para navegar até o site.

Etapa 11: Adicionar registros de alias para seu domínio e subdomínio

Nesta etapa, você cria os registros de alias adicionados à zona hospedada dos mapas de domínio `example.com` e `www.example.com`. 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. Crie dois registros de alias, um para o domínio raiz e um para o subdomínio.

Adicionar um registro de alias para seu domínio raiz e subdomínio

Como adicionar um registro de alias ao domínio raiz (`example.com`)

1. Abra o console do Route 53 em <https://console.aws.amazon.com/route53/>.

Note

Se você ainda não usa o Route 53, consulte [Etapa 1: registrar um domínio](#) no Guia do desenvolvedor do Amazon Route 53. Após concluir a configuração, é possível retomar as instruções.

2. Selecione Hosted zones (Zonas hospedadas).
3. Na lista de zonas hospedadas, escolha o nome da zona hospedada que corresponde ao nome de domínio.
4. Escolha Create record (Criar registro).
5. Escolha Switch to wizard (Alternar para assistente).

Note

Se você quiser usar a criação rápida para criar seus registros de alias, consulte [Configurar o Route 53 para rotear o tráfego para um bucket do S3](#).

6. Escolha Simple routing (Roteamento simples) e Next (Próximo).
7. Escolha Define simple record (Definir registro simples).
8. Em Record name (Nome do registro), aceite o valor padrão, que é o nome da zona hospedada e do domínio.
9. Em Value/Route traffic to (Valor/Encaminhar tráfego para), escolha Alias to S3 website endpoint (Alias para o endpoint do site do S3).
10. Escolha a região .
11. Escolha o bucket do S3.

O nome do bucket deve corresponder ao nome que aparece na caixa Name (Nome). Na lista Choose S3 bucket (Escolher bucket do S3), o nome do bucket aparece com o endpoint do site do Amazon S3 para a região onde o bucket foi criado, por exemplo, `s3-website-us-west-1.amazonaws.com` (`example.com`).

Choose S3 bucket (Escolher bucket do S3) lista um bucket se:

- Você configurou o bucket como um site estático.
- O nome do bucket é o mesmo que o nome do registro que você está criando.
- A Conta da AWS atual criou o bucket.

Se o bucket não aparecer na lista Choose S3 bucket (Escolher bucket do S3), insira o endpoint de site do Amazon S3 da região em que o bucket foi criado, por exemplo, `s3-website-us-west-2.amazonaws.com`. Para obter uma lista completa dos endpoints do site do Amazon S3, consulte [Endpoints de site do Amazon S3](#). Para obter mais informações sobre o destino de alias, consulte [Valor/rotear tráfego para](#) no Guia do desenvolvedor do Amazon Route 53.

12. Em Record type (Tipo de registro), escolha A - Routes traffic to an IPv4 address and some AWS resources (Encaminha o tráfego para um endereço IPv4 e alguns recursos da AWS).
13. Em Evaluate target health (Avaliar integridade do destino), escolha No (Não).
14. Escolha Define simple record (Definir registro simples).

Como adicionar um registro de alias ao subdomínio (`www.example.com`)

1. Em Configure records (Configurar registros), escolha Define simple record (Definir registro simples).
2. Em Record name (Nome do registro) para seu subdomínio, digite `www`.
3. Em Value/Route traffic to (Valor/Encaminhar tráfego para), escolha Alias to S3 website endpoint (Alias para o endpoint do site do S3).
4. Escolha a região .
5. Escolha o bucket do S3, por exemplo, `s3-website-us-west-2.amazonaws.com` (`www.example.com`).

Se o bucket não aparecer na lista Choose S3 bucket (Escolher bucket do S3), insira o endpoint de site do Amazon S3 da região em que o bucket foi criado, por exemplo, `s3-website-us-west-2.amazonaws.com`. Para obter uma lista completa dos endpoints do site do Amazon S3, consulte [Endpoints de site do Amazon S3](#). Para obter mais informações sobre o destino de alias, consulte [Valor/rotear tráfego para](#) no Guia do desenvolvedor do Amazon Route 53.

6. Em Record type (Tipo de registro), escolha A - Routes traffic to an IPv4 address and some AWS resources (Encaminha o tráfego para um endereço IPv4 e alguns recursos da AWS).
7. Em Evaluate target health (Avaliar integridade do destino), escolha No (Não).
8. Escolha Define simple record (Definir registro simples).
9. Na página Configure records (Configurar registros), escolha Create records (Criar registros).

Note

As alterações são geralmente propagadas para todos os servidores do Route 53 dentro de 53 segundos. Quando a propagação for concluída, será possível rotear o tráfego para o bucket do Amazon S3 usando os nomes de registros de alias criados nesse procedimento.

Adicionar um registro de alias para seu domínio raiz e subdomínio (antigo console do Route 53)

Como adicionar um registro de alias ao domínio raiz (example.com)

O console do Route 53 foi reprojetado. No console do Route 53, você pode usar temporariamente o console antigo. Se você optar por trabalhar com o console do Route 53 antigo, use o procedimento abaixo.

1. Abra o console do Route 53 em <https://console.aws.amazon.com/route53/>.

Note

Se você ainda não usa o Route 53, consulte [Etapa 1: registrar um domínio](#) no Guia do desenvolvedor do Amazon Route 53. Após concluir a configuração, é possível retomar as instruções.

2. Selecione Hosted Zones (Zonas hospedadas).
3. Na lista de zonas hospedadas, escolha o nome da zona hospedada que corresponde ao nome de domínio.
4. Escolha Create Record Set (Criar conjunto de registros).
5. Especifique os seguintes valores:

Nome

Aceite o valor padrão, que é o nome da zona hospedada e do domínio.

Para o domínio raiz, não é necessário inserir nenhuma informação adicional no campo Name (Nome).

Tipo

Escolha A - IPv4 address (A – Endereço IPv4).

Alias

Escolha Sim.

Alvo do alias

Na seção S3 website endpoints (Endpoints de site do S3) da lista, escolha o nome do bucket.

O nome do bucket deve corresponder ao nome que aparece na caixa Name (Nome). Na listagem Alias Target (Destino do alias), o nome do bucket é seguido pelo endpoint de site do Amazon S3 para a região onde o bucket foi criado, por exemplo, example.com (`s3-website-us-west-2.amazonaws.com`). Alias Target (Alvo do alias) lista um bucket se:

- Você configurou o bucket como um site estático.
- O nome do bucket é o mesmo que o nome do registro que você está criando.
- A Conta da AWS atual criou o bucket.

Se o bucket não aparecer na listagem Alias Target (Destino do alias), insira o endpoint de site do Amazon S3 da região em que o bucket foi criado, por exemplo, `s3-website-us-west-2`.

Para obter uma lista completa dos endpoints do site do Amazon S3, consulte [Endpoints de site do Amazon S3](#). Para obter mais informações sobre o destino de alias, consulte [Valor/rotear tráfego para](#) no Guia do desenvolvedor do Amazon Route 53.

Política de roteamento

Aceite o valor padrão de Simple (Simples).

Avaliar status do alvo

Aceite o valor padrão de No (Não).

6. Escolha Create (Criar).

Como adicionar um registro de alias ao subdomínio (www.example.com)

1. Na zona hospedada do domínio raiz (example.com), selecione Create Record Set (Criar conjunto de registros).
2. Especifique os seguintes valores:

Nome

Para o subdomínio, insira www na caixa.

Tipo

Escolha A - IPv4 address (A – Endereço IPv4).

Alias

Escolha Sim.

Alvo do alias

Na seção S3 website endpoints (Endpoints de site do S3) da lista, escolha o mesmo nome de bucket que é exibido no campo Name (Nome), por exemplo, www.example.com (s3-website-us-west-2.amazonaws.com).

Política de roteamento

Aceite o valor padrão de Simple (Simples).

Avaliar status do alvo

Aceite o valor padrão de No (Não).

3. Escolha Create (Criar).

Note

As alterações geralmente são propagadas para todos os servidores do Route 53 dentro de 53 segundos. Quando a propagação for concluída, será possível rotear o tráfego para o bucket do Amazon S3 usando os nomes de registros de alias criados nesse procedimento.

Etapa 12: Testar o site

Verifique se o site e o redirecionamento funcionam corretamente. No navegador, insira os URLs. Neste exemplo, é possível tentar os seguintes URLs:

- Domínio (<http://example.com>): exibe o documento de índice no bucket example.com.
- Subdomínio (<http://www.example.com>): redireciona sua solicitação para <http://example.com>. Veja o documento de índice no bucket example.com.

Se o seu site ou links de redirecionamento não funcionarem, tente o seguinte:

- Limpar cache: limpe o cache do seu navegador da Web.

- Verificar servidores de nomes: se a página da Web e os links de redirecionamento não funcionarem depois de limpar o cache, você poderá comparar os servidores de nome do seu domínio e os servidores de nome da sua zona hospedada. Se os servidores de nome não corresponderem, talvez seja necessário atualizar seus servidores de nome de domínio para corresponder aos listados na zona hospedada. Para obter mais informações, consulte [Adicionar ou alterar servidores de nomes e registros para um domínio](#).

Depois de testar com êxito o domínio raiz e o subdomínio, você pode configurar uma distribuição do [Amazon CloudFront](#) para melhorar a performance do seu site e fornecer logs que você possa usar para revisar o tráfego do site. Para obter mais informações, consulte [Acelerar seu site com o Amazon CloudFront \(p. 118\)](#).

Acelerar seu site com o Amazon CloudFront

Você pode usar o [Amazon CloudFront](#) para melhorar a performance do seu site do Amazon S3. O CloudFront disponibiliza os arquivos do site (como HTML, imagens e vídeos) por meio de datacenters em todo o mundo (conhecidos como 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.

Automatizar a configuração com um modelo do AWS CloudFormation

Para obter mais informações sobre como usar um modelo do AWS CloudFormation para configurar um site estático seguro que cria uma distribuição do CloudFront para veicular seu site, consulte [Introdução a um site estático seguro](#) no Guia do desenvolvedor do Amazon CloudFront.

Tópicos

- [Etapa 1: Criar uma distribuição do CloudFront \(p. 118\)](#)
- [Etapa 2: Atualizar os conjuntos de registros do domínio e do subdomínio \(p. 120\)](#)
- [\(Opcional\) Etapa 3: verificar os arquivos de log \(p. 121\)](#)

Etapa 1: Criar uma distribuição do CloudFront

Primeiro, você cria uma distribuição do CloudFront. Isso torna seu site disponível em datacenters em todo o mundo.

Como criar uma distribuição com uma origem do Amazon S3

1. Abra o console do CloudFront em <https://console.aws.amazon.com/cloudfront/v3/home>.
2. Escolha Criar distribuição.
3. Na página Select a delivery method for your content (Selecionar um método de entrega do conteúdo), em 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 do site do Amazon S3 para seu bucket, por exemplo, `example.com.s3-website.us-west-1.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), mantenha os valores padrão definidos.

Com as configurações padrão do Viewer Protocol Policy (Política de protocolo de visualizador), é possível usar HTTPS para o site estático. Para obter mais informações sobre essas opções de configuração, consulte [Valores que você especifica quando cria ou atualiza uma distribuição na 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 performance]).
- b. Defina Alternate Domain Names (CNAMEs) Nomes de domínio alternativos (CNAMEs) para o domínio raiz e para o subdomínio www. Neste tutorial, são example.com e www.example.com.

Important

Antes de executar essa etapa, veja os [requisitos para o uso de nomes de domínio alternativos](#), principalmente a necessidade de um certificado SSL/TLS válido.

- c. Em SSL Certificate (Certificado SSL), selecione Custom SSL Certificate (example.com) (Certificado SSL personalizado (exemplo.com)) e escolha o certificado personalizado que contém os nomes de domínio e subdomínio.

Para obter mais informações, consulte [Certificado SSL](#) no Guia do desenvolvedor do Amazon CloudFront.

- d. Em Default Root Object (Objeto raiz padrão), insira o nome do documento de índice, por exemplo, index.html.

Se o URL usado para acessar a distribuição não contiver um nome de arquivo, a distribuição do CloudFront retornará o documento de índice. O Default Root Object (Objeto raiz padrão) deve corresponder exatamente ao nome do documento de índice do seu site estático. Para obter mais informações, consulte [Configurar um documento de índice \(p. 1194\)](#).

- e. Defina Logging (Registro em log) como On (Ligado).

Important

Ao criar ou atualizar uma distribuição e habilitar o registro do CloudFront, o CloudFront atualiza a lista de controle de acesso (ACL) do bucket para fornecer à conta awslogsdelivery permissões FULL_CONTROL para registrar logs em seu bucket.

Para obter mais informações, consulte [Permissões necessárias para configurar o registro padrão e acessar seus arquivos de log](#) no Guia do desenvolvedor do Amazon CloudFront. Se o bucket que armazena os logs usar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership para desabilitar ACLs, o CloudFront não poderá gravar logs no bucket. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

- f. Em Bucket for Logs (Bucket para logs), escolha o bucket para o registro em log que você criou.

Para obter mais informações sobre como configurar um bucket de log, consulte [\(Opcional\) Registrar em log o tráfego da web \(p. 1202\)](#).

- g. Se quiser armazenar os logs gerados pelo tráfego para a distribuição do CloudFront em uma pasta, em Log Prefix (Prefixo de log), insira o nome da pasta.

- h. Mantenha todas as outras configurações segundo seus valores predefinidos.

7. Escolha Criar distribuição.

8. 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.

9. Registre o valor de Domain Name (Nome do domínio) mostrado no console do CloudFront, por exemplo, `dj4p1rv6mvubz.cloudfront.net`.
10. Para verificar se a distribuição do CloudFront está funcionando, insira o nome de domínio da distribuição em um navegador da Web.

Se o seu site estiver visível, a distribuição do CloudFront funciona. Se o seu site tiver um domínio personalizado registrado no Amazon Route 53, você precisará do nome de domínio do CloudFront para atualizar o conjunto de registros na próxima etapa.

Etapa 2: Atualizar os conjuntos de registros do domínio e do subdomínio

Agora que você criou com sucesso uma distribuição do CloudFront, atualize o registro de alias no Route 53 para apontar para a nova distribuição do CloudFront.

Para atualizar o registro de alias para apontar para uma distribuição do CloudFront

1. Abra o console do Route 53 em <https://console.aws.amazon.com/route53/>.
2. No painel de navegação, escolha Hosted zones (Zonas hospedadas).
3. Na página Hosted Zones (Zonas hospedadas), escolha a zona hospedada criada por você para o subdomínio, por exemplo, `www.example.com`.
4. Em Records (Registros), selecione o registro A que você criou para seu subdomínio.
5. Em Record details (Detalhes do registro), escolha Edit record (Editar registro).
6. Em Route traffic to (Rotear tráfego para), escolha Alias to CloudFront distribution (Alias para distribuição do CloudFront).
7. Em Choose distribution (Escolher distribuição), escolha a distribuição do CloudFront.
8. Escolha Save (Salvar).
9. Para redirecionar o registro A do domínio raiz para a distribuição do CloudFront, repita esse procedimento do domínio raiz, por exemplo, `example.com`.

A atualização dos conjuntos de registros entra em vigor dentro de 2 a 48 horas.

10. Para ver se os novos registros A entraram em vigor, em um navegador da Web, digite o URL do subdomínio, por exemplo, `http://www.example.com`.

Se o navegador não o redirecionar para o domínio raiz (por exemplo, `http://example.com`), os novos registros A estarão no lugar. 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. 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.

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 a aplicação do navegador ou usando outro navegador da Web.

(Opcional) Etapa 3: 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](#).

Os logs do CloudFront são armazenados no bucket e na pasta escolhidos ao criar uma distribuição do CloudFront e habilitar o registro em log. O CloudFront grava logs em seu bucket de log dentro de 24 horas a partir do momento em que as solicitações correspondentes são feitas.

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 nome do bucket de registro em log do site.
3. Escolha a pasta de logs do CloudFront.
4. Baixe os arquivos .gzip escritos pelo CloudFront antes de abri-los.

Se tiver criado o site apenas como um exercício de aprendizado, você poderá excluir os recursos alocados, para não acumular mais cobranças. Para fazer isto, consulte [Limpar seus recursos de exemplo \(p. 121\)](#). Depois que você excluir os recursos da AWS, o site deixará de estar disponível.

Limpar seus recursos de exemplo

Se você tiver criado o site estático apenas como um exercício de aprendizado, exclua os recursos da AWS alocados, para deixar de acumular cobranças. Depois que você excluir os recursos da AWS, o site deixará de estar disponível.

Tarefas

- [Etapa 1: exclua a distribuição do Amazon CloudFront \(p. 121\)](#)
- [Etapa 2: exclua a zona hospedada do Route 53 \(p. 121\)](#)
- [Etapa 3: Desabilitar o registro em log e excluir o bucket do S3 \(p. 122\)](#)

Etapa 1: exclua a distribuição do Amazon CloudFront

Antes de excluir uma distribuição do Amazon CloudFront, você deve 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.

Como desabilitar e excluir uma distribuição do CloudFront

1. Abra o console do CloudFront em <https://console.aws.amazon.com/cloudfront/v3/home>.
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).

Etapa 2: exclua 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).

Como 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).

Etapa 3: Desabilitar o registro em log e excluir o bucket do S3

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. Em Buckets, escolha o nome do bucket e, em seguida, 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 [Excluir um bucket \(p. 139\)](#).

Criar, configurar e trabalhar com buckets do Amazon S3

Para armazenar seus dados no Amazon S3, você trabalha com recursos conhecidos como buckets e objetos. Um bucket é um contêiner de objetos. Um objeto é um arquivo e qualquer metadado que descreva esse arquivo.

Para armazenar um objeto no Amazon S3, crie um bucket e faça upload do objeto em um bucket. Quando o objeto estiver no bucket, você poderá abri-lo, fazer download dele e movê-lo. Quando você não precisar mais de um objeto ou um bucket, poderá limpar seus recursos.

Note

Com o Amazon S3, você paga somente pelo que for usado. Para obter mais informações sobre os recursos e a definição de preço do Amazon S3, consulte [Amazon S3](#). Se você for um novo cliente do Amazon S3, você pode começar a usar o Amazon S3 gratuitamente. Para obter mais informações, consulte [Nível gratuito da AWS](#).

Os tópicos desta seção fornecem uma visão geral do trabalho com buckets no Amazon S3. Eles incluem informações sobre nomeação, criação, acesso e exclusão de buckets. Para obter mais informações sobre como visualizar ou listar objetos em um bucket, consulte [Organizar, listar e trabalhar com seus objetos \(p. 258\)](#).

Tópicos

- [Visão geral dos buckets \(p. 123\)](#)
- [Regras de nomeação de bucket \(p. 127\)](#)
- [Criação de um bucket \(p. 129\)](#)
- [Visualização das propriedades de um bucket do S3 \(p. 134\)](#)
- [Métodos de acesso a um bucket \(p. 135\)](#)
- [Esvaziar um bucket \(p. 137\)](#)
- [Excluir um bucket \(p. 139\)](#)
- [Definir o comportamento padrão da criptografia para os buckets do Amazon S3 \(p. 142\)](#)
- [Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration \(p. 147\)](#)
- [Configuração de buckets de Pagamento pelo solicitante para transferências de armazenamento e uso \(p. 155\)](#)
- [Restrições e limitações do bucket \(p. 159\)](#)

Visão geral dos buckets

Para fazer upload de seus dados (fotos, vídeos, documentos etc.) para o Amazon S3, primeiro é necessário criar um bucket do S3 em uma das Regiões da AWS.

Um bucket é um contêiner para objetos armazenados no Amazon S3. Você pode armazenar qualquer número de objetos em um bucket e pode ter até 100 buckets na sua conta. Para solicitar um aumento, visite o [Service Quotas Console](#) (Console de cotas de serviço).

Cada objeto está contido em um bucket. Por exemplo, se o objeto chamado `photos/puppy.jpg` estiver armazenado no bucket `DOC-EXAMPLE-BUCKET` da região oeste dos EUA (Oregon), ele poderá ser endereçado usando o URL `https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/photos/puppy.jpg`. Para obter mais informações, consulte [Accessing a Bucket \(p. 135\)](#) (Como acessar um bucket).

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

Esta seção descreve como trabalhar com buckets. Para obter mais informações sobre como trabalhar com objetos, consulte [Visão geral de objetos Amazon S3 \(p. 161\)](#).

O Amazon S3 oferece suporte a buckets globais, o que significa que cada nome de bucket deve ser exclusivo em todas as Contas da AWS de todas as Regiões da AWS dentro de uma partição. Uma partição é um agrupamento de regiões. Atualmente, a AWS tem três partições: `aws` (regiões padrão), `aws-cn` (regiões da China) e `aws-us-gov` (AWS GovCloud (US)).

Após a criação de um bucket, o nome dele não poderá ser usado por outra Conta da AWS na mesma partição enquanto ele não for 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 [Regras de nomeação de bucket \(p. 127\)](#).

O Amazon S3 cria buckets na região que você especificar. Para otimizar a latência, minimizar os custos ou atender a requisitos regulatórios, escolha qualquer Região da AWS geograficamente próxima a você. 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 pertencentes 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 que são armazenados na região UE (Irlanda) nunca saem dela.

Tópicos

- [Sobre permissões \(p. 124\)](#)
- [Gerenciar o acesso público aos buckets \(p. 125\)](#)
- [Opções de configuração do bucket \(p. 125\)](#)

Sobre permissões

É possível usar suas credenciais de usuário root da Conta da AWS para criar um bucket e executar qualquer outra operação do Amazon S3. No entanto, recomendamos não usar as credenciais de usuário root da sua Conta da AWS para fazer solicitações, como criar um bucket. Em vez disso, crie um usuário do AWS Identity and Access Management (IAM) e conceda a esse usuário acesso total (por padrão, os usuários não têm nenhuma permissão).

Esses usuários são referidos como administradores. As credenciais do usuário administrador podem ser usadas em vez das credenciais do usuário root 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 usuário root da Conta da AWS e credenciais de usuário do IAM](#) na Referência geral da AWS e [Práticas recomendadas de segurança no IAM](#) no Manual 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 que é a 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 [Identity and Access Management no Amazon S3 \(p. 419\)](#).

Gerenciar o acesso público aos buckets

O acesso público aos buckets e objetos é concedido através de listas de controle de acesso (ACLs), políticas de bucket ou ambos. Para ajudar você a gerenciar o acesso público aos recursos do Amazon S3, o Amazon S3 fornece configurações de bloqueio de acesso público. As configurações de bloqueio de acesso público do Amazon S3 podem substituir ACLs e políticas de bucket para que você possa impor limites uniformes ao acesso público a esses recursos. Você pode aplicar as configurações de bloqueio de acesso público a buckets individualmente ou grupos de buckets na sua conta.

Para ajudar a garantir que todos os seus buckets e objetos do Amazon S3 tenham o acesso público bloqueado, recomendamos ativar as quatro configurações de bloqueio de acesso público na sua conta. Estas configurações bloqueiam o acesso público a todos os buckets atuais e futuros.

Antes de aplicar estas configurações, verifique se seus aplicativos funcionarão corretamente sem acesso público. Se você precisa de um certo nível de acesso público aos seus buckets ou objetos, por exemplo, para hospedar um site estático como descrito em [Hospedagem de um site estático usando o Amazon S3 \(p. 1188\)](#), você pode personalizar as configurações individualmente para atender aos seus casos de uso de armazenamento. Para obter mais informações, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

Note

Se você vir um `Error` ao listar seus buckets e as respectivas configurações de acesso público, talvez você não tenha as permissões necessárias. Confira se você tem as seguintes permissões adicionadas à política de usuário ou perfil:

```
s3:GetAccountPublicAccessBlock
s3:GetBucketPublicAccessBlock
s3:GetBucketPolicyStatus
s3:GetBucketLocation
s3:GetBucketAcl
s3>ListAccessPoints
s3>ListAllMyBuckets
```

Em alguns casos raros, as solicitações também podem falhar devido a uma interrupção da Região da AWS.

Opções de configuração do bucket

O Amazon S3 é compatível com várias opções para que você configure o bucket. Por exemplo, você pode configurar seu bucket para hospedagem de site, adicionar configuração para gerenciar o ciclo de vida de objetos no bucket e configurar o bucket para registrar todos os acessos a ele. O Amazon S3 é compatível com sub-recursos para que você armazene e gerencie informações de configuração do bucket. É possível usar a API do Amazon S3 para criar e gerenciar esses sub-recursos. No entanto, 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
cors (compartilhamento de recurso de origem cruzada)	Você pode configurar seu bucket para autorizar solicitações de origem cruzada. Para obter mais informações, consulte Usar o compartilhamento de recursos de origem cruzada (CORS) (p. 607) .
notificação de evento	Você pode permitir que seu bucket envie notificações de eventos do bucket especificado. Para obter mais informações, consulte Notificações de eventos do Amazon S3 (p. 1083) .
ciclo de vida	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. Para obter mais informações, consulte Gerenciando seu ciclo de vida de armazenamento (p. 746) .
location	Ao criar um bucket, você especifica a Região da AWS onde deseja que o Amazon S3 crie o bucket. O Amazon S3 armazena essas informações no sub-recurso local e fornece uma API para que você recupere essas informações.
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 de acesso podem ser úteis em auditorias de segurança e acesso. Isso também pode ajudá-lo a conhecer sua base de clientes e entender a conta do Amazon S3. Para obter mais informações, consulte Registrar em log as solicitações com registro em log de acesso ao servidor (p. 1041) .
bloqueio de objetos	Para usar o bloqueio de objetos do S3, é necessário habilitá-lo para um bucket. Também é possível configurar um modo e um período de retenção padrão que se aplicam a novos objetos colocados no bucket. Para obter mais informações, consulte Configuração do bucket (p. 726) .
política e ACL (lista de controle de acesso)	Todos os seus recursos (como buckets e objetos) são privados por padrão. O Amazon S3 é compatível com opções de política de bucket e de lista de controle de acesso (ACL) para que você conceda e gerencie permissões no nível do bucket. O Amazon S3 armazena as informações de permissão nos sub-recursos política e acl. Para obter mais informações, consulte Identity and Access Management no Amazon S3 (p. 419) .

Sub-recurso	Descrição
replicação	A replicação é a cópia assíncrona automática de objetos em buckets na mesma Regiões da AWS ou em regiões diferentes. Para obter mais informações, consulte Replicação de objetos (p. 801) .
requestPayment	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 gerenciamento desse sub-recurso. Para obter mais informações, consulte Configuração de buckets de Pagamento pelo solicitante para transferências de armazenamento e uso (p. 155) .
marcação	Você pode adicionar tags de alocação de custo ao seu bucket para classificar e acompanhar seus custos com a 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. Para obter mais informações, consulte Relatórios de uso e faturamento dos buckets do S3 (p. 889) .
aceleração de transferência	O Transfer Acceleration possibilita transferências de arquivos rápidas, fáceis e seguras entre seu cliente e um bucket do S3 em longas distâncias. O Transfer Acceleration tira proveito dos pontos de presença distribuídos globalmente no Amazon CloudFront. Para obter mais informações, consulte Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration (p. 147) .
versionamento	O versionamento ajuda a recuperar substituições e exclusões acidentais. Recomendamos o versionamento como melhor prática para impedir a exclusão ou substituição de objetos por engano. Para obter mais informações, consulte Usando o versionamento em buckets do S3 (p. 678) .
site	Você pode configurar seu bucket para hospedagem de site estático. O Amazon S3 armazena essa configuração criando um sub-recurso site. Para obter mais informações, consulte Hospedagem de um site estático usando o Amazon S3 (p. 1188) .

Regras de nomeação de bucket

As seguintes regras se aplicam aos buckets de nomeação no Amazon S3:

- Os nomes dos buckets devem ter entre 3 (mínimo) e 63 (máximo) caracteres.
- Os nomes dos buckets podem consistir apenas em letras minúsculas, números, pontos (.) e hífens (-).
- Os nomes dos buckets devem começar e terminar com uma letra ou um número.
- O nome do bucket não deve conter pontos adjacentes.
- Os nomes de bucket não devem ser formatados como um endereço IP (por exemplo, 192.168.5.4).
- O nome do bucket não deve iniciar com o prefixo xn--.

- Os nomes de bucket não podem terminar com o sufixo `-s3alias`. Esse sufixo se reserva a nomes de alias de ponto de acesso. Para obter mais informações, consulte [Usar um alias em estilo de bucket para seu ponto de acesso \(p. 333\)](#).
- Os nomes de bucket devem ser exclusivos em todas as Contas da AWS e em todas as Regiões da AWS dentro de uma partição. Uma partição é um agrupamento de regiões. Atualmente, a AWS tem três partições: `aws` (regiões padrão), `aws-cn` (regiões da China) e `aws-us-gov` (AWS GovCloud (US)).
- Um nome de bucket não pode ser usado por outra Conta da AWS na mesma partição enquanto o bucket não for excluído.
- Os buckets usados com o Amazon S3 Transfer Acceleration não podem ter pontos (.) em seus nomes. Para obter mais informações sobre o Transfer Acceleration, consulte [Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration \(p. 147\)](#).

Para obter a melhor compatibilidade, recomendamos evitar o uso de pontos (.) em nomes de buckets, exceto em buckets usados apenas para hospedagem de sites estáticos. Se você incluir pontos no nome de um bucket, não poderá usar o endereçamento em estilo de host virtual por HTTPS, a menos que execute sua própria validação de certificado. Isso ocorre porque os certificados de segurança usados para hospedagem virtual de buckets não funcionam para buckets com pontos nos nomes.

Essa limitação não afeta os buckets usados para hospedagem de sites estáticos, pois essa hospedagem só está disponível via HTTP. Para obter mais informações sobre o endereçamento no estilo de hospedagem virtual, consulte [Hospedagem virtual de buckets \(p. 1249\)](#). Para obter mais informações sobre hospedagem de sites estáticos, consulte [Hospedagem de um site estático usando o Amazon S3 \(p. 1188\)](#).

Note

Antes de 1º de março de 2018, os buckets criados na região Leste dos EUA (Norte da Virgínia) podiam ter nomes com até 255 caracteres e incluíam letras maiúsculas e sublinhados. A partir de 1º de março de 2018, os novos buckets na região Leste dos EUA (Norte da Virgínia) devem estar em conformidade com as mesmas regras aplicadas em todas as outras regiões.

Para obter informações sobre nomes de chave de objeto, consulte [Criar nomes de chave de objeto](#).

Exemplo de nomes de bucket

Os nomes de buckets de exemplo a seguir são válidos e seguem as diretrizes de nomenclatura recomendadas:

- `docexamplebucket1`
- `log-delivery-march-2020`
- `my-hosted-content`

Os nomes de buckets de exemplo a seguir são válidos, mas não recomendados para usos que não sejam hospedagem de sites estáticos:

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

Os nomes de buckets de exemplo a seguir não são válidos:

- `doc_example_bucket` (contém sublinhados)
- `DocExampleBucket` (contém letras maiúsculas)
- `doc-example-bucket-` (termina com um hífen)

Criação de um bucket

Para fazer upload de seus dados para o Amazon S3, você deve primeiro criar um bucket do Amazon S3 em uma das Regiões da AWS. Ao criar um bucket, você deve escolher um nome de bucket e Região. Opcionalmente, você pode escolher outras opções de gerenciamento de armazenamento para o bucket. Assim que você cria um bucket, não pode mais alterar o respectivo nome ou região. Para obter informações sobre nomeação de buckets, consulte [Regras de nomeação de bucket \(p. 127\)](#).

A Conta da AWS que cria o bucket é a proprietária do bucket. Você pode fazer upload de um número ilimitado de objetos para o bucket. Por padrão, você pode criar até 100 buckets em cada Contas da AWS. Se precisar de mais buckets, você poderá aumentar o limite de bucket da conta para um máximo de 1.000 buckets enviando um aumento de limite de serviço. Para saber como solicitar um aumento de limite de bucket, consulte [Cotas de serviço da AWS](#) na Referência geral da AWS. Você pode armazenar qualquer número de objetos no bucket.

O S3 Object Ownership é uma configuração no nível do bucket do Amazon S3 que você pode usar para desabilitar as listas de controle de acesso (ACLs) e assumir a propriedade de cada objeto em seu bucket, simplificando o gerenciamento de acesso para dados armazenados no Amazon S3. Por padrão, quando outra Conta da AWS carrega um objeto no bucket do Amazon S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. Ao criar um bucket, é possível adotar a configuração aplicada pelo proprietário do bucket em Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Consequentemente, o controle de acesso para seus dados é baseado em políticas. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Você pode usar o console do Amazon S3, as APIs do Amazon S3, a AWS CLI ou os AWS SDKs para criar um bucket. Para obter mais informações sobre permissões necessárias para criar um bucket, consulte [CreateBucket](#) na Referência da API do Amazon Simple Storage Service.

Uso do console do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Selecione Create bucket (Criar bucket).

O assistente Create bucket (Criar bucket) é aberto.

3. Em Bucket name (Nome do bucket), insira um nome compatível com o DNS para seu bucket.

O nome do bucket deve:

- Seja exclusivo em todo o Amazon S3.
- Ter entre 3 e 63 caracteres.
- Não contém caracteres maiúsculos.
- Começar com uma letra minúscula ou um número.

Depois de criado o bucket, não é possível mudar o nome. Para obter informações sobre nomeação de buckets, consulte [Regras de nomeação de bucket \(p. 127\)](#).

Important

Evite incluir informações sigilosas, como números de conta, no nome do bucket. O nome do bucket é visível nos URLs que apontam para os objetos no bucket.

4. Em Region (Região), escolha a Região da AWS onde deseja que o bucket resida.

Escolha uma região próxima de você para minimizar a latência e os custos e atender aos requisitos regulatórios. Os objetos armazenados em uma região nunca saem dessa região, a menos que você os transfira para outra região. Para obter uma lista das Regiões da AWS do Amazon S3, consulte [Endpoints de serviço da AWS](#) na Referência geral da Amazon Web Services.

5. Em Object Ownership, para desabilitar ou habilitar ACLs e controlar a propriedade de objetos carregados em seu bucket, escolha uma das seguintes configurações:

ACLs desabilitadas

- Bucket owner enforced (Aplicada pelo proprietário do bucket) (recomendado): as ACLs são desabilitadas e o proprietário do bucket automaticamente tem a propriedade e o controle total sobre todos os objetos do bucket. As ACLs não afetam mais as permissões nos dados no bucket do S3. O bucket usa políticas para definir o controle de acesso.

Para exigir que todos os novos buckets sejam criados com ACLs desabilitadas usando IAM ou políticas do AWS Organizations, consulte [Desabilitar ACLs para todos os novos buckets \(aplicada pelo proprietário do bucket\)](#) (p. 660).

ACLs habilitadas

- Bucket owner preferred (Preferencial do proprietário do bucket): o proprietário do bucket tem a propriedade e o controle total sobre novos objetos que outras contas gravam no bucket com a ACL `bucket-owner-full-control` pré-configurada.

Se você aplicar a configuração preferencial do proprietário do bucket para exigir que todos os carregamentos do Amazon S3 incluam a ACL `bucket-owner-full-control` pré-configurada, será possível [adicionar uma política de bucket](#) (p. 661) que só permita carregamentos de objetos que usam essa ACL.

- Object writer (Gravador de objetos): a Conta da AWS que carrega um objeto é a proprietária do objeto, tem controle total sobre ele e pode conceder acesso a outros usuários por meio de ACLs.

Note

Para aplicar a configuração Bucket owner enforced (Aplicada pelo proprietário do bucket) ou Bucket owner preferred (Preferencial do proprietário do bucket), é necessário ter as seguintes permissões: `s3:CreateBucket` e `s3:PutBucketOwnershipControls`.

6. Em Bucket settings for Block Public Access (Configurações de bucket para o Bloqueio de acesso público), escolha as configurações de bloqueio de acesso público que deseja aplicar ao bucket.

Recomendamos que você mantenha todas as configurações ativadas, a menos que saiba que precisa desativar uma ou mais delas para seu caso de uso, como para hospedar um site público. As configurações de bloqueio de acesso público que você habilitar para o bucket também serão ativadas para todos os pontos de acesso criados no bucket. Para obter mais informações sobre como bloquear o acesso público, consulte [Bloquear o acesso público ao armazenamento do Amazon S3](#) (p. 618).

7. (Opcional) Se você quiser habilitar o Bloqueio de Objetos do S3, faça o seguinte:

- a. Selecione Advanced settings (Configurações avançadas).

Important

Você só pode habilitar o bloqueio de objetos do S3 para um bucket ao criá-lo. Se você habilitar o bloqueio de objeto para o bucket, não poderá desabilitá-lo mais tarde. A ativação do bloqueio de objetos também permite o versionamento para o bucket. Após habilitar o bloqueio de objeto para o bucket, é necessário definir as configurações padrão de retenção e de retenção legal do bloqueio de objeto para proteger novos objetos de

serem excluídos ou substituídos. Para obter mais informações, consulte [Configurar o bloqueio de objetos do S3 usando o console \(p. 727\)](#).

- b. Se quiser ativar o bloqueio de objetos, escolha **Enable**, leia o aviso que aparece e confirme-o.

Para obter mais informações sobre o recurso bloqueio de objeto do S3, consulte [Usar o bloqueio de objetos do S3 \(p. 722\)](#).

Note

Para criar um bucket habilitado para bloqueio de objetos, você deve ter as seguintes permissões: S3:CreateBucket, S3:PutBucketVersioning e S3:PutBucketObjectLockConfiguration.

8. Selecione Create bucket (Criar bucket).

Uso da SDKs AWS

Ao usar os AWS SDKs para criar um bucket, você deve criar um cliente e então usá-lo para enviar uma solicitação para criar um bucket. Como prática recomendada, crie o cliente e o bucket na mesma Região da AWS. Se você não especificar uma região ao criar um cliente ou um bucket, o Amazon S3 usará a região padrão Leste dos EUA (Norte da Virgínia).

Para criar um cliente a fim de acessar um endpoint de pilha dupla, é necessário especificar uma Região da AWS. Para obter mais informações, consulte [Endpoints de pilha dupla \(p. 1216\)](#). Para obter uma lista de Regiões da AWS disponíveis, consulte [Regiões e endpoints](#) na Referência geral da AWS.

Ao criar um cliente, a região mapeia para o endpoint específico da região. O cliente usa esse endpoint para se comunicar com o Amazon S3: s3.<region>.amazonaws.com. Se a sua região foi lançada após 20 de março de 2019, seu cliente e bucket devem estar na mesma região. No entanto, é possível usar um cliente na região Leste dos EUA (Norte da Virgínia) para criar um bucket em qualquer região iniciada antes de 20 de março de 2019. Para obter mais informações, consulte [Endpoints herdados \(p. 1253\)](#).

Os exemplos de código do AWS SDK executam as seguintes tarefas:

- Crie um cliente especificando explicitamente uma região da Região da AWS: no exemplo, o cliente usa o endpoint s3.us-west-2.amazonaws.com para se comunicar com o Amazon S3. É possível especificar qualquer Região da AWS. Para obter uma lista de Regiões da AWS, consulte [Regiões e endpoints](#) na Referência geral da AWS.
- Envie uma solicitação de bucket de criação especificando apenas um nome de bucket — O cliente envia uma solicitação ao Amazon S3 para criar o bucket na região onde você criou um cliente.
- Recupere as informações sobre a localização do bucket. O Amazon S3 armazena as informações de localização do bucket no sub-recurso location que está associado ao bucket.

Java

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 exemplos de código Java no Amazon S3 \(p. 1267\)](#).

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

```
import com.amazonaws.services.s3.model.GetBucketLocationRequest;
import java.io.IOException;

public class CreateBucket2 {

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

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

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

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

.NET

Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

Example

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

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

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

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

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

Ruby

Para obter informações sobre como criar e testar um exemplo funcional, consulte [Usar o AWS SDK for Ruby - versão 3 \(p. 1269\)](#).

Example

```
require "aws-sdk-s3"

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

  # @param bucket [Aws::S3::Bucket] An Amazon S3 bucket initialized with a name. This
  # is a client-side object until
  #                               create is called.
```

```
def initialize(bucket)
  @bucket = bucket
end

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

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

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

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

run_demo if $PROGRAM_NAME == __FILE__
```

Usar a AWS CLI

Você também pode usar a AWS Command Line Interface (AWS CLI) para criar um bucket do S3. Para obter mais informações, consulte [create-bucket](#) na Referência de comandos da AWS CLI.

Para obter informações sobre o AWS CLI, consulte [O que é AWS Command Line Interface?](#) no Guia do usuário do AWS Command Line Interface.

Visualização das propriedades de um bucket do S3

Você pode visualizar e configurar as propriedades de um bucket do Amazon S3, incluindo configurações para versionamento, tags, criptografia padrão, registro em log, notificações e muito mais.

Para visualizar as propriedades de um bucket do S3

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

2. Na lista Buckets, escolha o nome do bucket cujas propriedades deseja visualizar.
3. Escolha Properties (Propriedades).
4. Na página Properties (Propriedades), você pode configurar as seguintes propriedades para o bucket.
 - Bucket Versioning (Versionamento de bucket) — mantenha várias versões de um objeto em um bucket usando o versionamento. Por padrão, o versionamento é desabilitado para um novo bucket. Para obter informações sobre como ativar o versionamento, consulte [Habilitar o versionamento em buckets \(p. 683\)](#).
 - Etiquetas: com a alocação de custos da AWS, você pode usar etiquetas de buckets para anotar o faturamento de seu uso de um bucket. Uma tag é um par chave-valor que representa uma etiqueta que você atribui a um bucket. Para adicionar tags, escolha Tags e, em seguida, Add tag (Adicionar tag). Para obter mais informações, consulte [Usar tags de alocação de custos para buckets do S3 \(p. 888\)](#).
 - Default encryption (Criptografia padrão) — a ativação da criptografia padrão fornece criptografia automática no lado do servidor. O Amazon S3 criptografa um objeto antes de salvá-lo em um disco e descriptografa o objeto ao baixá-lo. Para obter mais informações, consulte [Definir o comportamento padrão da criptografia para os buckets do Amazon S3 \(p. 142\)](#).
 - Server access logging (Registro em log de acesso ao servidor) — obtenha registros detalhados das solicitações feitas ao bucket com registro em log de acesso ao servidor. Por padrão, o Amazon S3 não coleta logs de acesso ao servidor. Para obter informações sobre como habilitar o registro em log de acesso ao servidor, consulte [Habilitar o log de acesso ao servidor do Amazon S3 \(p. 1043\)](#).
 - Eventos de dados do AWS CloudTrail: use o CloudTrail para registrar eventos de dados. Por padrão, as trilhas não registram eventos de dados. Há cobranças adicionais para eventos de dados. Para obter mais informações, consulte [Registro eventos de dados em logs para trilhas](#) no Manual do usuário do AWS CloudTrail.
 - Event notifications (Notificações de eventos) – habilite certos eventos de bucket do Amazon S3 para enviar mensagens de notificação para um destino sempre que ocorrer eventos. Para ativar eventos, escolha Create event notification (Criar notificação de evento) e especifique as configurações que deseja usar. Para obter mais informações, consulte [Habilitar e configurar notificações de eventos usando o console do Amazon S3 \(p. 1092\)](#).
 - Transfer Acceleration — possibilita transferências de arquivos rápidas, fáceis e seguras entre seu cliente e um bucket do S3 em longas distâncias. Para obter informações sobre como habilitar o Transfer Acceleration, consulte [Habilitar e usar o S3 Transfer Acceleration \(p. 150\)](#).
 - Object Lock (Bloqueio de objetos) — use o bloqueio de objetos do S3 para evitar que um objeto seja excluído ou substituído por um período fixo ou indefinidamente. Para obter mais informações, consulte [Usar o bloqueio de objetos do S3 \(p. 722\)](#).
 - Requester Pays (Pagamento pelo solicitante) — habilite o Pagamento pelo solicitante para que o solicitante (em vez do proprietário do bucket) pague por solicitações e transferências de dados. Para obter mais informações, consulte [Configuração de buckets de Pagamento pelo solicitante para transferências de armazenamento e uso \(p. 155\)](#).
 - Static website hosting – você pode hospedar um site estático no Amazon S3. Para habilitar a hospedagem de um site estático, escolha Static website hosting (Hospedagem de sites estáticos) e especifique as configurações desejadas. Para obter mais informações, consulte [Hospedagem de um site estático usando o Amazon S3 \(p. 1188\)](#).

Métodos de acesso a um bucket

É possível 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 acessa um bucket de forma programática, o Amazon S3 oferece suporte à arquitetura RESTful na qual seus buckets e objetos são recursos, cada um com um URI de recurso que os identifica de maneira exclusiva.

O Amazon S3 é compatível com os URLs no estilo de hospedagem virtual e estilo de caminho para acessar um bucket. Como os buckets podem ser acessados usando URLs no estilo de hospedagem virtual e estilo de caminho, recomendamos criar buckets com nomes compatíveis com DNS. Para obter mais informações, consulte [Restrições e limitações do bucket \(p. 159\)](#).

Note

As solicitações no estilo de caminho e estilo de hospedagem virtual usam a estrutura de endpoint S3 ponto Região (`s3.Region`), por exemplo, `https://my-bucket.s3.us-west-2.amazonaws.com`. No entanto, algumas regiões mais antigas do Amazon S3 também são compatíveis com endpoints S3 traço Região (`s3-Region`), por exemplo, `https://my-bucket.s3-us-west-2.amazonaws.com`. Se o bucket estiver em uma dessas regiões, você poderá ver endpoints `s3-Region` nos logs de acesso ao servidor ou nos logs do AWS CloudTrail. Recomendamos que você não use essa estrutura de endpoint em suas solicitações.

Acesso no estilo de hospedagem virtual

Em uma solicitação no estilo de hospedagem virtual, o nome do bucket faz parte do nome do domínio no URL.

Os URLs no estilo de hospedagem virtual do Amazon S3 usam o seguinte formato:

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

Neste exemplo, `DOC-EXAMPLE-BUCKET1` é o nome do bucket, Oeste dos EUA (Oregon) é a região, e `puppy.png` é o nome da chave.

```
https://DOC-EXAMPLE-BUCKET1.s3.us-west-2.amazonaws.com/puppy.png
```

Para obter mais informações sobre acesso no estilo de hospedagem virtual, consulte [Solicitações no estilo de hospedagem virtual \(p. 1250\)](#).

Acesso ao estilo de caminho

No Amazon S3, os URLs no estilo de caminho usam o seguinte formato:

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

Por exemplo, se você criar um bucket chamado `DOC-EXAMPLE-BUCKET1` na região Oeste dos EUA (Oregon) e quiser acessar o objeto `puppy.jpg` nele, use o seguinte URL no estilo de caminho:

```
https://s3.us-west-2.amazonaws.com/DOC-EXAMPLE-BUCKET1/puppy.jpg
```

Para obter mais informações, consulte [Solicitações no estilo de caminho \(p. 1250\)](#).

Important

Atualização (23 de setembro de 2020): para garantir que os clientes tenham o tempo necessário para fazer a transição para URLs no estilo de hospedagem virtual, decidimos postergar a desativação de URLs no estilo de caminho. Para obter mais informações, consulte [Amazon S3 Path Deprecation Plan – The Rest of the Story](#) no Blog de notícias da AWS.

Acessar um bucket do S3 por IPv6

O Amazon S3 tem um conjunto de endpoints de pilha dupla, que são compatíveis com solicitações para buckets do S3 por Internet Protocol versão 6 (IPv6) e por IPv4. Para obter mais informações, consulte [Fazer solicitações por meio do IPv6 \(p. 1213\)](#).

Acessar um bucket por meio de pontos de acesso do S3

Além de acessar um bucket diretamente, você pode fazer isso por meio de um ponto de acesso. Para obter mais informações sobre o recurso pontos de acesso do S3, consulte [Gerenciamento de acesso a dados com pontos de acesso do Amazon S3 \(p. 320\)](#).

Os pontos de acesso do S3 oferecem suporte apenas ao endereçamento em estilo de host virtual. Para endereçar um bucket por meio de um ponto de acesso, use o formato a seguir.

```
https://AccessPointName-AccountId.s3-accesspoint.region.amazonaws.com.
```

Note

- Se o nome do ponto de acesso incluir caracteres de traço (-), inclua os traços no URL e insira outro traço antes do ID da conta. Por exemplo, para usar um ponto de acesso chamado `finance-docs`, de propriedade da conta `123456789012` na Região `us-west-2`, o URL apropriado seria `https://finance-docs-123456789012.s3-accesspoint.us-west-2.amazonaws.com`.
- Os pontos de acesso do S3 não suportam acesso por HTTP, apenas acesso seguro por HTTPS.

Acessar um bucket usando S3://

Alguns serviços da AWS exigem a especificação de um bucket do Amazon S3 usando `s3://bucket`. O exemplo a seguir mostra o formato correto. Lembre-se de que, ao usar esse formato, o nome do bucket não inclui a Região da AWS.

```
s3://bucket-name/key-name
```

O exemplo a seguir usa o bucket de amostra descrito na seção de estilo de caminho anterior.

```
s3://mybucket/puppy.jpg
```

Esvaziar um bucket

Você pode esvaziar o conteúdo de um bucket usando o console do Amazon S3, AWS SDKs ou a AWS Command Line Interface (AWS CLI). Quando você esvazia um bucket, você exclui todos os objetos, mas mantém o bucket. Não é possível desfazer a ação de esvaziar um bucket. Ao esvaziar um bucket com o S3 Bucket Versioning habilitado ou suspenso, todas as versões de todos os objetos no bucket são excluídas. Para obter mais informações, consulte [Trabalhar com objetos em um bucket com versionamento habilitado \(p. 689\)](#).

Você também pode especificar a configuração de ciclo de vida em um bucket para expirar objetos para que o Amazon S3 possa excluí-los. Para obter mais informações, consulte [Definir a configuração do ciclo de vida em um bucket \(p. 753\)](#)

Troubleshooting

Os objetos adicionados ao bucket enquanto a ação de esvaziar estiver em andamento também poderão ser excluídos. Para evitar que novos objetos sejam adicionados a um bucket enquanto a ação de esvaziar

estiver em andamento, talvez seja necessário impedir que suas trilhas do AWS CloudTrail registrem eventos no bucket. Para obter mais informações, consulte [Desativação do registro em log de uma trilha](#) no Manual do usuário do AWS CloudTrail.

Outra alternativa para impedir que as trilhas do CloudTrail sejam adicionadas ao bucket é adicionar uma instrução de negação s3:PutObject à sua política de bucket. Se você quiser armazenar novos objetos no bucket, remova a instrução de negação s3:PutObject da sua política de bucket. Para obter mais informações, consulte [Exemplo: operações de objeto \(p. 442\)](#) e [Elementos da política de JSON do IAM: efeito](#) no Manual do usuário do IAM.

Uso do console do S3

Você pode usar o console do Amazon S3 para esvaziar um bucket, que exclui todos os objetos no bucket sem excluir o bucket.

Para esvaziar um bucket do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Bucket name (Nome do bucket), selecione a opção ao lado do nome do bucket que você deseja esvaziar e selecione Empty (Esvaziar).
3. Na página Empty bucket (Esvaziar bucket), confirme se deseja esvaziar o bucket inserindo o nome do bucket no campo de texto e escolha Empty (Esvaziar).
4. Monitore o andamento do processo de esvaziamento do bucket na página Esvaziar bucket: status.

Usar a AWS CLI

Você só pode esvaziar um bucket usando a AWS CLI se o bucket não tiver o versionamento habilitado. Se o versionamento não estiver habilitado, você poderá usar o comando `rm` (remove) da AWS CLI 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 [Uso de comandos de alto nível do S3 com a AWS CLI](#) no Manual do usuário da AWS Command Line Interface.

Note

Não é possível remover objetos de um bucket que tenha o versionamento ativado. O Amazon S3 adiciona um marcador de exclusão quando você exclui um objeto, que é o que este comando faz. Para obter mais informações sobre o S3 Bucket Versioning, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

Uso da SDKs AWS

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 \(p. 139\)](#). O código exclui todos os objetos, independentemente de o bucket ter versionamento habilitado 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 a Amazon Web Services](#).

Usando uma configuração de ciclo de vida

Se usar uma política de ciclo de vida para esvaziar seu bucket, a política de ciclo de vida deve incluir [versões atuais](#), [versões não atuais](#), [marcadores de exclusão](#) e [uploads fracionados incompletos](#).

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.

O Amazon S3 é compatível com uma regra de ciclo de vida de bucket que você pode usar para interromper uploads fracionados que não são concluídos dentro de um número especificado de dias após a inicialização. Recomendamos que você configure essa regra de ciclo de vida para minimizar os custos de armazenamento. Para obter mais informações, consulte [Configurando uma política de ciclo de vida de bucket para anular multipart uploads incompletos \(p. 185\)](#).

Para obter mais informações sobre como usar uma configuração de ciclo de vida para esvaziar um bucket, consulte [Definir a configuração do ciclo de vida em um bucket \(p. 753\)](#) e [Expirando objetos \(p. 753\)](#).

Excluir um bucket

Você pode excluir um bucket vazio do Amazon S3. Antes de excluir um bucket, considere o seguinte:

- Nomes de bucket são exclusivos. Se você excluir um bucket, outro usuário da AWS poderá usar o nome.
- Se o bucket hospedar um site estático e você tiver criado e configurado uma zona hospedada do Amazon Route 53 conforme descrito em [Configurar um site estático usando um domínio personalizado registrado no Route 53 \(p. 105\)](#), você deverá limpar as configurações da zona hospedada do Route 53 relacionadas ao bucket. Para obter mais informações, consulte [Etapa 2: exclua a zona hospedada do Route 53 \(p. 121\)](#).
- Se o bucket receber dados de log do Elastic Load Balancing (ELB): recomendamos a interrupção da entrega de logs de ELB para o bucket antes da exclusão. Após excluir o bucket, se outro usuário criar um bucket usando o mesmo nome, os dados de log poderão ser entregues a esse bucket. Para obter informações sobre logs de acesso do ELB, consulte [Logs de acesso](#) no Guia do usuário para Classic Load Balancers e [Logs de Acesso](#) no Guia do usuário para Application Load Balancers.

Solução de problemas

Se você não conseguir excluir um bucket do Amazon S3, considere o seguinte:

- Verifique se o bucket está vazio: exclua apenas buckets que não contenham nenhum objeto. Verifique se o bucket está vazio.
- permissões s3:DeleteBucket: se você não puder excluir um bucket, trabalhe com o administrador do IAM para confirmar que você tem permissões s3:DeleteBucket na política de usuário do IAM. Para obter informações sobre como visualizar ou atualizar permissões do IAM, consulte [Alteração de permissões de um usuário do IAM](#) no Guia do usuário do IAM.
- instrução de negação s3:DeleteBucket: se você tiver as permissões s3:DeleteBucket em sua política do IAM e não puder excluir um bucket, a política de bucket poderá incluir uma instrução de negação

para `s3:DeleteBucket`. Os buckets criados pelo ElasticBeanstalk têm uma política que contém essa instrução por padrão. Antes de excluir o bucket, você deve excluir essa instrução ou a política de bucket.

Important

Nomes de bucket são exclusivos. Se você excluir um bucket, outro usuário da AWS poderá usar o nome. Se você deseja continuar a usar o mesmo nome do bucket, não exclua o bucket. Recomendamos que você esvazie o bucket e mantenha-o.

Uso do console do S3

Para excluir um bucket do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, selecione a opção ao lado do nome do bucket que você deseja excluir e escolha Delete (Excluir) na parte superior da página.
3. Na página Delete bucket (Excluir bucket), confirme se deseja excluir o bucket inserindo o nome do bucket no campo de texto e escolha Delete bucket (Excluir bucket).

Note

Se o bucket contiver objetos, esvazie o bucket antes de excluí-lo selecionando o link empty bucket configuration (configuração de esvaziamento de bucket) no alerta de erro This bucket is not empty (Este bucket não está vazio) e seguindo as instruções na página Empty bucket (Esvaziar bucket). Depois, volte para a página Delete bucket (Excluir bucket) e exclua o bucket.

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

Usar o AWS SDK for Java

O exemplo a seguir mostra como excluir um bucket usando o AWS SDK for Java. 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 a Amazon Web Services](#).

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 exemplos de código Java no Amazon S3 \(p. 1267\)](#).

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

```
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.Iterator;

public class DeleteBucket2 {

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

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

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

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

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

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

            // After all objects and object versions are deleted, delete the bucket.
            s3Client.deleteBucket(bucketName);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
        }
    }
}
```

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

Usar a AWS CLI

Você pode excluir um bucket que contém objetos com a AWS CLI se ele não tiver o versionamento habilitado. Ao excluir um bucket que contém objetos, todos os objetos no bucket são excluídos permanentemente, incluindo objetos que passaram para a classe de armazenamento S3 Glacier.

Se o bucket não tiver o versionamento habilitado, você poderá usar o comando `rb` (remover bucket) da AWS CLI com o parâmetro `--force` para excluir o bucket e todos os objetos nele. Esse comando exclui todos os objetos primeiro e, em seguida, exclui o bucket.

Se o versionamento estiver habilitado, os objetos versionados não serão excluídos nesse processo, o que causaria falha na exclusão do bucket porque ele não estaria vazio. Para obter mais informações sobre como excluir objetos com versionamento, consulte [Excluir versões de objetos](#).

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

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

Definir o comportamento padrão da criptografia para os buckets do Amazon S3

Com a criptografia padrão do Amazon S3, você pode definir o comportamento de criptografia padrão para um bucket do S3 para que todos os novos objetos sejam criptografados quando estiverem armazenados no bucket. Os objetos são criptografados usando a criptografia no lado do servidor com as chaves gerenciadas pelo Amazon S3 (SSE-S3) ou AWS KMS keys armazenadas no AWS Key Management Service (AWS KMS) (SSE-KMS).

Ao configurar seu bucket para usar criptografia padrão com o SSE-KMS, você também pode habilitar o recurso de chaves de bucket do S3 para diminuir o tráfego de solicitações do Amazon S3 para o AWS Key Management Service (AWS KMS) e reduzir o custo de criptografia. Para obter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3 \(p. 369\)](#).

Quando você usa a criptografia no lado do servidor, o Amazon S3 criptografa um objeto antes de salvá-lo no disco 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 no lado do servidor \(p. 359\)](#).

Para obter mais informações sobre permissões necessárias para criptografia padrão, consulte [PutBucketEncryption](#) na Referência da API do Amazon Simple Storage Service.

Para configurar a criptografia padrão em um bucket, use o console do Amazon S3, a AWS CLI, os AWS SDKs ou a API REST. Para obter mais informações, consulte [the section called “Habilitar criptografia padrão” \(p. 144\)](#).

Criptografar objetos existentes

Para criptografar objetos existentes do Amazon S3 com uma única solicitação, você pode usar o Amazon S3 Batch Operations. Você fornece uma lista de objetos às operações em lote do S3 que, por sua vez, chamam a respectiva API para realizar a operação especificada. É possível usar a [operação Copy do Batch Operations](#) para copiar objetos não criptografados existentes e gravá-los de volta no mesmo bucket que os objetos criptografados. Um único trabalho do Batch Operations pode realizar a operação especificada em bilhões de objetos. Para obter mais informações, consulte [Executar operações em lote de grande escala em objetos do Amazon S3 \(p. 938\)](#) e a publicação do Blog de armazenamento da AWS [Criptografia de objetos existentes com o Amazon S3 Batch Operations](#).

Você também pode criptografar objetos existentes usando a API Copy Object. Para obter mais informações, consulte a publicação do Blog de armazenamento da AWS [Criptografia de objetos existentes do Amazon S3 com a AWS CLI](#).

Note

Os buckets do Amazon S3 com criptografia de bucket padrão que usam SSE-KMS não podem ser usados como buckets de destino para [the section called “Registrando acesso ao servidor” \(p. 1041\)](#). Somente a criptografia padrão SSE-S3 é suportada para buckets de destino do log de acesso do servidor.

Usar criptografia para operações entre contas

Esteja ciente do seguinte ao usar criptografia para operações entre contas:

- A chave gerenciada da AWS (aws/s3) é usada quando um alias ou nome do recurso da Amazon (ARN) AWS KMS key não é fornecido no momento da solicitação nem por meio da configuração de criptografia padrão do bucket.
- Se você estiver carregando ou acessando objetos do S3 usando as entidades do AWS Identity and Access Management (IAM) que estão na mesma Conta da AWS da sua chave do KMS, você poderá usar a chave gerenciada pela AWS (aws/s3).
- Use uma chave gerenciada pelo cliente se desejar conceder acesso entre contas aos objetos do S3. Você pode configurar a política de uma chave gerenciada pelo cliente para permitir o acesso de outra conta.
- Se especificar sua própria chave do KMS, você deve usar um ARN de chave do KMS totalmente qualificado. Ao usar um alias da chave do KMS, esteja ciente de que o AWS KMS resolverá na chave na conta do solicitante. Isso pode resultar em dados criptografados com uma chave do KMS que pertence ao solicitante, e não ao administrador do bucket.
- É necessário especificar uma chave para a qual você (o solicitante) recebeu a permissão `Encrypt`. Para obter mais informações, consulte [Permitir que os usuários da chave usem uma chave do KMS para operações criptográficas](#) no Guia do desenvolvedor do AWS Key Management Service.

Para obter mais informações sobre quando usar as chaves gerenciadas pelo cliente e as chaves do KMS gerenciadas pela AWS, consulte [Devo usar uma chave gerenciada pela AWS ou uma chave do KMS gerenciada pelo cliente para criptografar meus objetos no Amazon S3?](#)

Como usar a criptografia padrão com a replicação

Ao habilitar a criptografia padrão para um bucket de destino de replicação, 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.

Para obter mais informações sobre como usar a criptografia padrão com SSE-KMS, consulte [Replicar objetos criptografados \(p. 867\)](#).

Usar Chaves de bucket do Amazon S3 com criptografia padrão

Quando você configura seu bucket para usar criptografia padrão para SSE-KMS em novos objetos, você também pode configurar o recurso Chave de bucket do S3. As chaves de bucket do S3 diminuem o número de transações do Amazon S3 para o AWS KMS a fim de reduzir o custo da criptografia do lado do servidor usando AWS Key Management Service (SSE-KMS).

Quando você configura seu bucket para usar chaves de bucket do S3 para SSE-KMS em novos objetos, o AWS KMS gera uma chave no nível de bucket usada para criar uma [chave de dados](#) exclusiva para objetos no bucket. Essa chave de bucket é usada por um período limitado no Amazon S3, reduzindo a necessidade do Amazon S3 fazer solicitações ao AWS KMS para concluir operações de criptografia.

Para obter mais informações sobre como usar uma Chave de bucket do S3, consulte [Uso de chaves de bucket do Amazon S3 \(p. 369\)](#).

Habilitar a criptografia de bucket padrão do Amazon S3

Você pode definir o comportamento de criptografia padrão em um bucket do Amazon S3 para que todos os objetos sejam criptografados quando estiverem armazenados no bucket. Os objetos são criptografados usando a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) ou chaves do AWS Key Management Service (AWS KMS).

Ao configurar a criptografia padrão usando o AWS KMS, você também pode configurar a chave de bucket do S3. Para obter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3 \(p. 369\)](#).

A criptografia padrão funciona com todos os buckets do Amazon 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 Amazon S3 para rejeitar solicitações de armazenamento que não incluem informações de criptografia.

Não há cobranças adicionais 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 preços, consulte [Preços do Amazon S3](#). Para o armazenamento de chave SSE-KMS do KMS, as tarifas do AWS KMS são aplicáveis e estão listadas em [Preços do AWS KMS](#).

Alterações feitas às observações antes de habilitar a criptografia padrão

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 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](#).

Uso do console do S3

Para habilitar a criptografia padrão em um bucket do Amazon S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista de Buckets, escolha o nome do bucket desejado.
3. Escolha Properties (Propriedades).
4. Em Default encryption (Criptografia padrão), escolha Edit (Editar).
5. Para ativar ou desativar a criptografia do lado do servidor, escolha Enable (Ativar) ou Disable (Desabilitar).
6. Para habilitar a criptografia no lado do servidor usando uma chave gerenciada pelo Amazon S3, em Encryption key type (Tipo de chave de criptografia), escolha Amazon S3 key (SSE-S3) (Chave do Amazon S3 (SSE-S3)).

Para obter mais informações sobre como usar a criptografia no lado do servidor do Amazon S3 para criptografar seus dados, consulte [Proteção de dados usando criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 \(SSE-S3\)](#) (p. 378).

7. Para habilitar a criptografia no lado do servidor usando uma AWS KMS key, siga estas etapas:
 - a. Em Encryption key type (Tipo de chave de criptografia), escolha AWS Key Management Service chave (SSE-KMS).

Important

Se você usar a opção do AWS KMS na sua configuração de criptografia padrão, estará sujeito aos limites de RPS (solicitações por segundo) do AWS KMS. Para obter mais informações sobre as cotas do AWS KMS e como solicitar um aumento de cota, consulte [Cotas](#).

- b. Em AWS KMS key (Chave do AWS KMS), escolha uma das seguintes opções:
 - Chave gerenciada da AWS (aws/s3)
 - Choose from your KMS root keys (Escolher de suas chaves-raiz do KMS) e selecione sua chave-raiz do KMS.
 - Enter KMS root key ARN (Inserir o ARN da chave-raiz do KMS) e insira o ARN de sua chave do AWS KMS.

Important

Você só pode usar chaves do KMS habilitadas na mesma Região da AWS que o bucket. Quando você seleciona Choose from your KMS keys , (Escolher de suas chaves do KMS), o console do S3 lista somente 100 chaves do KMS por região. Se você tiver mais de 100 chaves do KMS na mesma região, será possível ver somente as primeiras 100 chaves do KMS no console do S3. Para usar uma chave do KMS que não esteja listada

no console, escolha Custom KMS ARN (Personalizar o ARN do KMS) e insira o ARN da chave do KMS.

Ao usar uma AWS KMS key para criptografia no lado do servidor no Amazon S3, você deve escolher uma chave de criptografia do KMS simétrica. O Amazon S3 só é compatível com chaves do KMS de criptografia simétrica, e não com chaves assimétricas do KMS. Para obter mais informações, consulte [Como usar chaves simétricas e assimétricas](#) no AWS Key Management Service Guia do desenvolvedor.

Para obter mais informações sobre como criar uma AWS KMS key, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service. Para obter mais informações sobre como usar o AWS KMS com o Amazon S3, consulte [Proteção de dados usando criptografia do servidor com AWS Key Management Service \(SSE-KMS\)](#) (p. 360).

8. Para usar Chaves de bucket do S3, em Bucket Key (Chave de bucket), escolha Enable (Habilitar).

Quando você configura seu bucket para usar a criptografia padrão com o SSE-KMS, você também pode habilitar a chave de bucket do S3. As chaves de bucket do S3 diminuem o tráfego de solicitações do Amazon S3 para o AWS KMS e reduzem o custo da criptografia. Para obter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3](#) (p. 369).

9. Selecione Save changes.

Usar a AWS CLI

Esses exemplos mostram como configurar a criptografia padrão usando criptografia gerenciada pelo Amazon S3 (SSE-S3) ou criptografia do AWS KMS (SSE-KMS) com uma chave de bucket do S3.

Para obter mais informações sobre criptografia padrão, consulte [Definir o comportamento padrão da criptografia para os buckets do Amazon S3](#) (p. 142). Para obter mais informações sobre o uso da AWS CLI para configurar a criptografia padrão, consulte [put-bucket-encryption](#).

Example – Criptografia padrão com SSE-S3

Esse exemplo configura a criptografia de bucket padrão com criptografia gerenciada pelo Amazon S3.

```
aws s3api put-bucket-encryption --bucket bucket-name --server-side-encryption-configuration
{
    "Rules": [
        {
            "ApplyServerSideEncryptionByDefault": {
                "SSEAlgorithm": "AES256"
            }
        }
    ]
}
```

Example – Criptografia padrão com SSE-KMS usando uma chave de bucket do S3

Esse exemplo configura a criptografia de bucket padrão com o SSE-KMS usando uma chave de bucket do S3.

```
aws s3api put-bucket-encryption --bucket bucket-name --server-side-encryption-configuration
{
    "Rules": [
        {
            "ApplyServerSideEncryptionByDefault": {
                "SSEAlgorithm": "aws:kms",
                "KMSMasterKeyID": "KMS-Key-ARN"
            },
            "BucketKeyEnabled": true
        }
    ]
}
```

]']

Uso dos REST API

Use a operação de criptografia PUT Bucket da API REST para habilitar a criptografia padrão e definir o tipo de criptografia do lado do servidor de modo a usar SSE-S3 ou SSE-KMS.

Para obter mais informações, consulte [PutBucketEncryption](#) na Referência da API do Amazon Simple Storage Service.

Monitorar a criptografia padrão com CloudTrail e CloudWatch

Você pode rastrear solicitações de configuração de criptografia padrão para buckets do Amazon S3 usando eventos do AWS CloudTrail. Os seguintes nomes de eventos de API são usados nos logs do CloudTrail:

- [PutBucketEncryption](#)
- [GetBucketEncryption](#)
- [DeleteBucketEncryption](#)

Você também pode criar Amazon CloudWatch Events com operações no nível de bucket do S3 como o tipo de evento. Para obter mais informações sobre eventos do CloudTrail, consulte [Habilitar o registro em log de objetos em um bucket usando o console \(p. 1033\)](#).

Você pode usar logs do CloudTrail para ações do Amazon S3 no nível de objeto a fim de rastrear solicitações PUT e POST para o Amazon S3. Você pode usar essas ações para verificar se a criptografia padrão está sendo usada para criptografar objetos quando as solicitações PUT recebidas não têm 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 um dos seguintes campos 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 como usar o CloudTrail e o CloudWatch, consulte [Monitorar o Amazon S3 \(p. 1020\)](#).

Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration

O Amazon S3 Transfer Acceleration é um recurso em nível de bucket que possibilita transferências de arquivos rápidas, fáceis e seguras em longas distâncias entre o seu cliente e um bucket do S3. O Transfer Acceleration foi projetado para otimizar as velocidades de transferência de todo o mundo para buckets do S3. O Transfer Acceleration tira proveito dos pontos de presença distribuídos globalmente no Amazon

CloudFront. Conforme os dados chegam em um ponto de presença, eles são roteados para o Amazon S3 por um caminho de rede otimizado.

Quando você usa o Transfer Acceleration, podem ser aplicadas cobranças adicionais de transferência de dados. Para obter mais informações sobre a definição de preços, consulte [Definição de preços do Amazon S3](#).

Por que usar o Transfer Acceleration?

Você pode usar o Transfer Acceleration em um bucket por vários motivos:

- Seus clientes 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](#).

Requisitos para usar o Transfer Acceleration

Veja o que é necessário ao usar o Transfer Acceleration em um bucket do S3:

- O Transfer Acceleration só tem suporte em solicitações de estilo hospedadas virtualmente. Para obter mais informações sobre solicitações de estilo hospedadas virtualmente, consulte [Fazer solicitações usando a API REST \(p. 1247\)](#).
- O nome do bucket usado para o Transfer Acceleration deve ser compatível com DNS e não deve conter pontos (“.”).
- O Transfer Acceleration deve estar ativado no bucket. Para obter mais informações, consulte [Habilitar e usar o S3 Transfer Acceleration \(p. 150\)](#).

Depois de habilitar o Transfer Acceleration em um bucket, pode levar até 20 minutos para que a velocidade da transferência de dados para o bucket aumente.

Note

No momento, o Transfer Acceleration não é compatível com buckets localizados nas seguintes regiões:

- África (Cidade do Cabo) (af-south-1)
 - Ásia-Pacífico (Hong Kong) (ap-east-1)
 - Ásia-Pacífico (Osaka) (ap-northeast-3)
 - UE (Estocolmo) (eu-north-1)
 - UE (Milão) (eu-south-1)
 - Oriente Médio (Bahrein) (me-south-1)
- Para acessar o bucket que está habilitado para o Transfer Acceleration, você deve usar o endpoint `bucketname.s3-accelerate.amazonaws.com`. Ou use o endpoint de pilha dupla `bucketname.s3-accelerate.dualstack.amazonaws.com` para se conectar ao bucket habilitado por 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 habilitarem ou desabilitarem 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 [Exemplo: operações de sub-recursos de bucket \(p. 443\)](#) e [Identity and Access Management no Amazon S3 \(p. 419\)](#).

As seções a seguir descrevem como começar a usar o Amazon S3 Transfer Acceleration para transferir dados.

Tópicos

- [Conceitos básicos do Amazon S3 Transfer Acceleration \(p. 149\)](#)
- [Habilitar e usar o S3 Transfer Acceleration \(p. 150\)](#)
- [Usar a ferramenta de comparação de velocidade do Amazon S3 Transfer Acceleration \(p. 155\)](#)

Conceitos básicos do Amazon S3 Transfer Acceleration

Você pode usar o Amazon S3 Transfer Acceleration para transferências de arquivos rápidas, fáceis e seguras em longas distâncias entre o seu cliente e um bucket do S3. O Transfer Acceleration usa os pontos de presença distribuídos globalmente no Amazon CloudFront. Conforme os dados chegam em um ponto de presença, eles são roteados para o Amazon S3 por um caminho de rede otimizado.

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

1. Habilitar o Transfer Acceleration em um bucket

Você pode ativar o Transfer Acceleration em um bucket de qualquer uma das seguintes maneiras:

- Use o console do Amazon S3.
- Use a operação [PUT Bucket accelerate](#) da API REST.
- Use a AWS CLI e os AWS SDKs. Para obter mais informações, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers \(p. 1258\)](#).

Para obter mais informações, consulte [Habilitar e usar o S3 Transfer Acceleration \(p. 150\)](#).

Note

Para que seu bucket funcione com a aceleração de transferência, o nome dele deve estar em conformidade com os requisitos de nomenclatura de DNS e não deve conter pontos (“.”).

2. Transferir dados de e para o bucket habilitado para aceleração

Use um dos seguintes nomes de domínio de endpoint do s3-accelerate:

- Para acessar um bucket habilitado para aceleração, use `bucketname.s3-accelerate.amazonaws.com`.
- Para acessar um bucket habilitado para aceleração por IPv6, use `bucketname.s3-accelerate.dualstack.amazonaws.com`.

Os endpoints de pilha dupla do Amazon S3 oferecem suporte para buckets do S3 por 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 \(p. 1213\)](#) e [Usar endpoints de pilha dupla do Amazon S3 \(p. 1216\)](#).

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,

suponha que você tenha atualmente uma aplicação API REST usando [PUT Object](#) que usa o nome do host `mybucket.s3.us-east-1.amazonaws.com` na solicitação `PUT`. Para acelerar o `PUT`, altere o nome do host em sua solicitação para `mybucket.s3-accelerate.amazonaws.com`. Para voltar a usar a velocidade de upload padrão, altere o nome de volta para `mybucket.s3.us-east-1.amazonaws.com`.

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

Você pode usar o endpoint de aceleração na AWS CLI, em AWS SDKs e em outras ferramentas que transferem dados para e do Amazon S3. Se você estiver usando AWS SDKs, 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 [Habilitar e usar o S3 Transfer Acceleration \(p. 150\)](#).

Você pode usar todas as operações do Amazon S3 por meio dos endpoints de aceleração de transferência, exceto as seguintes:

- [GET Service \(listar buckets\)](#)
- [PUT Bucket \(criar bucket\)](#)
- [DELETE bucket](#)

Além disso, o Amazon S3 Transfer Acceleration não oferece suporte a cópias entre regiões usando [PUT Object - Copy](#).

Habilitar e usar o S3 Transfer Acceleration

Você pode usar arquivos de transferência do Amazon S3 Transfer Acceleration de forma rápida e segura em longas distâncias entre seu cliente e um bucket do S3. Você pode habilitar o Transfer Acceleration usando o console do S3, a AWS Command Line Interface (AWS CLI) ou os AWS SDKs.

Esta seção fornece exemplos de como ativar o Amazon S3 Transfer Acceleration em um bucket e usar o endpoint de aceleração para o bucket ativado.

Para obter mais informações sobre os requisitos do Transfer Acceleration, consulte [Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration \(p. 147\)](#).

Uso do console do S3

Note

Se você quiser comparar velocidades de upload aceleradas e não aceleradas, abra a [ferramenta Comparação de velocidade do Amazon S3 Transfer Acceleration](#).

A ferramenta de Comparação de velocidade usa o carregamento fracionado para transferir um arquivo do seu navegador para várias Regiões da AWS com e sem o uso do Amazon S3 Transfer Acceleration. É possível comparar a velocidade de upload para uploads diretos e uploads de transferência acelerada por região.

Para habilitar o Transfer Acceleration para um bucket do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket no qual você deseja habilitar a aceleração de transferência.
3. Escolha Properties (Propriedades).

4. Em Transfer acceleration (Aceleração de transferência), escolha Edit (Editar).
5. Escolha Enable (Habilitar) e Save changes (Salvar alterações).

Para acessar transferências de dados aceleradas

1. Depois que o Amazon S3 habilitar a aceleração de transferência para seu bucket, visualize a guia Properties (Propriedades) do bucket.
2. Em Transfer acceleration (Aceleração de transferência), o Accelerated endpoint (Endpoint acelerado) exibe o endpoint de aceleração de transferência para o bucket. Use esse endpoint para acessar transferências de dados aceleradas do bucket e para ele.

Se você suspender a Transfer Acceleration, o endpoint de aceleração não funcionará mais.

Usar a AWS CLI

Veja a seguir exemplos de comandos da AWS CLI usados para o Transfer Acceleration. Para obter instruções de configuração da AWS CLI, consulte [Desenvolvimento com o Amazon S3 usando a AWS CLI \(p. 1257\)](#).

Habilitar o Transfer Acceleration em um bucket

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 ativar 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
```

Habilitar o Transfer Acceleration

Você pode direcionar todas as solicitações do Amazon S3 feitas pelos comandos s3 e s3api da AWS CLI ao endpoint de aceleração: s3-accelerate.amazonaws.com. Para fazer isso, defina o valor de configuração use_accelerate_endpoint como true em um perfil no arquivo do AWS Config. O Transfer Acceleration deve ser ativado 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. Nenhuma solicitação ListBuckets, CreateBucket e DeleteBucket será enviada ao endpoint de aceleração porque esse endpoint não é compatível com essas operações.

Para obter mais informações sobre use_accelerate_endpoint, consulte [Configuração do S3 com a AWS CLI](#) na Referência de comandos da AWS CLI.

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:

- Use o endpoint de aceleração para qualquer comando s3 ou s3api definindo o parâmetro --endpoint-url como https://s3-accelerate.amazonaws.com.

- Configure 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`. Ao executar um comando, especifique qual perfil deseja usar, caso queira ou não usar o endpoint de aceleração.

Fazer 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 https://s3-accelerate.amazonaws.com
```

Uso da SDKs AWS

Veja a seguir exemplos de uso do Transfer Acceleration para fazer upload de objetos para o Amazon S3 usando o AWS SDK. 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`.

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 o Transfer Acceleration habilitado.
- Habilita o Transfer Acceleration em um bucket especificado. Essa etapa é necessária somente se o bucket que você especificar não tiver o Transfer Acceleration habilitado 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 Transfer Acceleration, consulte [Conceitos básicos do Amazon S3 Transfer Acceleration \(p. 149\)](#). Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

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

```
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketAccelerateConfiguration;
import com.amazonaws.services.s3.model.BucketAccelerateStatus;
import com.amazonaws.services.s3.model.GetBucketAccelerateConfigurationRequest;
import com.amazonaws.services.s3.model.SetBucketAccelerateConfigurationRequest;

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

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

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

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

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

.NET

O exemplo a seguir mostra como usar o AWS SDK for .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 do Amazon S3 .NET \(p. 1268\)](#).

Example

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

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

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

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

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

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

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

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.

Python (Boto)

Para obter um exemplo de ativação do Transfer Acceleration usando o SDK para Python, consulte [put_bucket_accelerate_configuration](#) na Referência da API do AWS SDK for Python (Boto3).

Other

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

Usar a 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 em regiões do Amazon S3. A ferramenta de comparação de velocidade usa o 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 qualque um dos seguintes métodos:

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

`https://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 [Endpoints e cotas do Amazon S3](#) na Referência geral da AWS.

- Use o console do Amazon S3.

Configuração de buckets de Pagamento pelo solicitante para transferências de armazenamento e uso

Geralmente, proprietários de bucket pagam por todos os custos de armazenamento e transferência de dados do Amazon S3 associados ao bucket. No entanto, você pode configurar um bucket para ser 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 da solicitação permite que o Amazon S3 identifique e cobre o solicitante pelo uso do bucket de Pagamento pelo solicitante.

Quando o solicitante assume 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 Manual do usuário do IAM.

Depois de configurar um bucket para ser um bucket de pagamento pelo solicitante, os solicitantes devem incluir `x-amz-request-payer` no cabeçalho da solicitação de API, para solicitações DELETE, GET, HEAD, POST e PUT, 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
- Solicitações de SOAP
- Usando um bucket Pagamento pelo solicitante como bucket de destino para log de usuário final ou vice-versa. No entanto, você pode ativar o log do usuário final em um bucket Pagamento pelo solicitante no qual o bucket de destino não é um bucket Pagamento pelo solicitante.

Como funcionam as cobranças de Pagamento pelo solicitante

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. Contudo, o proprietário do bucket é cobrado pela solicitação nas seguintes condições:

- O solicitante não inclui o parâmetro `x-amz-request-payer` no cabeçalho (DELETE, GET, HEAD, POST e PUT) nem 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.

Para obter mais informações sobre pagamentos pelo solicitante, consulte os tópicos abaixo.

Tópicos

- [Configurar Pagamento pelo solicitante em um bucket \(p. 156\)](#)
- [Recuperar a configuração requestPayment usando a API REST \(p. 158\)](#)
- [Fazer download de objetos em buckets de Pagamento pelo solicitante \(p. 158\)](#)

Configurar Pagamento pelo solicitante em um bucket

Você pode configurar um bucket do Amazon S3 para ser um bucket de Pagamento pelo solicitante, de modo que o solicitante pague o custo da solicitação e do download de dados em vez do proprietário do bucket.

Esta seção fornece exemplos de como configurar o pagamento pelo solicitante em um bucket do Amazon S3 usando o console e a API REST.

Uso do console do S3

Como habilitar o Pagamento pelo solicitante para um bucket do S3

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

2. Na lista Buckets escolha o nome do bucket para o qual você deseja habilitar Pagamento pelo solicitante.
3. Escolha Properties (Propriedades).
4. Em Requester pays (Pagamento pelo solicitante), escolha Edit (Editar).
5. Escolha Enable (Habilitar) e Save changes (Salvar alterações).

O Amazon S3 habilitará o Pagamento pelo solicitante para o bucket e exibirá a Bucket overview (Visão geral do bucket). Em Requester pays (Pagamento pelo solicitante), você verá a opção Enabled (Habilitado).

Uso dos REST API

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 fazer upload de 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 houver êxito na solicitação, o Amazon S3 retornará uma resposta similar ao seguinte:

```
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 pagamentos do solicitante somente no nível de bucket. Você não pode definir pagamentos pelo solicitante para objetos específicos dentro do bucket.

Você pode configurar um bucket para ser `BucketOwner` ou `Requester` a qualquer momento. No entanto, pode haver alguns minutos antes que o novo valor de configuração entre em vigor.

Note

Proprietários de bucket que abrem mão de URLs pré-assinadas 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.

Recuperar a configuração requestPayment usando a API REST

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 houver êxito na solicitação, o Amazon S3 retornará uma resposta similar ao seguinte:

```
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`.

Fazer 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 DELETE, GET, HEAD, POST e PUT, 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 \(p. 1289\)](#).

Uso dos REST API

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 informações, consulte [Erros Responses \(Respostas com erro\)](#) na Referência de APIs do Amazon Simple Storage Service.

Usar a AWS CLI

Para fazer download de objetos de um bucket de pagamento a cargo do solicitante usando a AWS CLI, especifique `--request-payer requester` como parte da sua solicitação `get-object`. Para obter mais informações, consulte [get-object](#) na Referência da AWS CLI.

Restrições e limitações do bucket

Um bucket do Amazon S3 é pertence à Conta da AWS que o criou. A propriedade do bucket não é transferível para outra conta.

Ao criar um bucket, você escolhe seu nome e a Região da AWS onde criá-lo. Depois de criar um bucket, não é necessário alterar o nome nem a região.

Ao nomear um bucket, escolha um nome que seja relevante para você ou para sua empresa. Evite usar nomes associados a outros. Por exemplo, você deve evitar usar `AWS` ou `Amazon` no nome do bucket.

Por padrão, você pode criar até 100 buckets em cada Contas da AWS. Se precisar de buckets adicionais, você poderá aumentar o limite de bucket da conta para um máximo de 1.000 buckets enviando um aumento de limite de serviço. Não há diferença no desempenho ao usar muitos buckets ou somente alguns.

Para obter informações sobre como aumentar o limite do bucket, acesse [Cotas de serviço da AWS](#) na Referência geral da AWS.

Reutilização de nomes de bucket

Se um bucket estiver vazio, você poderá excluí-lo. Depois de excluído, o nome do bucket fica disponível para reutilização. No entanto, depois de excluir o bucket, talvez você não consiga reutilizar o nome por diversos motivos.

Por exemplo, quando você exclui o bucket e o nome fica disponível para reutilização, uma outra Conta da AWS pode criar um bucket com esse nome. Além disso, pode demorar algum tempo até que seja possível

reutilizar o nome de um bucket excluído. Se você quiser usar o mesmo nome de bucket, recomendamos que você não exclua o bucket.

Para obter mais informações sobre nomes de bucket, consulte [Regras de nomeação de bucket \(p. 127\)](#).

Limitações de objetos e buckets

Não há tamanho máximo para o bucket ou limite para o número de objetos que você pode armazenar em um bucket. Você pode armazenar todos os objetos em um único bucket, ou pode organizá-los em vários buckets. No entanto, você não pode criar um bucket de dentro de outro bucket.

Operações de buckets

A engenharia de alta disponibilidade do Amazon S3 é focada nas operações get, put, list e delete. Como as operações de bucket funcionam utilizando um espaço centralizado e global de recursos, não se recomenda criar, excluir ou configurar buckets no caminho de código de alta disponibilidade da sua aplicação. É melhor criar, excluir ou configurar buckets em uma rotina de inicialização ou configuração separada que você execute com menor frequência.

Nomeação de bucket e buckets criados automaticamente

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.

Para obter mais informações sobre nomeação de bucket, consulte [Regras de nomeação de bucket \(p. 127\)](#).

Fazer upload, fazer download e trabalhar com objetos no Amazon S3

Para armazenar seus dados no Amazon S3, você trabalha com recursos conhecidos como buckets e objetos. Um bucket é um contêiner de objetos. Um objeto é um arquivo e qualquer metadado que descreva esse arquivo.

Para armazenar um objeto no Amazon S3, crie um bucket e faça upload do objeto em um bucket. Quando o objeto estiver no bucket, você poderá abri-lo, baixá-lo e copiá-lo. Quando você não precisa mais de um objeto ou um bucket, você pode limpar esses recursos.

Com o Amazon S3, você paga somente pelo que for usado. Para obter mais informações sobre os recursos e a definição de preço do Amazon S3, consulte [Amazon S3](#). Se você for um novo cliente do Amazon S3, você pode começar a usar o Amazon S3 gratuitamente. Para obter mais informações, consulte [Nível gratuito da AWS](#).

Tópicos

- [Visão geral de objetos Amazon S3 \(p. 161\)](#)
- [Criar nomes de chave de objeto \(p. 162\)](#)
- [Trabalhar com metadados de objeto \(p. 166\)](#)
- [Fazer upload de objetos \(p. 171\)](#)
- [Carregar e copiar objetos usando multipart upload \(p. 180\)](#)
- [Cópia de objetos \(p. 215\)](#)
- [Fazer download de um objeto \(p. 223\)](#)
- [Verificar a integridade do objeto \(p. 229\)](#)
- [Excluir objetos do Amazon S3 \(p. 237\)](#)
- [Organizar, listar e trabalhar com seus objetos \(p. 258\)](#)
- [Usar pre-signed URLs \(p. 273\)](#)
- [Transformar objetos com o S3 Object Lambda \(p. 288\)](#)

Visão geral de objetos Amazon S3

O Amazon S3 é um depósito de objetos que usa valores de chave exclusivos para armazenar quantos objetos você quiser. Esses objetos são armazenados em um ou mais buckets e cada objeto pode ter até 5 TB de tamanho. Um objeto consiste no seguinte:

Chave

O nome que você atribui a um objeto. Você usa a chave de objeto para recuperar o objeto. Para obter mais informações, consulte [Trabalhar com metadados de objetos \(p. 166\)](#).

ID da versão

Em um bucket, uma chave e um ID de versão identificam um objeto de maneira exclusiva. 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 [Usando o versionamento em buckets do S3 \(p. 678\)](#).

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 [Fazer upload de objetos \(p. 171\)](#).

Metadados

Um conjunto de pares de nome-valor com o qual é possível armazenar informações relacionadas ao objeto. Você pode atribuir os metadados, referidos como metadados definidos pelo usuário, a seus objetos no Amazon S3. O Amazon S3 também atribui metadados de sistema a esses objetos, os quais o sistema usa para gerenciar objetos. Para obter mais informações, consulte [Trabalhar com metadados de objetos \(p. 166\)](#).

Sub-recursos

O Amazon S3 usa o mecanismo de sub-recursos 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 do objeto \(p. 162\)](#).

Informações de controle de acesso

Você pode controlar o acesso aos objetos armazenados no Amazon S3. O Amazon S3 é compatível com o controle de acesso baseado em recursos, como uma lista de controle de acesso (ACL) e políticas de bucket, e com o controle de acesso de dados baseados no usuário. Para obter mais informações sobre controle de acesso, consulte:

- [Práticas recomendadas de controle de acesso \(p. 24\)](#)
- [Diretrizes para políticas de acesso \(p. 425\)](#)
- [Identity and Access Management no Amazon S3 \(p. 419\)](#)
- [Configurar ACLs \(p. 596\)](#)

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. Para obter mais informações sobre compartilhamento de objetos, consulte [Compartilhar objetos usando URLs pré-assinados \(p. 275\)](#).

Sub-recursos do objeto

O Amazon S3 define um conjunto de sub-recursos associados a buckets e objetos. Os subrecursos são subordinados aos objetos. Isso significa que os subrecursos não existem por conta própria. Eles são sempre associados a alguma outra entidade, 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 Visão geral da lista de controle de acesso (ACL) (p. 587) .

Criar nomes de chave de objeto

A chave de objeto (ou nome da chave) identifica o objeto em um bucket do Amazon S3 de maneira exclusiva. Metadados de objeto são um conjunto de pares de nome-valor. Para obter mais informações sobre metadados de objeto, consulte [Trabalhar com metadados de objeto \(p. 166\)](#).

Quando você cria um objeto, especifica o nome da chave que, exclusivamente, identifica o objeto no bucket. Por exemplo, no [console do Amazon S3](#), quando você destaca um bucket, uma lista de objetos no bucket é exibida. Esses nomes são as chaves de objeto. O nome da chave do objeto é uma sequência de caracteres Unicode com codificação UTF-8 de até 1.024 bytes de extensão.

Note

Os nomes de chave de objeto com o valor “soap” não são compatíveis com [solicitações de estilo de hospedagem virtual](#). Quanto a valores de nome de chave de objeto em que se usa “soap”, deve-se usar, em vez disso, um [URL de estilo de caminho](#).

O modelo de dados do Amazon S3 é uma estrutura plana: você cria um bucket e o bucket armazena objetos. Não há hierarquia de sub-buckets ou de subpastas. No entanto, é possível pressupor a hierarquia lógica usando prefixos e delimitadores de nome de chave como faz o console do Amazon S3. O console do Amazon S3 é compatível com o conceito de pastas. Para obter mais informações sobre como editar metadados do console do Amazon S3, consulte [Editar metadados de objeto no console do Amazon S3 \(p. 169\)](#).

Vamos supor que seu bucket (`admin-created`) tenha quatro objetos com as seguintes chaves de objeto:

`Development/Projects.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. 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.

- O Amazon S3 é compatível com buckets e objetos e não há nenhuma hierarquia. No entanto, ao usar prefixos e delimitadores em um nome de chave de objeto, o console do Amazon S3 e os AWS SDKs podem pressupor uma hierarquia e apresentar o conceito de pastas.
- O console do Amazon S3 implementa a criação de objetos de pasta criando objetos de zero bytes com o valor de prefixo e do delimitador da pasta como a chave. Esses objetos de pasta não aparecem no console. Caso contrário, eles se comportam como qualquer outro objeto e podem ser visualizados e manipulados por meio da API REST, AWS CLI e AWS SDKs.

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.

Alphanumeric characters

- 0-9
- a-z
- A-Z

Special characters

- Ponto de exclamação (!)
- Hífen (-)
- Sublinhado (_)
- Ponto final (.)
- Asterisco (*)
- Aspas simples ('')
- Abrir parênteses ((
- Fechar parênteses))

Os seguintes são exemplos de nomes de chave válidos:

- 4my-organization
- my.great_photos-2014/jan/myvacation.jpg
- videos/2014/birthday/video1.wmv

Note

Objetos com nomes de chave terminando com pontos “.” e baixados usando o console do Amazon S3 terão os pontos “.” removidos do nome da chave do objeto baixado. Para baixar um objeto com o nome da chave terminado com ponto “.” mantido no objeto baixado, será necessário usar a AWS Command Line Interface (AWS CLI), AWS SDKs ou a API REST.

Além disso, saiba das seguintes limitações do prefixo:

- Objetos com um prefixo de “./” devem ser carregados ou baixados com a AWS Command Line Interface (AWS CLI), AWS SDKs ou a API REST. Você não pode usar o console do Amazon S3.
- Objetos com um prefixo de “..” não podem ser carregados usando a AWS Command Line Interface (AWS CLI) ou o console do Amazon S3.

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 em 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 (“;”)
- Barra (“/”)
- 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 ("|")

Restrições de chave de objeto relacionado a XML

Conforme especificado pelo [padrão XML no processamento de fim de linha](#), todo o texto XML é normalizado de modo que os retornos de carro simples (código ASCII 13) e os retornos de carro imediatamente seguidos por uma nova linha (código ASCII 10) sejam substituídos por um único caractere de nova linha. Para garantir a análise correta de chaves de objeto em solicitações XML, retornos de carro e outros caracteres especiais devem ser substituídos por seu código de entidade XML equivalente quando forem inseridos em tags XML. A seguinte lista mostra os tais caracteres especiais e seus códigos de entidade equivalentes:

- ' como '
- " como "
- & como &
- < como <
- > como >
- \r como  ou 
- \n como
 ou

Example

O exemplo a seguir ilustra o uso de um código de entidade XML como uma substituição para um retorno de carro. Esta solicitação DeleteObjects exclui um objeto com o parâmetro key: /some/prefix/objectwith\rcarriagereturn (onde \r é o retorno de carro).

```
<Delete xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Object>
  <Key>/some/prefix/objectwith&#13;carriagereturn</Key>
</Object>
```

```
</Delete>
```

Trabalhar com metadados de objeto

Você pode definir metadados de objeto no Amazon S3 no momento do upload do objeto. Metadados de objeto são um conjunto de pares de nome-valor. 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.

Ao criar um objeto, você também especifica o nome da chave, que identifica exclusivamente o objeto no bucket. A chave de objeto (ou nome da chave) identifica o objeto em um bucket do Amazon S3 de maneira exclusiva. Para obter mais informações, consulte [Criar nomes de chave de objeto \(p. 162\)](#).

Há dois tipos de metadados no Amazon S3: definidos pelo sistema e definidos pelo usuário. As seções abaixo fornecem mais informações sobre metadados definidos pelo sistema e definidos pelo usuário. Para obter mais informações sobre como editar metadados usando o console do Amazon S3, consulte [Editar metadados de objeto no console do Amazon S3 \(p. 169\)](#).

Metadados do objeto definidos pelo sistema

Para cada objeto armazenado em um bucket, o Amazon S3 mantém um conjunto de metadados do sistema. O Amazon S3 processa estes metadados do sistema conforme necessário. Por exemplo, o Amazon S3 mantém a data de criação e o tamanho dos metadados e usa essas informações como parte do gerenciamento do objeto.

Existem duas categorias de metadados de sistema:

1. Os metadados, como a data de criação do objeto, são controlados pelo sistema e somente 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 é um objeto no bucket. O Amazon S3 armazena o valor de redirecionamento da página como metadados do sistema com valores que você controla.

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 [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#).

O Amazon S3 usa chaves AWS KMS para criptografar seus objetos do Amazon S3. O AWS KMS criptografa apenas os dados do objeto. A soma de verificação e o algoritmo especificado são armazenados como parte dos metadados do objeto. Se a criptografia no lado do servidor for solicitada para o objeto, a soma de verificação será armazenada em formato criptografado. Para obter mais informações sobre criptografia no lado do servidor, consulte [Proteção de dados usando criptografia \(p. 358\)](#).

Note

O cabeçalho da solicitação PUT é limitado a 8 KB. No cabeçalho da solicitação PUT, os metadados definidos pelo sistema são limitados a 2 KB. O tamanho de metadados definidos pelo sistema é medido pela soma do número de bytes na codificação US-ASCII de cada chave e valor.

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
Cache-Control	Um campo de cabeçalho geral usado para especificar políticas de armazenamento em cache.	Sim
Content-Disposition	Informações de apresentação do objeto.	Sim
Content-Length	Tamanho de objeto em bytes.	Não
Content-Type:	Tipo de objeto.	Sim
Última modificação	Data de criação do objeto ou data da última modificação, o que aconteceu por último.	Não
ETag	Representa uma versão específica de um objeto. Para objetos que não são carregados como um carregamento fracionado e não são criptografados ou são criptografados por criptografia no servidor com chaves gerenciadas pelo Simple Storage Service (Amazon S3) (SSE-S3), a ETag é um resumo MD5 dos dados.	Não
x-amz-server-side-encryption	Indica se a criptografia no lado do servidor está habilitada para o objeto e se essa criptografia é do AWS Key Management Service (AWS KMS) ou da criptografia gerenciada pelo Amazon S3 (SSE-S3). Para obter mais informações, consulte Proteção de dados usando criptografia no lado do servidor (p. 359) .	Sim
x-amz-checksum-crc32, x-amz-checksum-crc32c, x-amz-checksum-sha1, x-amz-checksum-sha256	Contém a soma de verificação ou o resumo do objeto. No máximo um desses cabeçalhos será definido por vez, dependendo do algoritmo de soma de verificação que você instruir o Amazon S3 a usar. Para obter mais informações sobre a escolha do algoritmo de soma de verificação, consulte Verificar a integridade do objeto (p. 229) .	Não
x-amz-version-id	Versão do objeto. 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 Usando o versionamento em buckets do S3 (p. 678) .	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 Uso de classes de armazenamento do Amazon S3 (p. 731) .	Sim
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 (Opcional) Configurar um redirecionamento de uma página da Web (p. 1203) . Esse valor é exclusivo para cada objeto e não é copiado por padrão.	Sim

Nome	Descrição	O usuário pode modificar o valor?
x-amz-server-side-encryption-aws-kms-key-id	Se x-amz-server-side-encryption estiver presente e tiver o valor de aws:kms, isso indicará o ID da chave do KMS de criptografia simétrica do 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 Proteger dados usando a criptografia de servidor com chaves de criptografia fornecidas pelo cliente (SSE-C) (p. 390).	Sim
x-amz-tagging	O conjunto de etiquetas do objeto. O conjunto de etiquetas deve ser codificado como parâmetros de consulta de URL.	Sim

Metadados do objeto 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 a SOAP via HTTP está obsoleto, mas o SOAP continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com SOAP. Em vez de SOAP, recomendamos usar a API REST ou os AWS SDKs.

Quando os metadados são recuperados por meio da API REST, o Amazon S3 combina os cabeçalhos que têm o mesmo nome (sem distinção entre letras maiúsculas e minúsculas) em uma lista delimitada por vírgulas. 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. A ação HeadObject recupera metadados de um objeto sem retornar o próprio objeto. Essa operação é útil se você estiver interessado apenas nos metadados de um objeto. Para usar HEAD, você deve ter acesso READ ao objeto. Para obter mais informações, consulte [HeadObject](#) na Referência de API do Amazon Simple Storage Service.

Metadados definidos pelo usuário são um conjunto de pares chave-valor. O Amazon S3 armazena chaves de metadados definidas pelo usuário em minúsculas.

O Amazon S3 permite caracteres Unicode arbitrários em seus valores de metadados.

Para evitar problemas em torno da apresentação desses valores de metadados, é necessário estar em conformidade com o uso de caracteres US-ASCII ao usar REST e UTF-8 ao usar SOAP ou uploads baseados em navegador via POST.

Ao usar caracteres que não pertencem ao US-ASCII em seus valores de metadados, a string Unicode fornecida é examinada quanto à presença desses caracteres. Os caracteres desses cabeçalhos são decodificados conforme o [RFC 2047](#) antes do armazenamento e codificados de acordo com o [RFC 2047](#) antes de retornar para garantir que eles possam ser enviados por e-mail com segurança. Se a string contiver apenas caracteres US-ASCII, ela será apresentada como está.

Veja um exemplo a seguir.

```
PUT /Key HTTP/1.1
Host: awsexamplebucket1.s3.amazonaws.com
x-amz-meta-nonascii: ÄMÄZÖN S3

HEAD /Key HTTP/1.1
Host: awsexamplebucket1.s3.amazonaws.com
x-amz-meta-nonascii: =?UTF-8?B?w4PChE3Dg8KEWsODwpXDg8KRIFMz?=

PUT /Key HTTP/1.1
Host: awsexamplebucket1.s3.amazonaws.com
x-amz-meta-ascii: AMAZONS3

HEAD /Key HTTP/1.1
Host: awsexamplebucket1.s3.amazonaws.com
x-amz-meta-ascii: AMAZONS3
```

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 informações sobre como alterar os metadados do seu objeto depois do upload criando uma cópia, modificando e substituindo o objeto antigo ou criando uma nova versão dele, consulte [Editar metadados de objeto no console do Amazon S3 \(p. 169\)](#).

Editar metadados de objeto no console do Amazon S3

Você pode usar o console do Amazon S3 para editar metadados de objetos do S3 existentes. Alguns metadados são definidos pelo Amazon S3 quando você faz upload do objeto. Por exemplo, Content-Length é a chave (nome) e o valor é o tamanho do objeto em bytes.

Você também pode definir alguns metadados ao carregar o objeto e depois editá-lo conforme suas necessidades mudarem. Por exemplo, você pode ter um conjunto de objetos que você armazena inicialmente na classe de armazenamento STANDARD. Com o tempo, talvez você não precise mais que esses dados estejam altamente disponíveis. Assim, você altera a classe de armazenamento para GLACIER editando o valor da chave x-amz-storage-class de STANDARD para GLACIER.

Note

Considere os seguintes problemas ao editar metadados de objeto no Amazon S3:

- Essa ação cria uma cópia do objeto com configurações atualizadas e a data da última modificação. Se o versionamento do S3 estiver habilitado, uma nova versão do objeto será criada e o objeto existente se tornará uma versão mais antiga. Se o versionamento do S3 não estiver habilitado, uma nova cópia do objeto substituirá o original. A Conta da AWS associada à função do IAM que altera a propriedade também se torna a proprietária do novo objeto ou (versão do objeto).
- A edição de metadados atualiza valores para nomes de chaves existentes.
- Objetos criptografados com chaves de criptografia fornecidas pelo cliente (SSE-C) não podem ser copiados usando o console. Você deve usar a AWS CLI, o AWS SDK ou a API REST do Amazon S3.

Warning

Ao editar metadados de pastas, aguarde a conclusão da operação `Edit metadata` antes de adicionar novos objetos à pasta. Caso contrário, novos objetos também podem ser editados.

Os tópicos a seguir descrevem como editar metadados de um objeto usando o console do Amazon S3.

Editar metadados definidos pelo sistema

Você pode configurar alguns metadados do sistema para um objeto do S3, mas não todos. Para obter uma lista de metadados definidos pelo sistema e saber se você pode modificar seus valores, consulte [Metadados do objeto definidos pelo sistema \(p. 166\)](#).

Como editar metadados definidos pelo sistema de um objeto

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Navegue até seu bucket ou pasta do Amazon S3 e marque a caixa de seleção à esquerda dos nomes dos objetos com metadados que você deseja editar.
3. No menu Actions (Ações), escolha Edit actions (Editar ações) e escolha Edit metadata (Editar metadados).
4. Revise os objetos listados e escolha Add metadata (Adicionar metadados).
5. Para Type (Tipo) de metadados, selecione System-defined (Definidos pelo sistema).
6. Especifique uma key (chave) exclusiva e o value (valor) dos metadados.
7. Para editar metadados adicionais, escolha Add metadata (Adicionar metadados). Você também pode escolher Remove (Remover) para remover um conjunto de valores de chave de tipo.
8. Quando terminar, escolha Edit metadata (Editar metadados) e o Amazon S3 editará os metadados dos objetos especificados.

Editar metadados definidos pelo usuário

Você pode editar metadados definidos pelo usuário de um objeto combinando o prefixo de metadados, `x-amz-meta-`, e um nome escolhido para criar uma chave personalizada. Por exemplo, se você adicionar o nome personalizado `alt-name`, a chave de metadados será `x-amz-meta-alt-name`.

Metadados definidos pelo usuário podem ter até 2 KB no total. Para calcular o tamanho total dos metadados definidos pelo usuário, some o número de bytes na codificação UTF-8 referente a cada chave e valor. As duas chaves e seus valores devem estar em conformidade com os padrões US-ASCII. Para obter mais informações, consulte [Metadados do objeto definidos pelo usuário \(p. 168\)](#).

Como editar metadados definidos pelo usuário de um objeto

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket que contém os objetos aos quais você deseja adicionar metadados.
Você também pode, opcionalmente, navegar até uma pasta.
3. Na lista Objects (Objetos), marque a caixa de seleção ao lado dos nomes dos objetos aos quais você deseja adicionar metadados.
4. No menu Actions (Ações), escolha Edit metadata (Editar metadados).
5. Revise os objetos listados e escolha Add metadata (Adicionar metadados).
6. Para Type (Tipo) de metadados, escolha User-defined (Definido pelo usuário).
7. Insira uma Key (Chave) personalizada única após `x-amz-meta-`. Insira também um value (valor) dos metadados.
8. Para adicionar metadados extras, escolha Add metadata (Adicionar metadados). Você também pode escolher Remove (Remover) para remover um conjunto de valores de chave de tipo.

9. Escolha Edit metadata (Editar metadados).

O Amazon S3 edita os metadados dos objetos especificados.

Fazer upload de objetos

Quando você faz upload de um arquivo no Amazon S3, ele é armazenado como um objeto do S3. Os objetos consistem em dados e metadados de arquivo que descrevem o objeto. Você pode ter um número ilimitado de objetos em um bucket. Antes de fazer upload de arquivos em um bucket do Amazon S3, você precisa escrever permissões para o bucket. Para obter mais informações sobre permissões de acesso, consulte [Identity and Access Management no Amazon S3 \(p. 419\)](#).

Você pode fazer upload de qualquer tipo de arquivo (imagens, backups, dados, filmes, etc.) para um bucket do S3. O tamanho máximo de arquivo que você pode carregar usando o console do Amazon S3 é de 160 GB. Para fazer upload de um arquivo com mais de 160 GB, use a AWS CLI, o AWS SDK ou a API REST do Amazon S3.

Se você fizer upload de um objeto com um nome de chave que já existe em um bucket com versionamento habilitado, o Amazon S3 criará outra versão de objeto em vez de substituir o objeto existente. Para obter mais informações sobre versionamento, consulte [Uso do console do S3 \(p. 684\)](#).

Dependendo do tamanho de dados enviados por upload, o Amazon S3 oferece as seguintes opções:

- Fazer upload de um objeto em uma única operação usando AWS SDKs, a API REST ou a AWS CLI: com uma única operação PUT, você pode fazer upload de um único objeto com até 5 GB.
- Fazer upload de um único objeto usando o console do Amazon S3: Com o console do Amazon S3, é possível fazer upload de um único objeto com até 160 GB de tamanho.
- Fazer upload de um objeto em partes usando AWS SDKs, a API REST ou a AWS CLI: com a API de upload fracionado, é possível fazer upload de um único objeto grande, com até 5 TB.

A API multipart upload API foi projetada para melhorar a experiência de upload de objetos maiores. É possível fazer upload de um objeto 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 [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).

Ao fazer upload de um objeto, é possível solicitar, opcionalmente, que o Amazon S3 criptografe o objeto antes de salvá-lo no disco e descriptografá-lo quando você fizer o download. Para obter mais informações, consulte [Proteção de dados usando criptografia \(p. 358\)](#).

Uso do console do S3

Este procedimento explica como fazer upload de objetos e pastas para um bucket S3 usando o console.

Quando você faz upload de um objeto, o nome da chave de objeto é o nome do arquivo e quaisquer prefixos opcionais. No console do Amazon S3, você pode criar pastas para organizar seus objetos. No Amazon S3, as pastas são representadas como prefixos que aparecem no nome da chave de objeto. Se você fizer upload um objeto individual para uma pasta no console do Amazon S3, o nome da pasta será incluído no nome da chave do objeto.

Por exemplo, se você carregar um objeto chamado sample1.jpg para uma pasta chamada backup, o nome da chave será backup/sample1.jpg. Contudo, o objeto é exibido no console como sample1.jpg na pasta backup. Para obter mais informações sobre nomes de chave, consulte [Trabalhar com metadados de objeto \(p. 166\)](#).

Note

Se você renomear um objeto ou alterar qualquer uma das propriedades no console do Amazon S3, por exemplo Storage Class (Classe de armazenamento), Encryption (Criptografia), Metadata (Metadados), um novo objeto será criado para substituir o antigo. Se o versionamento do S3 estiver habilitado, uma nova versão do objeto será criada e o objeto existente se tornará uma versão mais antiga. A função que altera a propriedade também se torna o proprietário do novo objeto ou (versão do objeto).

Quando você faz upload de uma pasta, o Amazon S3 faz upload de todos os arquivos e subpastas da pasta especificada em seu bucket. Ele então atribui um nome de chave de objeto que é uma combinação do nome de arquivo carregado com o nome da pasta. Por exemplo, se você fizer upload de uma pasta chamada /images que contém dois arquivos, sample1.jpg e sample2.jpg, o Amazon S3 fará upload dos arquivos e atribuirá a eles os nomes de chave correspondentes, images/sample1.jpg e images/sample2.jpg. Os nomes de chave incluem o nome da pasta como um prefixo. O console do Amazon S3 exibe somente a parte do nome de chave que vem depois da última /. Por exemplo, em uma pasta de imagens, os objetos images/sample1.jpg e images/sample2.jpg são exibidos como sample1.jpg e um sample2.jpg.

Para fazer upload de pastas e arquivos para um bucket do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket no qual você deseja carregar suas pastas ou arquivos.
3. Escolha Upload (Fazer upload).
4. Na janela Upload (Fazer upload), siga um destes procedimentos:
 - Arraste e solte arquivos e pastas para a janela Upload (Fazer upload) .
 - Escolha Add file (Adicionar arquivo) ou Add folder (Adicionar pasta), escolha arquivos ou pastas para fazer upload e depois escolha Open (Abrir).
5. Para habilitar o versionamento, em Destination (Destino), escolha Enable Bucket Versioning (Ativar versionamento de bucket).
6. Para fazer upload dos arquivos e pastas listados sem configurar opções de upload adicionais, na parte inferior da página, escolha Upload (Fazer upload).

O Amazon S3 faz o upload de seus objetos e pastas. Quando o carregamento for concluído, você verá uma mensagem de sucesso na página Upload: status.

Para configurar propriedades de objeto adicionais

1. Para alterar as permissões da lista de controle de acesso, escolha Permissions (Permissões).
2. Em Access control list (ACL) (Lista de controle de acesso (ACL)), edite as permissões.

Para informações sobre permissões de acesso a objeto, consulte [Usar o console do S3 para definir permissões de ACL para um objeto \(p. 599\)](#). Você pode conceder acesso de leitura aos seus objetos ao público (todos no mundo), para todos os arquivos que você está carregando. No entanto, recomendamos não alterar a configuração padrão para acesso de leitura público. Conceder acesso público de leitura é aplicável a um pequeno subconjunto de casos de uso, como quando buckets são usados para sites. Você sempre pode alterar as permissões de objeto depois de carregar o objeto.

3. Para configurar outras propriedades, escolha Properties (Propriedades).
4. Na seção Storage class (Classe de armazenamento) escolha a classe de armazenamento para os arquivos que você está fazendo upload.

Para obter mais informações sobre classes de armazenamento, consulte [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#).

5. Para atualizar as configurações de criptografia para seus objetos, em Server-side encryption settings (Configurações de criptografia do lado do servidor), faça o seguinte.
 - a. Escolha Specify an encryption key (Especificar uma chave de criptografia).
 - b. Para criptografar os arquivos carregados usando chaves gerenciadas pelo Amazon S3, escolha Amazon S3 key (Chave do Amazon S3) (SSE-S3).

Para obter mais informações, consulte [Proteção de dados usando criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 \(SSE-S3\)](#) (p. 378).
 - c. Para criptografar os arquivos carregados usando o AWS Key Management Service (AWS KMS), escolha AWS Key Management Service chave (SSE-KMS). Em seguida, escolha uma opção para a chave do AWS KMS.
 - Chave gerenciada pela AWS: escolha uma [chave gerenciada pela AWS](#).
 - Choose from your KMS root keys (Escolher entre as chaves raiz do KMS): escolha uma [chave gerenciada pelo cliente](#) de uma lista de chaves do KMS na mesma Região que seu bucket.

Para obter mais informações sobre como criar uma chave gerenciada pelo cliente, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service. Para obter mais informações sobre como proteger dados com o AWS KMS, consulte [Proteção de dados usando criptografia do servidor com AWS Key Management Service \(SSE-KMS\)](#) (p. 360).

- Enter KMS root key ARN (Inserir o ARN da chave raiz do KMS): especifique o ARN da chave do AWS KMS para uma chave gerenciada pelo cliente e insira o nome do recurso da Amazon (ARN).

Você pode usar o ARN da chave-raiz do KMS para dar a uma conta externa a capacidade de usar um objeto protegido por uma AWS KMS key. Para fazer isso, escolha Enter KMS master key ARN (Inserir o ARN da chave raiz do KMS) e insira o nome de recurso da Amazon (ARN) para a conta externa. Os administradores de uma conta externa com permissões de uso para um objeto protegido pela chave do KMS podem restringir ainda mais o acesso criando um política do IAM no nível do recurso.

Note

Para criptografar objetos em um bucket, você pode usar apenas as AWS KMS keys que estão disponíveis na mesma Região da AWS como o bucket.

6. Para usar somas de verificação adicionais, escolha On (Ativar). Em seguida, em Checksum function (Função de soma de verificação), escolha a função que você gostaria de usar. O Amazon S3 calcula e armazena o valor da soma de verificação depois de receber o objeto inteiro. Você pode usar a caixa Precalculated value (Valor pré-calculado) para fornecer esse valor. Se fizer isso, o Amazon S3 vai comparar o valor que você forneceu com o valor calculado. Se os dois valores não corresponderem, o Amazon S3 gerará um erro.

As somas de verificação adicionais permitem que você especifique o algoritmo de soma de verificação que gostaria de usar para verificar seus dados. Para obter mais informações sobre somas de verificação adicionais, consulte [Verificar a integridade do objeto](#) (p. 229).

7. Para adicionar tags a todos os objetos que você está carregando, escolha Add tag (Adicionar tag). Digite um nome de tag no campo Key (Chave) . Digite um valor para a tag.

A marcação de objetos é uma forma de categorizar o armazenamento. Cada tag é um par de chave-valor. Os valores de chave e tag diferenciam maiúsculas de minúsculas. É possível ter até dez tags por objeto. Um chave de tag pode ter até 128 caracteres Unicode e os valores de tag podem ter até 255 caracteres Unicode. Para obter mais informações sobre tags de objeto, consulte [Categorizando seu armazenamento usando tags](#) (p. 878).

8. Para adicionar metadados, escolha Add metadata (Adicionar metadados).

- a. Em Type (Tipo), escolha System defined (Definido pelo sistema) ou User defined (Definido pelo usuário).

Para metadados definidos pelo sistema, você pode selecionar cabeçalhos HTTP comuns, como Content-Type e Content-Disposition. Para obter uma lista de metadados definidos pelo sistema e informações sobre a possibilidade de adicionar o valor, consulte [Metadados do objeto definidos pelo sistema \(p. 166\)](#). Todos os metadados que começam com o prefixo x-amz-meta- são tratados como metadados definidos pelo usuário. Os metadados definidos pelo usuário são armazenados com o objeto e retornados quando você baixa o objeto. As chaves e seus valores devem estar em conformidade com os padrões US-ASCII. Metadados definidos pelo usuário podem ter até 2 KB. Para obter mais informações sobre metadados definidos pelo sistema e pelo usuário, consulte [Trabalhar com metadados de objeto \(p. 166\)](#).

- b. Para Key (Chave), escolha uma chave.

- c. Digite um valor para a chave.

9. Para carregar seus objetos, escolha Upload (Fazer upload).

O Amazon S3 faz o upload do objeto. Quando o upload for concluído, você pode ver uma mensagem de sucesso na página de Upload: status.

10. Selecione Exit (Sair).

Uso da SDKs AWS

Você pode usar o AWS SDK para fazer upload de objetos no Amazon S3. O SDK fornece bibliotecas wrapper para você para fazer upload de dados com facilidade. Para obter informações, consulte a [Lista de SDKs compatíveis](#).

Aqui estão alguns exemplos com alguns SDKs selecionados:

.NET

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 do Amazon S3 .NET \(p. 1268\)](#).

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

namespace Amazon.DocSamples.S3
{
    class UploadObjectTest
    {
        private const string bucketName = "*** bucket name ***";
        // For simplicity the example creates two objects from the same file.
        // You specify key names for these objects.
        private const string keyName1 = "*** key name for first object created ***";
        private const string keyName2 = "*** key name for second object created ***";
```

```
private const string filePath = @ "*** file path ***";
private static readonly RegionEndpoint bucketRegion = RegionEndpoint.EUWest1;

private static IAmazonS3 client;

public static void Main()
{
    client = new AmazonS3Client(bucketRegion);
    WritingAnObjectAsync().Wait();
}

static async Task WritingAnObjectAsync()
{
    try
    {
        // 1. Put object-specify only key name for the new object.
        var putRequest1 = new PutObjectRequest
        {
            BucketName = bucketName,
            Key = keyName1,
            ContentBody = "sample text"
        };

        PutObjectResponse response1 = await client.PutObjectAsync(putRequest1);

        // 2. Put the object-set ContentType and add metadata.
        var putRequest2 = new PutObjectRequest
        {
            BucketName = bucketName,
            Key = keyName2,
            FilePath = filePath,
            ContentType = "text/plain"
        };

        putRequest2.Metadata.Add("x-amz-meta-title", "someTitle");
        PutObjectResponse response2 = await client.PutObjectAsync(putRequest2);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine(
            "Error encountered ***. Message:'{0}' when writing an object"
            , e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine(
            "Unknown encountered on server. Message:'{0}' when writing an
object"
            , e.Message);
    }
}
}
```

Java

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 como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectMetadata;
import com.amazonaws.services.s3.model.PutObjectRequest;

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

public class UploadObject {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String stringObjKeyName = "*** String object key name ***";
        String fileObjKeyName = "*** File object key name ***";
        String fileName = "*** Path to file to upload ***";

        try {
            //This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withRegion(clientRegion)
                .build();

            // Upload a text string as a new object.
            s3Client.putObject(bucketName, stringObjKeyName, "Uploaded String Object");

            // Upload a file as a new object with ContentType and title specified.
            PutObjectRequest request = new PutObjectRequest(bucketName, fileObjKeyName,
new File(fileName));
            ObjectMetadata metadata = new ObjectMetadata();
            metadata.setContentType("plain/text");
            metadata.addUserMetadata("title", "someTitle");
            request.setMetadata(metadata);
            s3Client.putObject(request);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

JavaScript

O exemplo a seguir carrega um arquivo existente para um bucket do Amazon S3 em uma região específica.

```
// Import required AWS SDK clients and commands for Node.js.
import { PutObjectCommand } from "@aws-sdk/client-s3";
```

```
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
// Amazon S3 service client module.
import {path} from "path";
import {fs} from "fs";

const file = "OBJECT_PATH_AND_NAME"; // Path to and name of object. For example '../
myFiles/index.js'.
const fileStream = fs.createReadStream(file);

// Set the parameters
export const uploadParams = {
  Bucket: "BUCKET_NAME",
  // Add the required 'Key' parameter using the 'path' module.
  Key: path.basename(file),
  // Add the required 'Body' parameter
  Body: fileStream,
};

// Upload file to specified bucket.
export const run = async () => {
  try {
    const data = await s3Client.send(new PutObjectCommand(uploadParams));
    console.log("Success", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

PHP

Este tópico oriente sobre o uso de classes do AWS SDK for 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 [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).

Este tópico considera que você já está seguindo as instruções para [Usar o AWS SDK for PHP e executar exemplos do PHP \(p. 1268\)](#) e tem o AWS SDK for PHP devidamente instalado.

Example Criar um objeto em um bucket do Amazon S3 fazendo upload dos dados

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 dos exemplos de PHP neste guia, consulte [Executar exemplos do PHP \(p. 1269\)](#).

```
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;  
}
```

Ruby

O AWS SDK for Ruby versão 3 oferece 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"  
  
# Wraps Amazon S3 object actions.  
class ObjectUploadFileWrapper  
    attr_reader :object  
  
    # @param object [Aws::S3::Object] An existing Amazon S3 object.  
    def initialize(object)  
        @object = object  
    end  
  
    # Uploads a file to an Amazon S3 object by using a managed uploader.  
    #  
    # @param file_path [String] The path to the file to upload.  
    # @return [Boolean] True when the file is uploaded; otherwise false.  
    def upload_file(file_path)  
        @object.upload_file(file_path)  
        true  
    rescue Aws::Errors::ServiceError => e  
        puts "Couldn't upload file #{file_path} to #{@object.key}. Here's why:  
#{e.message}"  
        false  
    end  
end  
  
def run_demo  
    bucket_name = "doc-example-bucket"  
    object_key = "my-uploaded-file"  
    file_path = "object_upload_file.rb"  
  
    wrapper = ObjectUploadFileWrapper.new(Aws::S3::Object.new(bucket_name, object_key))  
    return unless wrapper.upload_file(file_path)  
  
    puts "File #{file_path} successfully uploaded to #{bucket_name}:#{object_key}."  
end
```

```
run_demo if $PROGRAM_NAME == __FILE__
```

A segunda forma que o AWS SDK for 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"

# Wraps Amazon S3 object actions.
class ObjectPutWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object(source_file_path)
    File.open(source_file_path, "rb") do |file|
      @object.put(body: file)
    end
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put #{source_file_path} to #{object.key}. Here's why: #{e.message}"
    false
  end
end

def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object-key"
  file_path = "my-local-file.txt"

  wrapper = ObjectPutWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  success = wrapper.put_object(file_path)
  return unless success

  puts "Put file #{file_path} into #{object_key} in #{bucket_name}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Uso dos REST API

Você pode enviar solicitações REST para fazer upload de um objeto. Você pode enviar uma solicitação PUT para fazer upload de dados em uma única operação. Para obter mais informações, consulte [Objeto PUT](#).

Usar a AWS CLI

Você pode enviar uma solicitação PUT para fazer upload de um objeto de até 5 GB em uma única operação. Para obter mais informações, consulte o exemplo de [PutObject](#) na Referência de comandos da AWS CLI.

Carregar e copiar objetos usando multipart upload

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 forem carregadas, o Amazon S3 montará essas partes e criará 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:

- Improved throughput (Throughput aprimorada): você pode carregar as partes em paralelo para melhorar o throughput.
- Quick recovery from any network issues (Recuperação rápida de qualquer problema de rede): o tamanho menor das partes minimiza o impacto de reiniciar um carregamento que falhou devido a um erro de rede.
- Pause and resume object uploads (Pausar e retomar carregamentos de objetos): você pode carregar as partes do objeto ao longo do tempo. Após ser iniciado um carregamento fracionado, ele não expira; você deve concluir ou interromper explicitamente o carregamento fracionado.
- Begin an upload before you know the final object size (Começar um carregamento antes de saber o tamanho final do objeto): você pode carregar um objeto à medida que ele é criado.

Recomendamos que você use o carregamento fracionado das seguintes maneiras:

- Se você estiver fazendo upload de objetos grandes 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 performance com vários threads.
- 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 carregamento fracionado, tente carregar novamente apenas as partes que foram interrompidas durante o carregamento. Você não precisa reiniciar o upload do seu objeto do começo.

Processo multipart upload

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 poderá 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. É necessário incluir esse ID de upload sempre que fizer upload de partes, listar as partes, concluir um upload ou interromper 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.

Carregar 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 escolheu não precisa estar em uma sequência consecutiva (por exemplo, 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 carregar uma parte, o Amazon S3 retornará um cabeçalho entity tag (ETag) etiqueta de entidade (ETag) na resposta. Para cada upload de parte, você deve registrar o número de parte e o valor de ETag. Você tem que incluir esses valores na solicitação subsequente para concluir o multipart upload.

Note

Depois de iniciar um carregamento fracionado e carregar uma ou mais partes, é necessário concluir ou interromper o carregamento fracionado para não ser mais cobrado pelo armazenamento de partes carregadas. Somente depois que você concluir ou interromper um multipart upload é que o Amazon S3 liberará o armazenamento das partes e deixará de cobrar pelo armazenamento das partes.

Depois de interromper um multipart upload, você não pode fazer upload de nenhuma parte usando esse ID de upload novamente. Se algum carregamento de parte estiver em andamento, ainda assim ele poderá ser bem-sucedido ou falhar mesmo depois da interrupção. Para liberar todo o armazenamento consumido por todas as partes, você deve interromper um carregamento fracionado somente depois que todos os carregamentos de partes tiverem sido concluídos.

Conclusão de carregamento fracionado

Quando você concluir um multipart upload, o Amazon S3 criará um objeto concatenando as partes em ordem crescente com base no número da parte. Se algum metadado de objeto for fornecido na solicitação iniciar 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 concluir multipart upload deve incluir o ID de upload e uma lista dos números de parte e dos valores de ETag correspondentes. A resposta do Amazon S3 inclui um ETag que identifica exclusivamente os dados do objeto combinados. Esse ETag não é necessariamente um hash MD5 dos dados do objeto.

Exemplos de chamada de carregamento fracionado

Para este exemplo, suponha que você esteja gerando um carregamento fracionado para um arquivo de 100 GB. Nesse caso, você teria as seguintes chamadas de API para todo o processo. Haveria um total de 1.002 chamadas de API.

- Uma chamada [CreateMultipartUpload](#) para iniciar o processo.
- 1.000 chamadas [UploadPart](#) individuais, cada uma carregando uma parte de 100 MB, para um tamanho total de 100 GB.
- Uma chamada [CompleteMultipartUpload](#) para finalizar o processo.

Listagens de carregamento fracionado

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 aquelas cujo carregamento não foi concluído. Usando a operação list multipart uploads (listar carregamentos fracionados), você pode obter uma lista daqueles que estão em andamento.

Um multipart upload em andamento é um upload que você iniciou, mas que ainda não concluiu nem interrompeu. Cada solicitação retorna no máximo 1.000 multipart uploads. Se houver mais de 1.000

carregamento fracionados em andamento, você precisará enviar solicitações adicionais para recuperar os carregamento fracionados restantes. Use a listagem retornada apenas para verificação. Não use o resultado dessa listagem ao enviar uma solicitação complete multipart upload (concluir carregamento fracionado). Em vez disso, mantenha sua própria lista de números de parte que você especificou ao carregar as partes e os valores correspondentes de ETag que o Amazon S3 retorna.

Somas de verificação com operações de carregamento fracionado

Ao carregar um objeto no Amazon S3, você pode especificar um algoritmo de soma de verificação a ser usado pelo Amazon S3. O Amazon S3 usa o MD5 por padrão para verificar a integridade dos dados, mas você pode especificar um algoritmo de soma de verificação adicional a ser usado. Ao usar o MD5, o Amazon S3 calcula a soma de verificação de do objeto fracionado completo após a conclusão do carregamento. Essa soma de verificação não é do objeto completo, mas das somas de verificação de cada parte.

Quando você instrui o Amazon S3 a usar somas de verificação adicionais, ele calcula o valor delas para cada parte e armazena os valores. Você pode usar a API ou o SDK para recuperar o valor da soma de verificação para partes específicas usando `GetObject` ou `HeadObject`.

Important

Se estiver usando um carregamento fracionado com somas de verificação adicionais, os números das partes fracionadas devem ser consecutivos. Ao usar somas de verificação adicionais, se tentar concluir uma solicitação de carregamento fracionado com números de parte não consecutivos, o Amazon S3 gerará um erro HTTP 500 Internal Server Error.

Para obter mais informações sobre como as somas de verificação funcionam com objetos fracionados, consulte [Verificar a integridade do objeto \(p. 229\)](#).

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, sua aplicação 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 do S3 habilitado, a conclusão de um carregamento fracionado sempre criará uma 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

Depois que você iniciar um multipart upload, o Amazon S3 reterá todas as partes até você concluir ou interromper 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 interromper 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 preços, consulte [Definição de preços do Amazon S3](#).

Suporte de API para multipart upload

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 na Referência de APIs do Amazon Simple Storage Service descrevem a API REST para multipart upload.

- [Criar carregamento fracionado](#)
- [Carregar parte](#)
- [Carregar parte \(copiar\)](#)
- [Concluir carregamento fracionado](#)
- [Anular carregamento fracionado](#)
- [Listar partes](#)
- [Listar carregamentos fracionados](#)

Suporte da AWS Command Line Interface para carregamento fracionado

Os tópicos a seguir na AWS Command Line Interface descrevem as operações de carregamento fracionado.

- [Iniciar multipart upload](#)
- [Carregar parte](#)
- [Carregar parte \(copiar\)](#)
- [Concluir carregamento fracionado](#)
- [Anular carregamento fracionado](#)
- [Listar partes](#)
- [Listar carregamentos fracionados](#)

Suporte do AWS SDK para upload fracionado

Você pode usar AWS SDKs para fazer upload de um objeto em partes. Para obter uma lista dos AWS SDKs compatíveis com a ação da API, consulte:

- [Criar carregamento fracionado](#)
- [Carregar parte](#)
- [Carregar parte \(copiar\)](#)
- [Concluir carregamento fracionado](#)
- [Anular carregamento fracionado](#)
- [Listar partes](#)
- [Listar carregamentos fracionados](#)

API de multipart upload e permissões

Você deve ter as permissões necessárias para usar as operações do multipart upload. É possível usar listas de controle de acesso (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, uma política de bucket ou uma política de usuário.

Ação	Permissões obrigatórias
Criar multipart upload	<p>Você deve ter permissão para realizar a ação <code>s3:PutObject</code> em um objeto para criar o multipart upload.</p> <p>O proprietário do bucket pode permitir que outros principais realizem a ação <code>s3:PutObject</code>.</p>
Iniciar multipart upload	<p>Você deve ter permissão para realizar a ação <code>s3:PutObject</code> em um objeto para iniciar o multipart upload.</p> <p>O proprietário do bucket pode permitir que outros principais realizem a ação <code>s3:PutObject</code>.</p>
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 Owner (Proprietário). Se o iniciador for um usuário do IAM, esse elemento fornecerá o ARN e o nome da exibição do usuário.
Carregar parte	<p>Você deve ter permissão para realizar a ação <code>s3:PutObject</code> em um objeto para fazer upload de uma parte.</p> <p>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.</p>
Carregar parte (Copiar)	<p>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.</p> <p>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.</p>
Concluir carregamento fracionado	<p>Você deve ter permissão para realizar a ação <code>s3:PutObject</code> em um objeto para concluir o multipart upload.</p> <p>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.</p>
Parar o multipart upload	<p>Você deve ter permissão para realizar a ação <code>s3:AbortMultipartUpload</code> em um objeto para interromper um multipart upload.</p> <p>Por padrão, o proprietário do bucket e o iniciador do carregamento fracionado têm permissão para executar essa ação como parte das políticas do IAM e de buckets. Se o iniciador for um usuário do IAM, a Conta da AWS desse usuário também terá permissão para interromper o carregamento fracionado. Com as políticas de endpoint da VPC, o iniciador do carregamento fracionado não recebe automaticamente a permissão para executar a ação <code>s3:AbortMultipartUpload</code>.</p> <p>Além desses padrões, o proprietário do bucket pode permitir que outras entidades 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>.</p>
Listar partes	<p>Você deve ter permissão para realizar a ação <code>s3>ListMultipartUploadParts</code> para listar partes em um multipart upload.</p> <p>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 listar partes do multipart upload específico. Se o iniciador do carregamento fracionado</p>

Ação	Permissões obrigatórias
	<p>for um usuário do IAM, a Conta da AWS que controla o usuário do IAM também terá permissão para listar partes desse carregamento.</p> <p>Além desses padrões, o proprietário do bucket pode permitir que outras entidades principais executem a ação <code>s3>ListMultipartUploadParts</code> em um objeto. O proprietário do bucket também pode negar que qualquer principal realize a ação <code>s3>ListMultipartUploadParts</code>.</p>
Listar multipart uploads	<p>Você deve ter permissão para realizar a ação <code>s3>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>ListBucketMultipartUploads</code> no bucket.</p>
Permissões relacionadas a criptografia e descriptografia do AWS KMS	<p>Para fazer um carregamento fracionado com criptografia usando uma chave do AWS Key Management Service (AWS KMS), o solicitante deve ter permissão para as ações <code>kms:Decrypt</code> e <code>kms:GenerateDataKey</code> na chave. Essas permissões são necessárias porque o Amazon S3 precisa descriptografar e ler os dados de partes de arquivos criptografados antes de concluir o multipart upload.</p> <p>Para obter mais informações, consulte Carregando um arquivo grande para o Amazon S3 com criptografia usando uma AWS KMS key na Central de Conhecimento da AWS.</p> <p>Se o seu usuário ou a sua função do IAM estiver na mesma Conta da AWS que a chave do KMS, será necessário ter essas permissões na política da chave. Se o seu usuário ou a sua função do IAM pertencer a uma conta diferente da chave do KMS, será necessário ter as permissões na política da chave e no usuário ou na função do IAM.</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 \(p. 591\)](#). Para obter informações sobre usuários do IAM, acesse [Trabalhar com usuários e grupos](#).

Tópicos

- [Configurando uma política de ciclo de vida de bucket para anular multipart uploads incompletos \(p. 185\)](#)
- [Fazer upload de um objeto usando multipart upload \(p. 187\)](#)
- [Carregar um diretório usando a classe TransferUtility .NET de alto nível \(p. 200\)](#)
- [Listar multipart uploads \(p. 202\)](#)
- [Monitorar um multipart upload \(p. 204\)](#)
- [Abortar um multipart upload \(p. 206\)](#)
- [Copiar um objeto usando multipart upload \(p. 210\)](#)
- [Limites do multipart upload do Amazon S3 \(p. 214\)](#)

Configurando uma política de ciclo de vida de bucket para anular multipart uploads incompletos

Como melhor prática, recomendamos que você configure uma regra de ciclo de vida usando a ação `AbortIncompleteMultipartUpload` para minimizar os custos de armazenamento. Para obter mais informações sobre como anular um multipart upload, consulte [Abortar um multipart upload \(p. 206\)](#).

O Amazon S3 é compatível com uma regra de ciclo de vida de bucket que pode ser usada para fazer com que o Amazon S3 interrompa multipart uploads que não são concluídos dentro de um número especificado de dias após a inicialização. Quando um carregamento fracionado não é concluído dentro do prazo, ele se torna elegível para uma operação de anulação e o Simple Storage Service (Amazon S3) interrompe o carregamento fracionado (e exclui as partes a ele associadas).

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 do nome da chave do objeto](#)). Portanto, ela se aplica a todos os objetos no bucket para o qual você iniciou multipart uploads. Multipart uploads que foram iniciados e não concluídos dentro de sete dias tornam-se qualificados para uma operação de anulação. A ação de anulação não tem efeito em multipart uploads concluídos.

Para obter mais informações sobre a configuração do ciclo de vida de bucket, consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

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 executará nenhuma ação). Além disso, essa ação não se aplica a objetos. Nenhum objeto é excluído por essa ação do ciclo de vida.

O comando `put-bucket-lifecycle-configuration` da CLI adiciona a configuração de ciclo de vida do bucket especificado.

```
$ aws s3api put-bucket-lifecycle-configuration \
  --bucket bucketname \
  --lifecycle-configuration filename-containing-lifecycle-configuration
```

Para testar o comando da CLI, faça o seguinte:

- Configure o AWS CLI. Para obter instruções, consulte [Desenvolvimento com o Amazon S3 usando a AWS CLI \(p. 1257\)](#).
- 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",
```

```
        "Filter": {  
            "Prefix": ""  
        },  
        "AbortIncompleteMultipartUpload": {  
            "DaysAfterInitiation": 7  
        }  

```

3. Execute o comando da CLI a seguir para definir a configuração do ciclo de vida no seu bucket.

```
aws s3api put-bucket-lifecycle-configuration \  
--bucket bucketname \  
--lifecycle-configuration file://lifecycle.json
```

4. 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
```

Fazer upload de um objeto usando multipart upload

Você pode usar o multipart upload para fazer upload programático de um único objeto para o Amazon S3.

Para obter mais informações, consulte as seções a seguir.

Usar os AWS SDKs (API de alto nível)

O AWS SDK expõe uma API de alto nível, chamada de `TransferManager`, que simplifica os uploads fracionados. Para obter mais informações, consulte [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).

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 simultâneos que você quer usar quando fizer o upload das partes. Também é possível definir propriedades opcionais de objetos, a classe de armazenamento ou a lista de controle de acesso (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 no throughput.

Além da funcionalidade de upload de arquivos, a classe `TransferManager` possibilita a você parar um multipart upload em andamento. Um upload é considerado como em andamento depois que você o inicia até ser concluído ou parado. O `TransferManager` para todos os multipart uploads em andamento em um bucket especificado que foi iniciado antes de uma data e hora especificadas.

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 sobre multipart upload, incluindo a funcionalidade adicional oferecida por métodos API de nível baixo, consulte [Usar os AWS SDKs \(API de baixo nível\) \(p. 194\)](#).

Java

O exemplo a seguir carrega um objeto usando a API Java de alto nível de multipart upload (a classe `TransferManager`). Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.TransferManagerBuilder;
import com.amazonaws.services.s3.transfer.Upload;

import java.io.File;

public class HighLevelMultipartUpload {

    public static void main(String[] args) throws Exception {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Object key ***";
        String filePath = "*** Path for file to upload ***";

        try {
            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();
        }
    }
}
```

.NET

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, 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 de 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 for .NET e instruções para criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadFileMPUHighLevelAPITest
    {
        private const string bucketName = "*** provide bucket name ***";
        private const string keyName = "*** provide a name for the uploaded object ***";
        private const string filePath = "*** provide the full path name of the file to upload ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

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

        private static async Task UploadFileAsync()
        {
            try
            {
                var fileTransferUtility =
                    new TransferUtility(s3Client);

                // Option 1. Upload a file. The file name is used as the object key name.
                await fileTransferUtility.UploadAsync(filePath, bucketName);
                Console.WriteLine("Upload 1 completed");

                // Option 2. Specify object key name explicitly.
                await fileTransferUtility.UploadAsync(filePath, bucketName, keyName);
                Console.WriteLine("Upload 2 completed");

                // Option 3. Upload data from a type of System.IO.Stream.
                using (var fileToUpload =
                    new FileStream(filePath, FileMode.Open, FileAccess.Read))
                {
                    await fileTransferUtility.UploadAsync(fileToUpload,
                        bucketName, keyName);
                }
                Console.WriteLine("Upload 3 completed");
            }
        }
    }
}
```

```
// Option 4. Specify advanced settings.  
var fileTransferUtilityRequest = new TransferUtilityUploadRequest  
{  
    BucketName = bucketName,  
    FilePath = filePath,  
    StorageClass = S3StorageClass.StandardInfrequentAccess,  
    PartSize = 6291456, // 6 MB.  
    Key = keyName,  
    CannedACL = S3CannedACL.PublicRead  
};  
fileTransferUtilityRequest.Metadata.Add("param1", "Value1");  
fileTransferUtilityRequest.Metadata.Add("param2", "Value2");  
  
await fileTransferUtility.UploadAsync(fileTransferUtilityRequest);  
Console.WriteLine("Upload 4 completed");  
}  
catch (AmazonS3Exception e)  
{  
    Console.WriteLine("Error encountered on server. Message:{0} when  
writing an object", e.Message);  
}  
catch (Exception e)  
{  
    Console.WriteLine("Unknown encountered on server. Message:{0} when  
writing an object", e.Message);  
}  
  
}  
}  
}
```

PHP

Este tópico explica como usar a classe `Aws\S3\Model\MultipartUpload\UploadBuilder` de alto nível do AWS SDK for PHP para multipart uploads de arquivos. Pressupõe-se que você já esteja seguindo as instruções para [Usar o AWS SDK for PHP e executar exemplos do PHP \(p. 1268\)](#) e tenha o AWS SDK for 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 \(p. 1269\)](#).

```
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;  
}
```

Python

O exemplo a seguir carrega um objeto usando a API do Python de multipart upload de alto nível (a classe `TransferManager`).

```
import sys  
import threading  
  
import boto3  
from boto3.s3.transfer import TransferConfig  
  
MB = 1024 * 1024  
s3 = boto3.resource('s3')  
  
class TransferCallback:  
    """  
    Handle callbacks from the transfer manager.  
  
    The transfer manager periodically calls the __call__ method throughout  
    the upload and download process so that it can take action, such as  
    displaying progress to the user and collecting data about the transfer.  
    """  
  
    def __init__(self, target_size):  
        self._target_size = target_size  
        self._total_transferred = 0  
        self._lock = threading.Lock()  
        self.thread_info = {}  
  
    def __call__(self, bytes_transferred):  
        """  
        The callback method that is called by the transfer manager.  
  
        Display progress during file transfer and collect per-thread transfer  
        data. This method can be called by multiple threads, so shared instance  
        data is protected by a thread lock.  
        """  
        thread = threading.current_thread()  
        with self._lock:  
            self._total_transferred += bytes_transferred  
            if thread.ident not in self.thread_info.keys():  
                self.thread_info[thread.ident] = bytes_transferred  
            else:  
                self.thread_info[thread.ident] += bytes_transferred  
  
            target = self._target_size * MB  
            sys.stdout.write(  
                f"\r{self._total_transferred} of {target} transferred "  
                f"({(self._total_transferred / target) * 100:.2f}%)")  
            sys.stdout.flush()  
  
def upload_with_default_configuration(local_file_path, bucket_name,  
                                      object_key, file_size_mb):
```

```
"""
Upload a file from a local folder to an Amazon S3 bucket, using the default
configuration.
"""
transfer_callback = TransferCallback(file_size_mb)
s3.Bucket(bucket_name).upload_file(
    local_file_path,
    object_key,
    Callback=transfer_callback)
return transfer_callback.thread_info

def upload_with_chunksize_and_meta(local_file_path, bucket_name, object_key,
                                    file_size_mb, metadata=None):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart chunk size and adding metadata to the Amazon S3 object.

    The multipart chunk size controls the size of the chunks of data that are
    sent in the request. A smaller chunk size typically results in the transfer
    manager using more threads for the upload.

    The metadata is a set of key-value pairs that are stored with the object
    in Amazon S3.
    """
    transfer_callback = TransferCallback(file_size_mb)

    config = TransferConfig(multipart_chunksize=1 * MB)
    extra_args = {'Metadata': metadata} if metadata else None
    s3.Bucket(bucket_name).upload_file(
        local_file_path,
        object_key,
        Config=config,
        ExtraArgs=extra_args,
        Callback=transfer_callback)
    return transfer_callback.thread_info

def upload_with_high_threshold(local_file_path, bucket_name, object_key,
                               file_size_mb):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart threshold larger than the size of the file.

    Setting a multipart threshold larger than the size of the file results
    in the transfer manager sending the file as a standard upload instead of
    a multipart upload.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
    s3.Bucket(bucket_name).upload_file(
        local_file_path,
        object_key,
        Config=config,
        Callback=transfer_callback)
    return transfer_callback.thread_info

def upload_with_sse(local_file_path, bucket_name, object_key,
                   file_size_mb, sse_key=None):
    """
    Upload a file from a local folder to an Amazon S3 bucket, adding server-side
    encryption with customer-provided encryption keys to the object.

    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is

```

```
provided in the download request.  
"""  
transfer_callback = TransferCallback(file_size_mb)  
if sse_key:  
    extra_args = {  
        'SSECustomerAlgorithm': 'AES256',  
        'SSECustomerKey': sse_key}  
else:  
    extra_args = None  
s3.Bucket(bucket_name).upload_file(  
    local_file_path,  
    object_key,  
    ExtraArgs=extra_args,  
    Callback=transfer_callback)  
return transfer_callback.thread_info  
  
def download_with_default_configuration(bucket_name, object_key,  
                                         download_file_path, file_size_mb):  
    """  
    Download a file from an Amazon S3 bucket to a local folder, using the  
    default configuration.  
    """  
    transfer_callback = TransferCallback(file_size_mb)  
    s3.Bucket(bucket_name).Object(object_key).download_file(  
        download_file_path,  
        Callback=transfer_callback)  
    return transfer_callback.thread_info  
  
def download_with_single_thread(bucket_name, object_key,  
                                 download_file_path, file_size_mb):  
    """  
    Download a file from an Amazon S3 bucket to a local folder, using a  
    single thread.  
    """  
    transfer_callback = TransferCallback(file_size_mb)  
    config = TransferConfig(use_threads=False)  
    s3.Bucket(bucket_name).Object(object_key).download_file(  
        download_file_path,  
        Config=config,  
        Callback=transfer_callback)  
    return transfer_callback.thread_info  
  
def download_with_high_threshold(bucket_name, object_key,  
                                 download_file_path, file_size_mb):  
    """  
    Download a file from an Amazon S3 bucket to a local folder, setting a  
    multipart threshold larger than the size of the file.  
  
    Setting a multipart threshold larger than the size of the file results  
    in the transfer manager sending the file as a standard download instead  
    of a multipart download.  
    """  
    transfer_callback = TransferCallback(file_size_mb)  
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)  
    s3.Bucket(bucket_name).Object(object_key).download_file(  
        download_file_path,  
        Config=config,  
        Callback=transfer_callback)  
    return transfer_callback.thread_info  
  
def download_with_sse(bucket_name, object_key, download_file_path,  
                      file_size_mb, sse_key):
```

```
"""
Download a file from an Amazon S3 bucket to a local folder, adding a
customer-provided encryption key to the request.

When this kind of encryption is specified, Amazon S3 encrypts the object
at rest and allows downloads only when the expected encryption key is
provided in the download request.

"""

transfer_callback = TransferCallback(file_size_mb)

if sse_key:
    extra_args = {
        'SSECustomerAlgorithm': 'AES256',
        'SSECustomerKey': sse_key}
else:
    extra_args = None
s3.Bucket(bucket_name).Object(object_key).download_file(
    download_file_path,
    ExtraArgs=extra_args,
    Callback=transfer_callback)
return transfer_callback.thread_info
```

Usar os AWS SDKs (API de baixo nível)

O AWS SDK expõe uma API de baixo nível que se assemelha à API REST do Amazon S3 para carregamentos fracionados (consulte [Carregar e copiar objetos usando multipart upload \(p. 180\)](#)). Use a API de baixo nível quando precisar pausar e retomar carregamentos fracionados, variar os tamanhos das partes durante o carregamento ou não souber o tamanho dos dados com antecedência. Quando esses requisitos não existirem, use a API de alto nível (consulte [Usar os AWS SDKs \(API de alto nível\) \(p. 187\)](#)).

Java

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 como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
```

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

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

public class LowLevelMultipartUpload {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ***";
        String filePath = "*** Path to file to upload ***";

        File file = new File(filePath);
        long contentLength = file.length();
        long partSize = 5 * 1024 * 1024; // Set part size to 5 MB.

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

            // Create a list of ETag objects. You retrieve ETags for each object part
            uploaded,
            // then, after each individual part has been uploaded, pass the list of
            ETags to
            // the request to complete the upload.
            List<PartETag> partETags = new ArrayList<PartETag>();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
            InitiateMultipartUploadRequest(bucketName, keyName);
            InitiateMultipartUploadResult initResponse =
            s3Client.initiateMultipartUpload(initRequest);

            // Upload the file parts.
            long filePosition = 0;
            for (int i = 1; filePosition < contentLength; i++) {
                // Because the last part could be less than 5 MB, adjust the part size
                as needed.
                partSize = Math.min(partSize, (contentLength - filePosition));

                // Create the request to upload a part.
                UploadPartRequest uploadRequest = new UploadPartRequest()
                    .withBucketName(bucketName)
                    .withKey(keyName)
                    .withUploadId(initResponse.getUploadId())
                    .withPartNumber(i)
                    .withFileOffset(filePosition)
                    .withFile(file)
                    .withPartSize(partSize);

                // Upload the part and add the response's ETag to our list.
                UploadPartResult uploadResult = s3Client.uploadPart(uploadRequest);
                partETags.add(uploadResult.getPartETag());

                filePosition += partSize;
            }

            // Complete the multipart upload.
        }
    }
}
```

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

.NET

O exemplo do C# a seguir mostra como usar a API de multipart upload do AWS SDK for .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 [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).

Note

Ao usar a AWS SDK for .NET API para fazer upload de objetos grandes, um tempo limite pode ocorrer 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 for .NET e instruções para criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

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

namespace Amazon.DocSamples.S3
{
    class UploadFileMPULowLevelAPITest
    {
        private const string bucketName = "*** provide bucket name ***";
        private const string keyName = "*** provide a name for the uploaded object
***";
        private const string filePath = "*** provide the full path name of the file to
upload ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            Console.WriteLine("Uploading an object");
            UploadObjectAsync().Wait();
        }
    }
}
```

```
private static async Task UploadObjectAsync()
{
    // Create list to store upload part responses.
    List<UploadPartResponse> uploadResponses = new List<UploadPartResponse>();

    // Setup information required to initiate the multipart upload.
    InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
    {
        BucketName = bucketName,
        Key = keyName
    };

    // Initiate the upload.
    InitiateMultipartUploadResponse initResponse =
        await s3Client.InitiateMultipartUploadAsync(initiateRequest);

    // Upload parts.
    long contentLength = new FileInfo(filePath).Length;
    long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB

    try
    {
        Console.WriteLine("Uploading parts");

        long filePosition = 0;
        for (int i = 1; filePosition < contentLength; i++)
        {
            UploadPartRequest uploadRequest = new UploadPartRequest
            {
                BucketName = bucketName,
                Key = keyName,
                UploadId = initResponse.UploadId,
                PartNumber = i,
                PartSize = partSize,
                FilePosition = filePosition,
                FilePath = filePath
            };

            // Track upload progress.
            uploadRequest.StreamTransferProgress +=
                new
EventHandler<StreamTransferProgressArgs>(UploadPartProgressEventCallback);

            // Upload a part and add the response to our list.
            uploadResponses.Add(await s3Client.UploadPartAsync(uploadRequest));

            filePosition += partSize;
        }

        // Setup to complete the upload.
        CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
        {
            BucketName = bucketName,
            Key = keyName,
            UploadId = initResponse.UploadId
        };
        completeRequest.AddPartETags(uploadResponses);

        // Complete the upload.
        CompleteMultipartUploadResponse completeUploadResponse =
            await s3Client.CompleteMultipartUploadAsync(completeRequest);
    }
    catch (Exception exception)
```

```
        {
            Console.WriteLine("An AmazonS3Exception was thrown: { 0}",
exception.Message);

            // Abort the upload.
            AbortMultipartUploadRequest abortMPURequest = new
AbortMultipartUploadRequest
            {
                BucketName = bucketName,
                Key = keyName,
                UploadId = initResponse.UploadId
            };
            await s3Client.AbortMultipartUploadAsync(abortMPURequest);
        }
    }
    public static void UploadPartProgressEventCallback(object sender,
StreamTransferProgressArgs e)
{
    // Process event.
    Console.WriteLine("{0}/{1}", e.TransferredBytes, e.TotalBytes);
}
}
}
```

PHP

Este tópico mostra como usar o método de baixo nível `uploadPart`, da versão 3 do AWS SDK for PHP, para carregar um arquivo em várias partes. Pressupõe-se que você já esteja seguindo as instruções para [Usar o AWS SDK for PHP e executar exemplos do PHP \(p. 1268\)](#) e tenha o AWS SDK for PHP devidamente instalado.

O exemplo de PHP a seguir faz upload de um arquivo para um bucket do Amazon S3 usando o multipart upload da API de baixo nível do PHP. Para obter informações sobre a execução dos exemplos de PHP neste guia, consulte [Executar exemplos do PHP \(p. 1269\)](#).

```
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',
    '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;
```

Uso do AWS SDK for Ruby

O AWS SDK for Ruby versão 3 é compatível com carregamentos fracionados do Amazon S3 de duas formas. Para a primeira opção, é possível usar uploads de arquivos gerenciados. Para obter mais informações, consulte [Uploading Files to Amazon S3](#) no Blog do desenvolvedor da AWS. O upload de arquivos gerenciado é o método recomendado para fazer upload de arquivos em um bucket. Eles fornecem os seguintes benefícios:

- Gerenciam multipart uploads para objetos com mais de 15 MB.
- Abrem corretamente arquivos em modo binário para evitar problemas de codificação.
- Usam vários threads para upload de partes de grandes objetos em paralelo.

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 obtidas por upload anteriormente.
- [abort_multipart_upload](#): interrompe um multipart upload.

Para obter mais informações, consulte [Usar o AWS SDK for Ruby - versão 3 \(p. 1269\)](#).

Uso dos REST API

As seções a seguir na Referência de APIs do Amazon Simple Storage Service descrevem a API REST para multipart upload.

- [Iniciar multipart upload](#)
- [Carregar parte](#)
- [Concluir carregamento fracionado](#)
- [Parar o multipart upload](#)
- [Listar partes](#)
- [Listar carregamentos fracionados](#)

Uso do AWS CLI

As seções a seguir na AWS Command Line Interface (AWS CLI) descrevem as operações para carregamento fracionado.

- [Iniciar multipart upload](#)
- [Carregar parte](#)
- [Carregar parte \(copiar\)](#)
- [Concluir carregamento fracionado](#)
- [Anular carregamento fracionado](#)
- [Listar partes](#)
- [Listar carregamentos fracionados](#)

Também é possível usar essa API REST para fazer suas solicitações REST ou usar um dos AWS SDKs. Para obter mais informações sobre a API REST, consulte [Uso dos REST API \(p. 200\)](#). Para obter mais informações sobre os SDKs, consulte [Fazer upload de um objeto usando multipart upload \(p. 187\)](#).

Carregar um diretório usando a classe TransferUtility .NET de alto nível

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 da chave para os objetos resultantes. O Amazon S3 cria os nomes de chave usando o caminho original do arquivo. 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
```

quando você faz upload desse diretório, o Amazon S3 usa os seguintes nomes de chave:

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 do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadDirMPUHighLevelAPITest
    {
        private const string existingBucketName = "*** bucket name ***";
        private const string directoryPath = @"/*** directory path ***";
        // The example uploads only .txt files.
        private const string wildCard = "*.*txt";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            UploadDirAsync().Wait();
        }

        private static async Task UploadDirAsync()
        {
            try
            {
                var directoryTransferUtility =
                    new TransferUtility(s3Client);

                // 1. Upload a directory.
                await directoryTransferUtility.UploadDirectoryAsync(directoryPath,
                    existingBucketName);
                Console.WriteLine("Upload statement 1 completed");

                // 2. Upload only the .txt files from a directory
                //     and search recursively.
                await directoryTransferUtility.UploadDirectoryAsync(
                    directoryPath,
                    existingBucketName,
                    wildCard,
                    SearchOption.AllDirectories);
                Console.WriteLine("Upload statement 2 completed");

                // 3. The same as Step 2 and some optional configuration.
                //     Search recursively for .txt files to upload.
                var request = new TransferUtilityUploadDirectoryRequest
```

```
        {
            BucketName = existingBucketName,
            Directory = directoryPath,
            SearchOption = SearchOption.AllDirectories,
            SearchPattern = wildCard
        };

        await directoryTransferUtility.UploadDirectoryAsync(request);
        Console.WriteLine("Upload statement 3 completed");
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine(
            "Error encountered ***. Message:'{0}' when writing an object",
            e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine(
            "Unknown encountered on server. Message:'{0}' when writing an object",
            e.Message);
    }
}
}
```

Listar multipart uploads

Você pode usar os AWS SDKs (API de baixo nível) para recuperar uma lista de uploads fracionados em andamento no Amazon S3.

Listar uploads fracionados usando o AWS SDK (API de baixo nível)

Java

As tarefas a seguir fornecem orientações para usar as classes Java de baixo nível para listar todos os multipart uploads em andamento em um bucket.

Processo de listagem de multipart uploads da API de baixo nível

1	Crie uma instância da classe <code>ListMultipartUploadsRequest</code> e forneça o nome do bucket.
2	Execute o método <code>AmazonS3Client.listMultipartUploads</code> . O método retorna uma instância da classe <code>MultipartUploadListing</code> que fornece informações sobre os multipart uploads em andamento.

O exemplo de código Java a seguir demonstra as tarefas anteriores.

Example

```
ListMultipartUploadsRequest allMultipartUploadsRequest =
    new ListMultipartUploadsRequest(existingBucketName);
MultipartUploadListing multipartUploadListing =
    s3Client.listMultipartUploads(allMultipartUploadsRequest);
```

.NET

Para listar todos os multipart uploads em andamento em um bucket específico, use a classe `ListMultipartUploadsRequest` da API de multipart upload do AWS SDK for .NET de nível

baixo. O `AmazonS3Client.ListMultipartUploads` método retorna uma instância da classe `ListMultipartUploadsResponse` que fornece informações sobre multipart uploads em andamento.

Um 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 parado. Para obter mais informações sobre multipart uploads do Amazon S3, consulte [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).

O exemplo do C# a seguir mostra como usar o AWS SDK for .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 for .NET e instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
ListMultipartUploadsRequest request = new ListMultipartUploadsRequest
{
    BucketName = bucketName // Bucket receiving the uploads.
};

ListMultipartUploadsResponse response = await
    AmazonS3Client.ListMultipartUploadsAsync(request);
```

PHP

Este tópico mostra como usar as classes da API de baixo nível da versão 3 do AWS SDK for 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 for PHP e executar exemplos do PHP \(p. 1268\)](#) e tenha o AWS SDK for PHP devidamente instalado.

O exemplo de PHP a seguir demonstra a listagem de todos os multipart uploads em andamento em um bucket.

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

use Aws\S3\S3Client;

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

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

// Retrieve a list of the current multipart uploads.
$result = $s3->listMultipartUploads([
    'Bucket' => $bucket
]);

// Write the list of uploads to the page.
print_r($result->toArray());
```

Listar multipart uploads usando a API REST

As seções a seguir na Referência de APIs do Amazon Simple Storage Service descrevem a API REST para listagem de multipart uploads:

- [ListParts](#)-lista as partes enviadas por upload de um multipart upload específico.
- [ListMultipartUploads](#) lista multipart uploads em andamento.

Listagem de carregamentos fracionados usando a AWS CLI

As seções a seguir na AWS Command Line Interface descrevem as operações de listagem de carregamentos fracionados.

- [list-parts](#): liste as partes enviadas por upload de um multipart upload específico.
- [list-multipart-uploads](#): liste multipart uploads em andamento.

Monitorar um multipart upload

A API multipart upload de alto nível upload fornece uma interface de escuta, `ProgressListener`, para acompanhar o progresso ao fazer upload de um objeto para o Amazon S3. Os eventos de progresso ocorrem periodicamente e notificam o listener que os bytes foram transferidos.

Java

Example

```
TransferManager tm = new TransferManager(new ProfileCredentialsProvider());

PutObjectRequest request = new PutObjectRequest(
    existingBucketName, keyName, new File(filePath));

// Subscribe to the event and provide event handler.
request.setProgressListener(new ProgressListener() {
    public void progressChanged(ProgressEvent event) {
        System.out.println("Transferred bytes: " +
            event.getBytesTransferred());
    }
});
```

Example

O código Java a seguir faz upload de um arquivo e usa `ProgressListener` para rastrear o progresso do upload. Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
import java.io.File;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.event.ProgressEvent;
import com.amazonaws.event.ProgressListener;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.Upload;

public class TrackMPUProgressUsingHighLevelAPI {

    public static void main(String[] args) throws Exception {
        String existingBucketName = "*** Provide bucket name ***";
        String keyName           = "*** Provide object key ***";
        String filePath          = "*** file to upload ***";

        TransferManager tm = new TransferManager(new ProfileCredentialsProvider());

        // For more advanced uploads, you can create a request object
        // and supply additional request parameters (ex: progress listeners,
```

```
// canned ACLs, etc.)
PutObjectRequest request = new PutObjectRequest(
    existingBucketName, keyName, new File(filePath));

// You can ask the upload for its progress, or you can
// add a ProgressListener to your request to receive notifications
// when bytes are transferred.
request.setGeneralProgressListener(new ProgressListener() {
@Override
public void progressChanged(ProgressEvent progressEvent) {
    System.out.println("Transferred bytes: " +
        progressEvent.getBytesTransferred());
}
});

// TransferManager processes all transfers asynchronously,
// so this call will return immediately.
Upload upload = tm.upload(request);

try {
    // You can block and wait for the upload to finish
    upload.waitForCompletion();
} catch (AmazonClientException amazonClientException) {
    System.out.println("Unable to upload file, upload aborted.");
    amazonClientException.printStackTrace();
}
}
```

.NET

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 for .NET e instruções para criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class TrackMPUUsingHighLevelAPITest
    {
        private const string bucketName = "*** provide the bucket name ***";
        private const string keyName = "*** provide the name for the uploaded object
***";
        private const string filePath = " *** provide the full path name of the file to
upload ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

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

```
private static async Task TrackMPUAsync()
{
    try
    {
        var fileTransferUtility = new TransferUtility(s3Client);

        // Use TransferUtilityUploadRequest to configure options.
        // In this example we subscribe to an event.
        var uploadRequest =
            new TransferUtilityUploadRequest
            {
                BucketName = bucketName,
                FilePath = filePath,
                Key = keyName
            };

        uploadRequest.UploadProgressEvent +=
            new EventHandler<UploadProgressArgs>
            (uploadRequest_UploadPartProgressEvent);

        await fileTransferUtility.UploadAsync(uploadRequest);
        Console.WriteLine("Upload completed");
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
}

static void uploadRequest_UploadPartProgressEvent(object sender,
UploadProgressArgs e)
{
    // Process event.
    Console.WriteLine("{0}/{1}", e.TransferredBytes, e.TotalBytes);
}
```

Abortar um multipart upload

Após iniciar um multipart upload, você começa a fazer upload de partes. O Amazon S3 armazena essas partes, mas cria o objeto a partir das partes somente após o upload de todas e envia uma solicitação `successful` para concluir o multipart upload (você deve verificar se a 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.

Você é cobrado por todo o armazenamento associado a partes enviadas por upload. Para obter mais informações, consulte [Multipart upload e definição de preço \(p. 182\)](#). Portanto, é importante que você conclua o multipart upload para ter o objeto criado ou pare o multipart upload para remover todas as partes enviadas por upload.

Você pode interromper um carregamento fracionado em andamento no Amazon S3 usando a AWS Command Line Interface (AWS CLI), a API REST ou os AWS SDKs. Você também pode interromper um multipart upload incompleto usando uma política de ciclo de vida de bucket.

Usar os AWS SDKs (API de alto nível)

Java

A classe `TransferManager` fornece o método `abortMultipartUploads` para interromper multipart uploads em andamento. Um upload é considerado como em andamento depois que você o inicia até ser concluído ou parado. Você fornece um valor `Date` e essa API interrompe todos os multipart uploads, naquele bucket, que foram iniciados antes da `Date` especificada e que ainda estão em andamento.

As tarefas a seguir orientam sobre a utilização de classes Java de alto nível para interromper multipart uploads.

Processo de interrupção de multipart uploads com API de alto nível

1	Crie uma instância da classe <code>TransferManager</code> .
2	Execute o método <code>TransferManager.abortMultipartUploads</code> passando o nome do bucket e um valor de <code>Date</code> .

O código Java a seguir interrompe todos os multipart uploads em andamento que foram iniciados em um bucket específico uma semana atrás. Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
import java.util.Date;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.transfer.TransferManager;

public class AbortMPUUsingHighLevelAPI {

    public static void main(String[] args) throws Exception {
        String existingBucketName = "**** Provide existing bucket name ****";

        TransferManager tm = new TransferManager(new ProfileCredentialsProvider());

        int sevenDays = 1000 * 60 * 60 * 24 * 7;
        Date oneWeekAgo = new Date(System.currentTimeMillis() - sevenDays);

        try {
            tm.abortMultipartUploads(existingBucketName, oneWeekAgo);
        } catch (AmazonClientException amazonClientException) {
            System.out.println("Unable to upload file, upload was aborted.");
            amazonClientException.printStackTrace();
        }
    }
}
```

Note

Você também pode parar um multipart upload específico. Para obter mais informações, consulte [Usar os AWS SDKs \(API de baixo nível\) \(p. 208\)](#).

.NET

O exemplo do C# a seguir interrompe 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 for .NET e instruções para criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.Threading.Tasks;

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

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

        private static async Task AbortMPUAsync()
        {
            try
            {
                var transferUtility = new TransferUtility(s3Client);

                // Abort all in-progress uploads initiated before the specified date.
                await transferUtility.AbortMultipartUploadsAsync(
                    bucketName, DateTime.Now.AddDays(-7));
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }
    }
}
```

Note

Você também pode parar um multipart upload específico. Para obter mais informações, consulte [Usar os AWS SDKs \(API de baixo nível\) \(p. 208\)](#).

Usar os AWS SDKs (API de baixo nível)

É possível interromper um multipart upload em andamento chamando o método `AmazonS3.abortMultipartUpload`. Esse método exclui todas as partes que foram carregadas no Amazon S3 e libera os recursos. Você deve fornecer o ID do upload, o nome do bucket e o nome da chave. O código Java de exemplo a seguir demonstra como interromper um multipart upload em andamento.

Para parar um multipart upload, é preciso fornecer o ID de upload e nomes do bucket e da chave usados no upload. Após parar 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 [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).

Java

O exemplo de código Java a seguir interrompe um multipart upload em andamento.

Example

```
InitiateMultipartUploadRequest initRequest =
    new InitiateMultipartUploadRequest(existingBucketName, keyName);
InitiateMultipartUploadResult initResponse =
    s3Client.initiateMultipartUpload(initRequest);

AmazonS3 s3Client = new AmazonS3Client(new ProfileCredentialsProvider());
s3Client.abortMultipartUpload(new AbortMultipartUploadRequest(
    existingBucketName, keyName, initResponse.getUploadId()));
```

Note

Em vez de um multipart upload específico, você pode interromper todos os multipart iniciados antes de uma determinada hora que ainda estiverem em andamento. Essa operação de limpeza é útil para interromper multipart uploads antigos que você iniciou, mas não concluiu ou parou. Para obter mais informações, consulte [Usar os AWS SDKs \(API de alto nível\) \(p. 207\)](#).

.NET

O exemplo do C# a seguir mostra como parar um multipart upload. Para um exemplo do C# completo que inclui o código seguinte, consulte [Usar os AWS SDKs \(API de baixo nível\) \(p. 194\)](#).

```
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 [Usar os AWS SDKs \(API de alto nível\) \(p. 207\)](#).

PHP

Este exemplo descreve como usar uma classe da versão 3 do AWS SDK for PHP para anular um carregamento fracionado que está em andamento. Pressupõe-se que você já esteja seguindo as instruções para [Usar o AWS SDK for PHP e executar exemplos do PHP \(p. 1268\)](#) e tenha o AWS SDK for PHP devidamente instalado. O exemplo do método `abortMultipartUpload()`.

Para obter informações sobre a execução dos exemplos de PHP neste guia, consulte [Executar exemplos do PHP \(p. 1269\)](#).

```
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,
]);
```

Uso dos REST API

Para obter mais informações sobre o uso da API REST para interromper um multipart upload, consulte [AbortMultipartUpload](#) na Referência da API do Amazon Simple Storage Service.

Uso do AWS CLI

Para obter mais informações sobre o uso da AWS CLI para interromper um upload fracionado, consulte [abort-multipart-upload](#) na Referência de comandos da AWS CLI.

Copiar um objeto usando 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 \(p. 215\)](#).

Uso de AWS SDKs

Para copiar um objeto usando a API de baixo nível, faça o seguinte:

- Inicie um multipart upload chamando 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.

Java

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 como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
```

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

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

public class LowLevelMultipartCopy {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String sourceBucketName = "*** Source bucket name ***";
        String sourceObjectKey = "*** Source object key ***";
        String destBucketName = "*** Target bucket name ***";
        String destObjectKey = "*** Target object key ***";

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

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
InitiateMultipartUploadRequest(destBucketName, destObjectKey);
            InitiateMultipartUploadResult initResult =
s3Client.initiateMultipartUpload(initRequest);

            // Get the object size to track the end of the copy operation.
            GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest(sourceBucketName, sourceObjectKey);
            ObjectMetadata metadataResult =
s3Client.getObjectMetadata(metadataRequest);
            long objectSize = metadataResult.getContentLength();

            // Copy the object using 5 MB parts.
            long partSize = 5 * 1024 * 1024;
            long bytePosition = 0;
            int partNum = 1;
            List<CopyPartResult> copyResponses = new ArrayList<CopyPartResult>();
            while (bytePosition < objectSize) {
                // The last part might be smaller than partSize, so check to make sure
                // that lastByte isn't beyond the end of the object.
                long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);

                // Copy this part.
                CopyPartRequest copyRequest = new CopyPartRequest()
                    .withSourceBucketName(sourceBucketName)
                    .withSourceKey(sourceObjectKey)
                    .withDestinationBucketName(destBucketName)
                    .withDestinationKey(destObjectKey)
                    .withUploadId(initResult.getUploadId())
                    .withFirstByte(bytePosition)
                    .withLastByte(lastByte)
                    .withPartNumber(partNum++);
                copyResponses.add(s3Client.copyPart(copyRequest));
                bytePosition += partSize;
            }

            // Complete the upload request to concatenate all uploaded parts and make
            // the copied object available.
            CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest(
```

```
        destBucketName,
        destObjectKey,
        initResult.getUploadId(),
        getETags(copyResponses));
    s3Client.completeMultipartUpload(completeRequest);
    System.out.println("Multipart copy complete.");
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}

// This is a helper function to construct a list of ETags.
private static List<PartETag> getETags(List<CopyPartResult> responses) {
    List<PartETag> etags = new ArrayList<PartETag>();
    for (CopyPartResult response : responses) {
        etags.add(new PartETag(response.getPartNumber(), response.getETag()));
    }
    return etags;
}
}
```

.NET

O exemplo em C# a seguir mostra como usar o AWS SDK for .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 [Uso de AWS SDKs \(p. 218\)](#). Para obter mais informações sobre multipart uploads do Amazon S3, consulte [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).

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 carregamento fracionado do AWS SDK for .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 do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CopyObjectUsingMPUapiTest
    {
        private const string sourceBucket = "**** provide the name of the bucket with
source object ***";
        private const string targetBucket = "**** provide the name of the bucket to copy
the object to ***";
        private const string sourceObjectKey = "**** provide the name of object to copy
***";
        private const string targetObjectKey = "**** provide the name of the object copy
***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
```

```
public static void Main()
{
    s3Client = new AmazonS3Client(bucketRegion);
    Console.WriteLine("Copying an object");
    MPUCopyObjectAsync().Wait();
}
private static async Task MPUCopyObjectAsync()
{
    // Create a list to store the upload part responses.
    List<UploadPartResponse> uploadResponses = new List<UploadPartResponse>();
    List<CopyPartResponse> copyResponses = new List<CopyPartResponse>();

    // Setup information required to initiate the multipart upload.
    InitiateMultipartUploadRequest initiateRequest =
        new InitiateMultipartUploadRequest
    {
        BucketName = targetBucket,
        Key = targetObjectKey
    };

    // Initiate the upload.
    InitiateMultipartUploadResponse initResponse =
        await s3Client.InitiateMultipartUploadAsync(initiateRequest);

    // Save the upload ID.
    String uploadId = initResponse.UploadId;

    try
    {
        // Get the size of the object.
        GetObjectMetadataRequest metadataRequest = new GetObjectMetadataRequest
        {
            BucketName = sourceBucket,
            Key = sourceObjectKey
        };

        GetObjectMetadataResponse metadataResponse =
            await s3Client.GetObjectMetadataAsync(metadataRequest);
        long objectSize = metadataResponse.ContentLength; // Length in bytes.

        // Copy the parts.
        long partSize = 5 * (long)Math.Pow(2, 20); // Part size is 5 MB.

        long bytePosition = 0;
        for (int i = 1; bytePosition < objectSize; i++)
        {
            CopyPartRequest copyRequest = new CopyPartRequest
            {
                DestinationBucket = targetBucket,
                DestinationKey = targetObjectKey,
                SourceBucket = sourceBucket,
                SourceKey = sourceObjectKey,
                UploadId = uploadId,
                FirstByte = bytePosition,
                LastByte = bytePosition + partSize - 1 >= objectSize ?
objectSize - 1 : bytePosition + partSize - 1,
                PartNumber = i
            };

            copyResponses.Add(await s3Client.CopyPartAsync(copyRequest));

            bytePosition += partSize;
        }

        // Set up to complete the copy.
    }
}
```

```
        CompleteMultipartUploadRequest completeRequest =
        new CompleteMultipartUploadRequest
        {
            BucketName = targetBucket,
            Key = targetObjectKey,
            UploadId = initResponse.UploadId
        };
        completeRequest.AddPartETags(copyResponses);

        // Complete the copy.
        CompleteMultipartUploadResponse completeUploadResponse =
            await s3Client.CompleteMultipartUploadAsync(completeRequest);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
}
}
```

Uso dos REST API

As seções a seguir na Referência de APIs do Amazon Simple Storage Service 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](#)
- [Carregar parte](#)
- [Carregar parte \(copiar\)](#)
- [Concluir carregamento fracionado](#)
- [Anular carregamento fracionado](#)
- [Listar partes](#)
- [Listar carregamentos fracionados](#)

Use essas APIs para fazer suas solicitações REST ou use um dos SDKs que fornecemos. Para obter mais informações sobre como usar o multipart upload com a AWS CLI, consulte [Uso do AWS CLI \(p. 200\)](#). Para obter mais informações sobre os SDKs, consulte [Suporte do AWS SDK para upload fracionado \(p. 183\)](#).

Limites do multipart upload do Amazon S3

A tabela a seguir fornece especificações básicas do multipart upload. Para obter mais informações, consulte [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).

Item	Especificação
Tamanho máximo do objeto	5 TiB

Item	Especificação
Número máximo de partes por upload	10.000
Números de parte	1 a 10.000 (inclusive)
Tamanho da parte	5 MiB a 5 GiB. Não há limite de tamanho mínimo na última parte do carregamento fracionado.
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

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 seu objeto de até 5 GB em uma única operação atômica. Contudo, para copiar um objeto maior 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 entre locais do Amazon S3 (por exemplo, us-west-1 e Europa)
- 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, defina o mesmo objeto como origem e destino.

Cada objeto tem metadados. Alguns deles são metadados de sistema e outros são definidos pelo usuário. Você pode controlar alguns dos metadados de sistema, como a configuração de classe de armazenamento para usar o objeto e configurar a criptografia no 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 for configurado para usar o armazenamento padrão do S3, você poderá optar por usar o S3 Intelligent-Tiering 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 pelo 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 [Trabalhar com metadados de objeto \(p. 166\)](#).

Note

- Cópia de objetos por locais incorre em alterações de banda larga.
- Se o objeto de origem estiver arquivado em S3 Glacier Flexible Retrieval ou em S3 Glacier Deep Archive, será necessário primeiramente restaurar uma cópia temporária antes de copiar o objeto para outro bucket. Para obter informações sobre objetos de arquivo, consulte [Transição para as classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive \(arquivamento de objetos\)](#) (p. 750).

Ao copiar objetos, é possível solicitar que o Amazon S3 salve o objeto de destino criptografado com uma AWS KMS key, 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 no lado do servidor com chave fornecida pelo cliente, você deverá fornecer informações de criptografia na solicitação, de maneira que o Amazon S3 possa descriptografar o objeto para a cópia. Para obter mais informações, consulte [Proteção de dados usando criptografia](#) (p. 358).

Ao copiar objetos, você pode optar por usar um algoritmo de soma de verificação diferente para eles. Se optar por usar o mesmo algoritmo ou um novo, o Amazon S3 calculará um novo valor de soma de verificação depois que o objeto for copiado. O Amazon S3 não copia diretamente o valor da soma de verificação. O valor da soma de verificação de objetos que foram carregados usando carregamentos fracionados pode ser alterado. Para obter mais informações sobre como essa soma de verificação é calculada, consulte [Usar somas de verificação de nível parcial para carregamentos fracionados](#) (p. 236).

Para copiar mais de um objeto do Amazon S3 com uma única solicitação, é possível usar operações em lote do Amazon S3. Você fornece às operações em lote do S3 uma lista de objetos nos quais operar. O S3 Batch Operations chama a respectiva API para executar a operação especificada. Um único trabalho de operações em lote pode realizar a operação especificada em bilhões de objetos contendo exabytes de dados.

O recurso S3 Batch Operations rastreia o progresso, envia notificações e armazena um relatório de conclusão detalhado de todas as ações, fornecendo uma experiência totalmente gerenciada, auditável e sem servidor. Use as operações em lote do S3 por meio do AWS Management Console, da AWS CLI, dos AWS SDKs ou da API REST. Para obter mais informações, consulte [the section called “Conceitos básicos do Batch Operations”](#) (p. 938).

Para copiar um objeto

Para copiar um objeto, use os métodos a seguir.

Uso do console do S3

No console do Amazon S3, você pode copiar ou mover um objeto. Para obter mais informações, consulte os procedimentos a seguir.

Note

Objetos criptografados com chaves de criptografia fornecidas pelo cliente (SSE-C) não podem ser copiados usando o console do S3. Para copiar ou mover objetos criptografados com SSE-C, use a AWS CLI, o AWS SDK ou a API REST do Amazon S3.

Para copiar um objeto

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

2. Navegue até o bucket ou pasta do Amazon S3 que contém os objetos que você deseja copiar.
3. Marque a caixa de seleção à esquerda dos nomes dos objetos que você deseja copiar.
4. Escolha Actions (Ações) e escolha Copy (Copiar) na lista de opções exibida.

Você também pode escolher Copy (Copiar) nas opções no canto superior direito.

5. Selecione o tipo de destino e a conta de destino. Para especificar o caminho de destino, escolha Browse S3 (Procurar S3), navegue até o destino e marque a caixa de seleção à esquerda do destino. Escolha Choose destination (Escolher destino) no canto inferior direito.

Outra alternativa é inserir o caminho de destino.

6. Se você não tiver o versionamento de bucket habilitado, pode ser solicitado que você confirme que objetos existentes com o mesmo nome serão substituídos. Se estiver tudo certo, marque a caixa de seleção e prossiga. Se você quiser manter todas as versões de objetos neste bucket, selecione Enable Bucket Versioning (Ativar versionamento de bucket). Você também pode atualizar as propriedades de criptografia padrão e de bloqueio de objetos do S3.
7. Em Additional checksums (Somas de verificação adicionais), escolha se deseja copiar os objetos usando a função de soma de verificação existente ou substituí-la por uma nova. Ao carregar os objetos, você teve a opção de especificar o algoritmo de soma de verificação usado para verificar a integridade dos dados. Ao copiar o objeto, você tem a opção de escolher uma nova função. Se você não tiver especificado originalmente uma soma de verificação adicional, poderá usar essa seção de opções de cópia para adicionar uma.

Note

Mesmo se optar por usar a mesma função de soma de verificação, o valor dela poderá mudar se você copiar o objeto e ele tiver mais de 16 MB. O valor da soma de verificação pode mudar devido à forma como as somas de verificação são calculadas para carregamentos fracionados. Para obter mais informações sobre como a soma de verificação pode mudar ao copiar esse objeto, consulte [Usar somas de verificação de nível parcial para carregamentos fracionados \(p. 236\)](#).

Para alterar a função de soma de verificação, escolha Replace with a new checksum function (Substituir por uma nova função de soma de verificação). Escolha a nova função de soma de verificação na caixa. Quando o objeto é copiado, a nova soma de verificação é calculada e armazenada usando o algoritmo especificado.

8. Escolha Copy (Copiar) no canto inferior direito. O Amazon S3 copia o objeto no destino.

Como mover objetos

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Navegue até o bucket ou pasta do Amazon S3 que contém os objetos que você deseja mover.
3. Marque a caixa de seleção à esquerda dos nomes dos objetos que você deseja mover.
4. Escolha Actions (Ações) e escolha Move (Mover) na lista de opções exibida.

Você também pode escolher Move (Mover) nas opções no canto superior direito.

5. Para especificar o caminho de destino, escolha Browse S3 (Procurar S3), navegue até o destino e marque a caixa de seleção à esquerda do destino. Escolha Choose destination (Escolher destino) no canto inferior direito.

Outra alternativa é inserir o caminho de destino.

6. Se você não tiver o versionamento de bucket habilitado, pode ser solicitado que você confirme que objetos existentes com o mesmo nome serão substituídos. Se estiver tudo certo, marque a caixa de seleção e prossiga. Se você quiser manter todas as versões de objetos neste bucket,

selecione Enable Bucket Versioning (Ativar versionamento de bucket). Você também pode atualizar as propriedades de criptografia padrão e de bloqueio de objetos.

7. Selecione Move (Mover) no canto inferior direito. O Amazon S3 move seus objetos para o destino.

Note

- Essa ação cria uma cópia de todos os objetos especificados com configurações atualizadas, atualiza a data da última modificação no local especificado e adiciona um marcador de exclusão ao objeto original.
- Ao mover pastas, aguarde a conclusão da ação de movimentação antes de fazer alterações adicionais nas pastas.
- Essa ação atualiza metadados para versionamento de bucket, criptografia, recursos de bloqueio de objetos e objetos arquivados.

Uso de AWS SDKs

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 um objeto usando multipart upload \(p. 210\)](#).

Java

Example

O exemplo a seguir copia um objeto no Amazon S3 usando o AWS SDK for Java. Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

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

import java.io.IOException;

public class CopyObjectSingleOperation {

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

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

            // Copy the object into a new object in the same bucket.
            CopyObjectRequest copyObjRequest = new CopyObjectRequest(bucketName,
                sourceKey, bucketName, destinationKey);
            s3Client.copyObject(copyObjRequest);
        } catch (AmazonServiceException e) {
```

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

.NET

O exemplo em C# a seguir usa o AWS SDK for .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 usando multipart upload \(p. 210\)](#).

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 do Amazon S3 .NET \(p. 1268\)](#).

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

namespace Amazon.DocSamples.S3
{
    class CopyObjectTest
    {
        private const string sourceBucket = "*** provide the name of the bucket with
source object ***";
        private const string destinationBucket = "*** provide the name of the bucket to
copy the object to ***";
        private const string objectKey = "*** provide the name of object to copy ***";
        private const string destObjectKey = "*** provide the destination object key
name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            Console.WriteLine("Copying an object");
            CopyingObjectAsync().Wait();
        }

        private static async Task CopyingObjectAsync()
        {
            try
            {
                CopyObjectRequest request = new CopyObjectRequest
                {
                    SourceBucket = sourceBucket,
                    SourceKey = objectKey,
                    DestinationBucket = destinationBucket,
                    DestinationKey = destObjectKey
                };
                CopyObjectResponse response = await s3Client.CopyObjectAsync(request);
            }
        }
    }
}
```

```
        }
        catch (AmazonS3Exception e)
        {
            Console.WriteLine("Error encountered on server. Message:{0}' when
writing an object", e.Message);
        }
        catch (Exception e)
        {
            Console.WriteLine("Unknown encountered on server. Message:{0}' when
writing an object", e.Message);
        }
    }
}
```

PHP

Este tópico fornece orientações sobre como usar classes da versão 3 do AWS SDK for PHP para copiar um ou múltiplos objetos no Amazon S3 de um bucket para outro ou no mesmo bucket.

Este tópico considera que você já está seguindo as instruções para [Usar o AWS SDK for PHP e executar exemplos do PHP \(p. 1268\)](#) e tem o AWS SDK for PHP devidamente instalado.

O exemplo de PHP a seguir ilustra o uso do método `copyObject()` para copiar um único objeto no Amazon S3. Mostra também como um lote de chamadas para `CopyObject`, por meio do método `getCommand()`, é usado para fazer várias cópias de um objeto.

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 várias cópias de um objeto, execute um lote de chamadas para o método <code>getCommand()</code> do cliente do Amazon S3, 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.

```
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
}
```

Ruby

As tarefas a seguir orientam como usar classes Ruby para copiar um objeto no Amazon S3 de um bucket para outro ou no mesmo bucket.

Cópia de objetos

1	Use o gem modularizado do Amazon S3 da versão 3 do AWS SDK for Ruby, exija 'aws-sdk-s3' e forneça suas credenciais da AWS. Para obter mais informações sobre como fornecer suas credenciais, consulte Fazer solicitações usando credenciais de usuário do IAM ou da Conta da AWS (p. 1220) .
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.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyWrapper
    attr_reader :source_object

    # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is used
    # as the source object for
    #                                         copy actions.
    def initialize(source_object)
        @source_object = source_object
    end

    # Copy the source object to the specified target bucket and rename it with the target
    # key.
    #
    # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
    # object is copied.
    # @param target_object_key [String] The key to give the copy of the object.
    # @return [Aws::S3::Object, nil] The copied object when successful; otherwise, nil.
    def copy_object(target_bucket, target_object_key)
        @source_object.copy_to(bucket: target_bucket.name, key: target_object_key)
        target_bucket.object(target_object_key)
```

```
rescue Aws::Errors::ServiceError => e
  puts "Couldn't copy #{@source_object.key} to #{target_object.key}. Here's why:
#{e.message}"
end

# Replace the source and target bucket names with existing buckets you own and replace
# the source object key
# with an existing object in the source bucket.
def run_demo
  source_bucket_name = "doc-example-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "doc-example-bucket2"
  target_key = "my-target-file.txt"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key)
  return unless target_object

  puts "Copied #{source_key} from #{source_bucket_name} to
#{target_object.bucket_name}:#{target_object.key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Copiar um objeto usando a API REST

Este exemplo descreve como copiar um objeto usando REST. Para obter mais informações sobre a API REST, consulte [PUT Object \(Cópia\)](#).

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 resposta a seguir, que especifica a ETag do objeto e quando ele foi modificado pela última vez.

```
HTTP/1.1 200 OK
x-amz-id-2: Vyaxt7qEbqv34BnSu5hctyyNSlHTYZFMWK4FtzO+iX8JQnyaLdTshL0KxatbaOzt
x-amz-request-id: 6B13C3C5B34AF333
Date: Wed, 20 Feb 2008 22:13:01 +0000
```

```
Content-Type: application/xml
Transfer-Encoding: chunked
Connection: close
Server: AmazonS3
<?xml version="1.0" encoding="UTF-8"?>

<CopyObjectResult>
    <LastModified>2008-02-20T22:13:01</LastModified>
    <ETag>"7e9c608af58950deeb370c98608ed097"</ETag>
</CopyObjectResult>
```

Fazer download de um objeto

Esta seção explica como fazer download de objetos de um bucket do S3.

As taxas de transferência de dados se aplicam ao baixar objetos. Para obter informações sobre os recursos e a definição de preço do Amazon S3, consulte [Amazon S3](#).

É possível fazer download de um único objeto por solicitação usando o console do Amazon S3. Para baixar vários objetos, use a AWS CLI, os AWS SDKs ou a API REST.

Quando você faz download de um objeto programaticamente, seus metadados são retornados nos cabeçalhos de 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. A API REST GET Object (consulte [GET Object](#)) permite que você especifique parâmetros de query string 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 \(p. 358\)](#).

Uso do console do S3

Esta seção explica como usar o console do Amazon S3 para fazer download de um objeto contido em um bucket do S3 usando um URL pré-assinado.

Note

- Apenas um objeto pode ser obtido por download de cada vez.
- Objetos com nomes de chave terminando com pontos “.” e baixados usando o console do Amazon S3 terão os pontos “.” removidos do nome da chave do objeto baixado. Para baixar um objeto com o nome da chave terminado com ponto “.” mantido no objeto baixado, será necessário usar a AWS Command Line Interface (AWS CLI), AWS SDKs ou a API REST. Para obter exemplos e informações sobre a AWS CLI, a API REST e o AWS SDK, consulte [Baixar um objeto](#).

Fazer download de um objeto de um bucket do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket do qual você deseja fazer download de um objeto.

3. Você pode baixar um objeto de um bucket do S3 de uma das seguintes formas:

- No menu Actions (Ações), selecione o objeto e escolha Download (Baixar) ou Download as (Baixar como) se quiser baixar o objeto para uma pasta específica.
- Se você quiser baixar uma versão específica do objeto, selecione o botão Show versions (Mostrar versões). No menu Actions (Ações), selecione a versão do objeto desejado e escolha Download (Baixar) ou Download as (Baixar como) se quiser baixar o objeto para uma pasta específica.

Uso de AWS SDKs

Java

Quando você baixa um objeto por meio do AWS SDK for Java, o Amazon S3 retorna todos os metadados do objeto e um fluxo de entrada de onde é possível 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.
- Opcionalmente, é possível substituir os valores do cabeçalho de resposta usando um objeto `ResponseHeaderOverrides` e definindo a propriedade de 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 um objeto completo, depois, como uma faixa de bytes do objeto e, por fim, como um objeto completo com os valores do cabeçalho da resposta substituídos. Para obter mais informações sobre como obter objetos do Amazon S3, consulte [GET Object](#). Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.GetObjectRequest;
import com.amazonaws.services.s3.model.ResponseHeaderOverrides;
import com.amazonaws.services.s3.model.S3Object;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
```

```
public class GetObject2 {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_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();
}
```

.NET

Ao fazer download de um objeto, você obtém todos os metadados do objeto e um fluxo do qual você lê o conteúdo. É necessário ler o conteúdo do fluxo o mais rápido possível porque os dados são transmitidos 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 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 [Fazer download de um objeto \(p. 223\)](#)) 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. 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 do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
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 medata 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)
            {
                // If bucket or object does not exist
                Console.WriteLine("Error encountered ***. Message:{0}' when reading
object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:{0}' when
reading object", e.Message);
            }
        }
    }
}
```

```
        }
    }
}
```

PHP

Este tópico explica como usar uma classe do AWS SDK for 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 for PHP e executar exemplos do PHP \(p. 1268\)](#) e tem o AWS SDK for 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 [Fazer download de um objeto \(p. 223\)](#).

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 \(p. 1269\)](#).

```
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;
}
```

Uso dos REST API

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, consulte [Objeto GET](#).

Uso do AWS CLI

O exemplo a seguir mostra como você pode usar a AWS CLI para baixar um objeto do Amazon S3. Para obter mais informações e exemplos, consulte [get-object](#) na Referência de comandos da AWS CLI.

```
aws s3api get-object --bucket DOC-EXAMPLE-BUCKET1 --key dir/my_images.tar.bz2  
my_images.tar.bz2
```

Verificar a integridade do objeto

O Amazon S3 usa valores de soma de verificação para verificar a integridade dos dados que você carrega ou baixa do Amazon S3. Além disso, você pode solicitar que outro valor de soma de verificação seja calculado para qualquer objeto armazenado no Amazon S3. Você pode selecionar um dos vários algoritmos de soma de verificação para usar ao carregar ou copiar seus dados. O Amazon S3 usa esse algoritmo para calcular um valor de soma de verificação adicional e armazená-lo como parte dos metadados do objeto.

Ao carregar um objeto, você pode incluir uma soma de verificação pré-calculada como parte de sua solicitação. O Amazon S3 compara a soma de verificação fornecida com a soma de verificação calculada usando o algoritmo especificado. Se os dois valores não corresponderem, o Amazon S3 gerará um erro.

Usar algoritmos de soma de verificação compatíveis

O Amazon S3 oferece a opção de escolher o algoritmo de soma de verificação usado para validar seus dados durante o upload ou o download. Você pode selecionar um dos seguintes algoritmos de verificação de integridade de dados de hash seguro (SHA) ou de verificação de redundância ciclica (CRC):

- CRC32
- CRC32C
- SHA-1
- SHA-256

Ao carregar um objeto, você pode especificar o algoritmo que deseja usar:

- Quando você estiver usando o AWS Management Console, selecione o algoritmo de soma de verificação que deseja usar. Ao fazer isso, você pode especificar o valor da soma de verificação do objeto. Quando o Amazon S3 recebe o objeto, ele calcula a soma de verificação usando o algoritmo especificado. Se os dois valores de soma de verificação não corresponderem, o Amazon S3 gerará um erro.

- Ao usar um SDK, você pode definir o valor do parâmetro `x-amz-sdk-checksum-algorithm` para o algoritmo que deseja que o Amazon S3 use ao calcular a soma de verificação. O Amazon S3 calcula automaticamente o valor da soma de verificação.
- Ao utilizar a API REST, não use o parâmetro `x-amz-sdk-checksum-algorithm`. Em vez disso, use um dos cabeçalhos específicos do algoritmo (por exemplo, `x-amz-checksum-crc32`).

Para obter mais informações sobre como carregar objetos, consulte [Fazer upload de objetos \(p. 171\)](#).

Para aplicar qualquer um desses valores de soma de verificação a objetos já carregados para o Amazon S3, você pode copiar o objeto. Ao copiar um objeto, você pode especificar se deseja usar o algoritmo de soma de verificação existente ou usar um novo. É possível especificar um algoritmo de soma de verificação ao usar qualquer mecanismo compatível para copiar objetos, incluindo operações em lote do S3. Para obter mais informações sobre o S3 Batch Operations, consulte [Executar operações em lote de grande escala em objetos do Amazon S3 \(p. 938\)](#).

Important

Se você estiver usando um carregamento fracionado com somas de verificação adicionais, os números das partes fracionadas deverão ser consecutivos. Ao usar somas de verificação adicionais, se você tentar concluir uma solicitação de carregamento fracionado com números de parte não consecutivos, o Amazon S3 gerará um erro HTTP 500 Internal Server Error.

Depois de carregar objetos, você pode obter o valor da soma de verificação e compará-lo com um valor de soma de verificação pré-calculado ou armazenado anteriormente que tenha sido calculado com o mesmo algoritmo.

Uso do console do S3

Para saber mais sobre como usar o console e especificar algoritmos de soma de verificação a serem usados ao carregar objetos, consulte [Fazer upload de objetos \(p. 171\)](#)

Uso de AWS SDKs

O exemplo a seguir mostra como usar os AWS SDKs para carregar um arquivo grande com carregamento fracionado, baixar um arquivo grande e validar um arquivo de carregamento fracionado por meio do SHA-256 para validação de arquivos.

Java

Example Exemplo: carregar, baixar e verificar um arquivo grande com SHA-256

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
import software.amazon.awssdk.auth.credentials.AwsCredentials;
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;
import software.amazon.awssdk.core.ResponseInputStream;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.AbortMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.ChecksumAlgorithm;
import software.amazon.awssdk.services.s3.model.ChecksumMode;
import software.amazon.awssdk.services.s3.model.CompleteMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.CompleteMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.CompletedMultipartUpload;
import software.amazon.awssdk.services.s3.model.CompletedPart;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadResponse;
```

```
import software.amazon.awssdk.services.s3.model.GetObjectAttributesRequest;
import software.amazon.awssdk.services.s3.model.GetObjectAttributesResponse;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.GetObjectTaggingRequest;
import software.amazon.awssdk.services.s3.model.ObjectAttributes;
import software.amazon.awssdk.services.s3.model.PutObjectTaggingRequest;
import software.amazon.awssdk.services.s3.model.Tag;
import software.amazon.awssdk.services.s3.model.Tagging;
import software.amazon.awssdk.services.s3.model.UploadPartRequest;
import software.amazon.awssdk.services.s3.model.UploadPartResponse;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.nio.ByteBuffer;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.Base64;
import java.util.List;

public class LargeObjectValidation {
    private static String FILE_NAME = "sample.file";
    private static String BUCKET = "sample-bucket";
    //Optional, if you want a method of storing the full multipart object checksum
    in S3.
    private static String CHECKSUM_TAG_KEYNAME = "fullObjectChecksum";
    //If you have existing full-object checksums that you need to validate against,
    you can do the full object validation on a sequential upload.
    private static String SHA256_FILE_BYTES = "htCM5g7ZNdoSw8bN/mkgiAhXt5MFoVowVg
+LE9aIQmI=";
    //Example Chunk Size - this must be greater than or equal to 5MB.
    private static int CHUNK_SIZE = 5 * 1024 * 1024;

    public static void main(String[] args) {
        S3Client s3Client = S3Client.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(new AwsCredentialsProvider() {
                @Override
                public AwsCredentials resolveCredentials() {
                    return new AwsCredentials();
                }
                @Override
                public String accessKeyId() {
                    return Constants.ACCESS_KEY;
                }
                @Override
                public String secretAccessKey() {
                    return Constants.SECRET;
                }
            });
        uploadLargeFileBracketedByChecksum(s3Client);
        downloadLargeFileBracketedByChecksum(s3Client);
        validateExistingFileAgainstS3Checksum(s3Client);
    }

    public static void uploadLargeFileBracketedByChecksum(S3Client s3Client) {
        System.out.println("Starting uploading file validation");
        File file = new File(FILE_NAME);
```

```
try (InputStream in = new FileInputStream(file)) {
    MessageDigest sha256 = MessageDigest.getInstance("SHA-256");
    CreateMultipartUploadRequest createMultipartUploadRequest =
CreateMultipartUploadRequest.builder()
        .bucket(BUCKET)
        .key(FILE_NAME)
        .checksumAlgorithm(ChecksumAlgorithm.SHA256)
        .build();
    CreateMultipartUploadResponse createdUpload =
s3Client.createMultipartUpload(createMultipartUploadRequest);
    List<CompletedPart> completedParts = new ArrayList<CompletedPart>();
    int partNumber = 1;
    byte[] buffer = new byte[CHUNK_SIZE];
    int read = in.read(buffer);
    while (read != -1) {
        UploadPartRequest uploadPartRequest = UploadPartRequest.builder()
            .partNumber(partNumber).uploadId(createdUpload.uploadId()).key(FILE_NAME).bucket(BUCKET).checksumAlgorithm(SHA256)
            UploadPartResponse uploadedPart =
s3Client.uploadPart(uploadPartRequest,
RequestBody.fromByteBuffer(ByteBuffer.wrap(buffer, 0, read)));
        CompletedPart part =
CompletedPart.builder().partNumber(partNumber).checksumSHA256(uploadedPart.checksumSHA256()).eTag(uploadedPart.eTag())
            completedParts.add(part);
        sha256.update(buffer, 0, read);
        read = in.read(buffer);
        partNumber++;
    }
    String fullObjectChecksum =
Base64.getEncoder().encodeToString(sha256.digest());
    if (!fullObjectChecksum.equals(SHA256_FILE_BYTES)) {
        //Because the SHA256 is uploaded after the part is uploaded; the
        upload is bracketed and the full object can be fully validated.
    }
    CompletedMultipartUpload completedMultipartUpload =
CompletedMultipartUpload.builder().parts(completedParts).build();
    CompleteMultipartUploadResponse completedUploadResponse =
s3Client.completeMultipartUpload(
    CompleteMultipartUploadRequest.builder().bucket(BUCKET).key(FILE_NAME).uploadId(createdUpload.uploadId())
        Tag checksumTag =
        Tag.builder().key(CHECKSUM_TAG_KEYNAME).value(fullObjectChecksum).build();
        // Optionally, if you need the full object checksum stored with the
        file; you could add it as a tag after completion.
    s3Client.putObjectTagging(PutObjectTaggingRequest.builder().bucket(BUCKET).key(FILE_NAME).tagging()
        } catch (IOException | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
    GetObjectAttributesResponse
        objectAttributes =
s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_NAME)
        .objectAttributes(ObjectAttributes.OBJECT_PARTS,
ObjectAttributes.CHECKSUM).build());
        System.out.println(objectAttributes.objectParts().parts());
        System.out.println(objectAttributes.checksum().checksumSHA256());
    }

    public static void downloadLargeFileBracketedByChecksum(S3Client s3Client) {
        System.out.println("Starting downloading file validation");
        File file = new File("DOWNLOADED_" + FILE_NAME);
        try (OutputStream out = new FileOutputStream(file)) {
```

```
        GetObjectAttributesResponse
            objectAttributes =
s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_NAME)
                                .objectAttributes(ObjectAttributes.OBJECT_PARTS,
ObjectAttributes.CHECKSUM).build());
        //Optionally if you need the full object checksum, you can grab a tag
you added on the upload
        List<Tag> objectTags =
s3Client.getObjectTagging(GetObjectTaggingRequest.builder().bucket(BUCKET).key(FILE_NAME).build())
        String fullObjectChecksum = null;
        for (Tag objectTag : objectTags) {
            if (objectTag.key().equals(CHECKSUM_TAG_KEYNAME)) {
                fullObjectChecksum = objectTag.value();
                break;
            }
        }
        MessageDigest sha256FullObject = MessageDigest.getInstance("SHA-256");
        MessageDigest sha256ChecksumOfChecksums =
MessageDigest.getInstance("SHA-256");

        //If you retrieve the object in parts, and set the ChecksumMode to
enabled, the SDK will automatically validate the part checksum
        for (int partNumber = 1; partNumber <=
objectAttributes.objectParts().totalPartsCount(); partNumber++) {
            MessageDigest sha256Part = MessageDigest.getInstance("SHA-256");
            ResponseInputStream<GetObjectResponse> response =
s3Client.getObject(GetObjectRequest.builder().bucket(BUCKET).key(FILE_NAME).partNumber(partNumber)
                                GetObjectResponse getObjectResponse = response.response();
            byte[] buffer = new byte[CHUNK_SIZE];
            int read = response.read(buffer);
            while (read != -1) {
                out.write(buffer, 0, read);
                sha256FullObject.update(buffer, 0, read);
                sha256Part.update(buffer, 0, read);
                read = response.read(buffer);
            }
            byte[] sha256PartBytes = sha256Part.digest();
            sha256ChecksumOfChecksums.update(sha256PartBytes);
            //Optionally, you can do an additional manual validation again the
part checksum if needed in addition to the SDK check
            String base64PartChecksum =
Base64.getEncoder().encodeToString(sha256PartBytes);
            String base64PartChecksumFromObjectAttributes =
objectAttributes.objectParts().parts().get(partNumber - 1).checksumSHA256();
            if (!base64PartChecksum.equals(getObjectResponse.checksumSHA256()))
|| !base64PartChecksum.equals(base64PartChecksumFromObjectAttributes)) {
                throw new IOException("Part checksum didn't match for the
part");
            }
            System.out.println(partNumber + " " + base64PartChecksum);
        }
        //Before finalizing, do the final checksum validation.
        String base64FullObject =
Base64.getEncoder().encodeToString(sha256FullObject.digest());
        String base64ChecksumOfChecksums =
Base64.getEncoder().encodeToString(sha256ChecksumOfChecksums.digest());
        if (fullObjectChecksum != null && !
fullObjectChecksum.equals(base64FullObject)) {
            throw new IOException("Failed checksum validation for full
object");
        }
        System.out.println(fullObjectChecksum);
        String base64ChecksumOfChecksumFromAttributes =
objectAttributes.checksum().checksumSHA256();
        if (base64ChecksumOfChecksumFromAttributes != null && !
base64ChecksumOfChecksums.equals(base64ChecksumOfChecksumFromAttributes)) {
```

```
        throw new IOException("Failed checksum validation for full object
checksum of checksums");
    }
    System.out.println(base64ChecksumOfChecksumFromAttributes);
    out.flush();
} catch (IOException | NoSuchAlgorithmException e) {
    //Cleanup bad file
    file.delete();
    e.printStackTrace();
}
}

public static void validateExistingFileAgainstS3Checksum(S3Client s3Client) {
    System.out.println("Starting existing file validation");
    File file = new File("DOWNLOADED_" + FILE_NAME);
    GetObjectAttributesResponse
        objectAttributes =
s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_NAME)
        .objectAttributes(ObjectAttributes.OBJECT_PARTS,
ObjectAttributes.CHECKSUM).build());
    try (InputStream in = new FileInputStream(file)) {
        MessageDigest sha256ChecksumOfChecksums =
MessageDigest.getInstance("SHA-256");
        MessageDigest sha256Part = MessageDigest.getInstance("SHA-256");
        byte[] buffer = new byte[CHUNK_SIZE];
        int currentPart = 0;
        int partBreak =
objectAttributes.objectParts().parts().get(currentPart).size();
        int totalRead = 0;
        int read = in.read(buffer);
        while (read != -1) {
            totalRead += read;
            if (totalRead >= partBreak) {
                int difference = totalRead - partBreak;
                byte[] partChecksum;
                if (totalRead != partBreak) {
                    sha256Part.update(buffer, 0, read - difference);
                    partChecksum = sha256Part.digest();
                    sha256ChecksumOfChecksums.update(partChecksum);
                    sha256Part.reset();
                    sha256Part.update(buffer, read - difference, difference);
                } else {
                    sha256Part.update(buffer, 0, read);
                    partChecksum = sha256Part.digest();
                    sha256ChecksumOfChecksums.update(partChecksum);
                    sha256Part.reset();
                }
                String base64PartChecksum =
Base64.getEncoder().encodeToString(partChecksum);
                if (!
base64PartChecksum.equals(objectAttributes.objectParts().parts().get(currentPart).checksumSHA256()))
{
                    throw new IOException("Part checksum didn't match S3");
                }
                currentPart++;
                System.out.println(currentPart + " " + base64PartChecksum);
                if (currentPart <
objectAttributes.objectParts().totalPartsCount()) {
                    partBreak +=
objectAttributes.objectParts().parts().get(currentPart - 1).size();
                }
            } else {
                sha256Part.update(buffer, 0, read);
            }
            read = in.read(buffer);
        }
    }
}
```

```
        if (currentPart != objectAttributes.objectParts().totalPartsCount()) {
            currentPart++;
            byte[] partChecksum = sha256Part.digest();
            sha256ChecksumOfChecksums.update(partChecksum);
            String base64PartChecksum =
                Base64.getEncoder().encodeToString(partChecksum);
                System.out.println(currentPart + " " + base64PartChecksum);
        }

        String base64CalculatedChecksumOfChecksums =
    Base64.getEncoder().encodeToString(sha256ChecksumOfChecksums.digest());
        System.out.println(base64CalculatedChecksumOfChecksums);
        System.out.println(objectAttributes.checksum().checksumSHA256());
        if (!
base64CalculatedChecksumOfChecksums.equals(objectAttributes.checksum().checksumSHA256()))
{
        throw new IOException("Full object checksum of checksums don't
match S3");
    }

} catch (IOException | NoSuchAlgorithmException e) {
    e.printStackTrace();
}
}
}
```

Uso dos REST API

Você pode enviar solicitações REST para carregar um objeto com um valor de soma de verificação para verificar a integridade dos dados com [PutObject](#). Você também pode recuperar o valor da soma de verificação para objetos usando [GetObject](#) ou [HeadObject](#).

Uso do AWS CLI

Você pode enviar uma solicitação `PUT` para carregar um objeto de até 5 GB em uma única operação. Para obter mais informações, consulte [PutObject](#) na Referência de comando da AWS CLI. Você também pode usar `get-object` e `head-object` para recuperar a soma de verificação de um objeto já carregado a fim de verificar a integridade dos dados.

Usar Content-MD5 ao carregar objetos

Outra maneira de verificar a integridade do objeto após o carregamento é fornecer um resumo MD5 do objeto quando você o carrega. Se você calcular o resumo MD5 para seu objeto, poderá fornecê-lo com o comando `PUT` usando o cabeçalho `Content-MD5`.

Após o carregamento do objeto, o Amazon S3 calcula o resumo MD5 do objeto e o compara com o valor que você forneceu. A solicitação só será bem-sucedida se os dois resumos corresponderem.

Não é necessário fornecer um resumo MD5, mas você pode usá-lo para verificar a integridade do objeto como parte do processo de carregamento.

Usar Content-MD5 e a ETag para verificar objetos carregados

A etiqueta de entidade (ETag) de um objeto representa uma versão específica dele. Lembre-se de que a ETag reflete as alterações apenas no conteúdo de um objeto, não nos respectivos metadados. Se apenas os metadados de um objeto forem alterados, a ETag permanecerá a mesma.

Dependendo do objeto, a ETag do objeto pode ser um resumo MD5 dos dados do objeto:

- Se um objeto for criado pela operação `PUT Object`, `POST Object` ou `Copy`, ou por meio do AWS Management Console, e ele também for em texto simples ou criptografado no lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3), ele terá uma ETag que é um resumo MD5 dos respectivos dados.
- Se um objeto for criado pela operação `PUT Object`, `POST Object` ou `Copy`, ou por meio do AWS Management Console, e for criptografado no lado do servidor com chaves fornecidas pelo cliente (SSE-C) ou com criptografia no lado do servidor com chaves AWS Key Management Service(AWS KMS) (SSE-KMS), ele terá uma ETag que não é um resumo MD5 dos respectivos dados.
- Se um objeto for criado pela operação `Multipart Upload` ou `Part Copy`, a respectiva ETag não será um resumo MD5, independentemente do método de criptografia. Se um objeto for maior que 16 MB, o AWS Management Console o carregará ou copiará como carregamento fracionado e, por isso, a ETag não será um resumo MD5.

Para objetos em que a ETag é o resumo Content-MD5 do objeto, você pode comparar o valor da ETag do objeto com um resumo Content-MD5 calculado ou armazenado anteriormente.

Usar somas de verificação à direita

Ao carregar objetos para o Amazon S3, você pode fornecer uma soma de verificação pré-calculada para o objeto ou usar um AWS SDK para criar automaticamente somas de verificação à direita em seu nome. Se você decidir usar uma soma de verificação à direita, o Amazon S3 a gerará automaticamente usando o algoritmo especificado e a usará para validar a integridade do objeto durante o carregamento.

Para criar uma soma de verificação à direita ao usar um AWS SDK, preencha o parâmetro `ChecksumAlgorithm` com seu algoritmo preferido. O SDK usa esse algoritmo para calcular a soma de verificação do objeto (ou partes do objeto) e o anexa automaticamente ao final da solicitação de carregamento. Esse comportamento economiza tempo porque o Amazon S3 executa a verificação e o carregamento de seus dados em uma única passagem.

Important

Se estiver usando o S3 Object Lambda, todas as solicitações ao S3 Object Lambda serão assinadas usando `s3-object-lambda` em vez de `s3`. Esse comportamento afeta a assinatura dos valores de soma de verificação à direita. Para obter mais informações sobre o S3 Object Lambda, consulte [Transformar objetos com o S3 Object Lambda \(p. 288\)](#).

Usar somas de verificação de nível parcial para carregamentos fracionados

Quando os objetos são carregados para o Amazon S3, eles podem ser carregados como um único objeto ou por meio do processo de carregamento fracionado. Objetos maiores que 16 MB e carregados pelo console são carregados automaticamente por meio de carregamentos fracionados. Para obter mais informações sobre multipart uploads, consulte [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).

Quando um objeto é carregado como um carregamento fracionado, a ETag do objeto não é um resumo MD5 do objeto completo. O Amazon S3 calcula o resumo MD5 de cada parte à medida que ela é carregada. Os resumos MD5 são usados para determinar a ETag do objeto final. O Amazon S3 concatena os bytes para os resumos MD5 e, depois, calcula o resumo MD5 desses valores concatenados. A etapa final para criar a ETag ocorre quando o Amazon S3 adiciona um traço com o número total de partes ao final.

Por exemplo, considere um objeto com carregamento fracionado que tenha uma ETag de `C9A5A6878D97B48CC965C1E41859F034-14`. Nesse caso, o `C9A5A6878D97B48CC965C1E41859F034`

é o resumo MD5 de todos os resumos concatenados. O -14 indica que há 14 partes associadas ao carregamento fracionado desse objeto.

Se você tiver habilitado valores de soma de verificação adicionais para o objeto fracionado, o Amazon S3 calculará a soma de verificação de cada parte usando o algoritmo de soma de verificação especificado. A soma de verificação do objeto concluído é calculada da mesma forma que o Amazon S3 calcula o resumo MD5 do carregamento fracionado. Você pode usar essa soma de verificação para verificar a integridade do objeto.

Para recuperar informações sobre o objeto, incluindo o número de partes que compõem o objeto inteiro, você pode usar a operação [GetObjectAttributes](#). Com somas de verificação adicionais, também é possível recuperar informações de cada parte que inclui o valor da soma de verificação de cada uma.

Você também pode obter a soma de verificação de uma parte usando a operação [GetObject](#) ou [HeadObject](#) e especificando um número de parte ou um intervalo de bytes que se alinhe a uma única parte. Com esse método, você pode usar essa soma de verificação para validar a parte sem precisar esperar a conclusão do carregamento de todas as partes para verificar a integridade dos dados. Ao usar esse método, você também pode solicitar somente as partes que falharam no teste de integridade.

Devido à forma como o Amazon S3 calcula a soma de verificação para objetos fracionados, o valor da soma de verificação do objeto poderá ser alterado se você copiá-lo. Se estiver usando um SDK ou a API REST e chamar [CopyObject](#), o Amazon S3 copiará qualquer objeto até as limitações de tamanho da operação `CopyObject` da API. O Amazon S3 faz essa cópia como uma única ação, independentemente de o objeto ter sido carregado em uma única solicitação ou como parte de um carregamento fracionado. Com um comando `copy`, a soma de verificação do objeto é uma soma de verificação direta do objeto completo. Se o objeto tiver sido originalmente carregado usando um carregamento fracionado, o valor da soma de verificação será alterado mesmo que os dados não mudem.

Note

Os objetos que ultrapassam as limitações de tamanho da operação `CopyObject` da API devem usar comandos de cópia fracionada.

Important

Quando você executar algumas operações usando o AWS Management Console, o Amazon S3 usará o carregamento fracionado se o objeto tiver mais de 16 MB. Nesse caso, a soma de verificação não é uma soma de verificação direta do objeto completo, mas um cálculo baseado nos valores da soma de verificação de cada parte.

Por exemplo, considere um objeto de 100 MB que você carregou como um carregamento direto de uma única parte usando a API REST. Nesse caso, a soma de verificação é do objeto completo. Se você usar o console posteriormente para renomear esse objeto, copiá-lo, alterar a classe de armazenamento ou editar os metadados, o Amazon S3 usará a funcionalidade de carregamento fracionado para atualizá-lo. Por isso, o Amazon S3 cria um valor de soma de verificação para o objeto que é calculado com base nos valores de soma de verificação das partes.

A lista anterior de operações do console não é uma lista completa de todas as ações possíveis que você pode executar no AWS Management Console que resultam na atualização do objeto do Amazon S3 usando a funcionalidade de carregamento fracionado. Lembre-se de que sempre que usar o console para lidar com objetos com mais de 16 MB, o valor da soma de verificação pode não ser a soma de verificação do objeto completo.

Excluir objetos do Amazon S3

Você pode excluir um ou mais objetos diretamente do Amazon S3 usando o console do Amazon S3, os AWS SDKs, a AWS Command Line Interface (AWS CLI) ou a API REST. Como todos os objetos em seu bucket do S3 incorrem em custos de armazenamento, você deve excluir os objetos de que não precisa

mais. Se estiver coletando arquivos de log, por exemplo, é uma boa ideia excluí-los quando eles não forem mais necessários. Você pode definir uma regra de ciclo de vida para excluir automaticamente objetos como arquivos de log. Para obter mais informações, consulte [the section called “Definir a configuração do ciclo de vida” \(p. 753\)](#).

Para obter informações sobre os recursos e a definição de preço do Amazon S3, consulte a [definição de preço do Amazon S3](#).

Você tem as seguintes opções da API para excluir um objeto:

- Excluir um único objeto: o Amazon S3 fornece a API DELETE que pode ser usada para excluir um objeto em uma única solicitação HTTP.
- Excluir vários objetos: o Amazon S3 fornece a API para exclusão de vários objetos que pode ser usada para excluir até 1.000 objetos em uma única solicitação HTTP.

Ao excluir objetos de um bucket que não esteja habilitado para versionamento, você fornece apenas o nome da chave de objeto. No entanto, ao excluir objetos de um bucket habilitado para versionamento, você tem a opção de fornecer o ID da versão do objeto para excluir uma versão específica dele.

Excluir objetos de um bucket habilitado para versionamento de maneira programática

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: especifique 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 [Usando o versionamento em buckets do S3 \(p. 678\)](#).
- Especificar uma solicitação de exclusão versionada: especifique 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 você 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.

No entanto, nos seguintes casos, a solicitação é bem-sucedida:

- Se você tiver um bucket habilitado para MFA, fizer uma solicitação de exclusão não versionada (sem excluir um objeto versionado) e não fornecer um token de MFA, a exclusão será concluída.

- Se você tiver uma solicitação de exclusão de vários objetos especificando somente objetos não versionados 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 a exclusão de MFA, consulte [Configurando a exclusão de MFA \(p. 688\)](#).

Tópicos

- [Excluir um único objeto \(p. 239\)](#)
- [Excluir vários objetos \(p. 246\)](#)

Excluir um único objeto

Você pode usar o console do Amazon S3 ou a API DELETE para excluir um único objeto existente de um bucket do S3. Para obter mais informações sobre como excluir objetos no Amazon S3, consulte [Excluir objetos do Amazon S3 \(p. 237\)](#).

Como todos os objetos em seu bucket do S3 incorrem em custos de armazenamento, você deve excluir os objetos de que não precisa mais. Por exemplo, se estiver coletando arquivos de log, é uma boa ideia excluí-los quando não forem mais necessários. Você pode definir uma regra de ciclo de vida para excluir automaticamente objetos como arquivos de log. Para obter mais informações, consulte [the section called "Definir a configuração do ciclo de vida" \(p. 753\)](#).

Para obter informações sobre os recursos e a definição de preço do Amazon S3, consulte a [definição de preço do Amazon S3](#).

Uso do console do S3

Siga as etapas a seguir para usar o console do Amazon S3 a fim de excluir um único objeto de um bucket.

Para excluir um objeto

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Bucket name (Nome do bucket), escolha o nome do bucket do qual você deseja excluir um objeto.
3. Para excluir um objeto em um bucket habilitado para versionamento utilizando o comando de versionamento:
 - Off (Desativar), o Amazon S3 criará um marcador de exclusão. Para excluir o objeto, selecione o objeto, escolha delete (excluir) e confirme a sua escolha digitando **delete** no campo de texto.
 - On (Ativar), o Amazon S3 excluirá permanentemente a versão do objeto. Selecione a versão do objeto que deseja excluir, escolha delete (excluir) e confirme sua escolha digitando **permanently delete** no campo de texto.

Uso de AWS SDKs

Os exemplos a seguir mostram como você pode usar os AWS SDKs para excluir um objeto de um bucket. Para obter mais informações, acesse [DELETE Object](#) na Referência de APIs do Amazon Simple Storage Service.

Se tiver habilitado o versionamento do S3 no bucket, você terá 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 Amazon S3 adicionará um marcador de exclusão para o objeto.

Para obter mais informações sobre o S3 Versioning, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

Java

Example Exemplo 1: Excluir um objeto (bucket sem versionamento)

O exemplo a seguir 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 como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

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

import java.io.IOException;

public class DeleteObjectNonVersionedBucket {

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

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

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

Example 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 exemplo ao bucket. O Amazon S3 retorna o ID de 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`.

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

import java.io.IOException;

public class DeleteObjectVersionEnabledBucket {

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

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

            // Check to ensure that the bucket is versioning-enabled.
            String bucketVersionStatus =
s3Client.getBucketVersioningConfiguration(bucketName).getStatus();
            if (!bucketVersionStatus.equals(BucketVersioningConfiguration.ENABLED)) {
                System.out.printf("Bucket %s is not versioning-enabled.", bucketName);
            } else {
                // Add an object.
                PutObjectResult putResult = s3Client.putObject(bucketName, keyName,
"Sample content for deletion example.");
                System.out.printf("Object %s added to bucket %s\n", keyName,
bucketName);

                // Delete the version of the object that we just created.
                System.out.println("Deleting versioned object " + keyName);
                s3Client.deleteVersion(new DeleteVersionRequest(bucketName, keyName,
putResult.getVersionId()));
                System.out.printf("Object %s, version %s deleted\n", keyName,
putResult.getVersionId());
            }
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

```
        }  
    }  
}
```

.NET

Os exemplos a seguir mostram como excluir um objeto de buckets com e sem versionamento. Para obter mais informações sobre o S3 Versioning, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

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 do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;  
using Amazon.S3;  
using Amazon.S3.Model;  
using System;  
using System.Threading.Tasks;  
  
namespace Amazon.DocSamples.S3  
{  
    class DeleteObjectNonVersionedBucketTest  
    {  
        private const string bucketName = "*** bucket name ***";  
        private const string keyName = "*** object key ***";  
        // Specify your bucket region (an example region is shown).  
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;  
        private static IAmazonS3 client;  
  
        public static void Main()  
        {  
            client = new AmazonS3Client(bucketRegion);  
            DeleteObjectNonVersionedBucketAsync().Wait();  
        }  
        private static async Task DeleteObjectNonVersionedBucketAsync()  
        {  
            try  
            {  
                var deleteObjectRequest = new DeleteObjectRequest  
                {  
                    BucketName = bucketName,  
                    Key = keyName  
                };  
  
                Console.WriteLine("Deleting an object");  
                await client.DeleteObjectAsync(deleteObjectRequest);  
            }  
            catch (AmazonS3Exception e)  
            {  
                Console.WriteLine("Error encountered on server. Message:'{0}' when  
deleting an object", e.Message);  
            }  
            catch (Exception e)  
            {  
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when  
deleting an object", e.Message);  
            }  
        }  
    }  
}
```

```
        }
    }
```

Example 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 do S3 no bucket que você especificar (se o versionamento do S3 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 do Amazon S3 .NET \(p. 1268\)](#).

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

namespace Amazon.DocSamples.S3
{
    class DeleteObjectVersion
    {
        private const string bucketName = "*** versioning-enabled bucket name ***";
        private const string keyName = "*** Object Key Name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            CreateAndDeleteObjectVersionAsync().Wait();
        }

        private static async Task CreateAndDeleteObjectVersionAsync()
        {
            try
            {
                // Add a sample object.
                string versionID = await PutAnObject(keyName);

                // Delete the object by specifying an object key and a version ID.
                DeleteObjectRequest request = new DeleteObjectRequest
                {
                    BucketName = bucketName,
```

```
        Key = keyName,
        VersionId = versionID
    };
    Console.WriteLine("Deleting an object");
    await client.DeleteObjectAsync(request);
}
catch (AmazonS3Exception e)
{
    Console.WriteLine("Error encountered on server. Message:{0} when
deleting an object", e.Message);
}
catch (Exception e)
{
    Console.WriteLine("Unknown encountered on server. Message:{0} when
deleting an object", e.Message);
}
}

static async Task<string> PutAnObject(string objectKey)
{
    PutObjectRequest request = new PutObjectRequest
    {
        BucketName = bucketName,
        Key = objectKey,
        ContentBody = "This is the content body!"
    };
    PutObjectResponse response = await client.PutObjectAsync(request);
    return response.VersionId;
}
}
```

PHP

Este exemplo mostra como usar classes da versão 3 do AWS SDK for PHP para excluir um objeto de um bucket sem versionamento. Para obter informações sobre como excluir um objeto de um bucket com versões, consulte [Uso dos REST API \(p. 246\)](#).

Este exemplo considera que você já está seguindo as instruções para [Usar o AWS SDK for PHP e executar exemplos do PHP \(p. 1268\)](#) e tem o AWS SDK for PHP devidamente instalado. Para obter informações sobre a execução dos exemplos de PHP neste guia, consulte [Executar exemplos do PHP \(p. 1269\)](#).

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.

```
<?php

require 'vendor/autoload.php';

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

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

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

// 1. Delete the object from the bucket.
try
```

```
{  
    echo 'Attempting to delete ' . $keyname . '...' . PHP_EOL;  
  
    $result = $s3->deleteObject([  
        'Bucket' => $bucket,  
        'Key'     => $keyname  
    ]);  
  
    if ($result['DeleteMarker'])  
    {  
        echo $keyname . ' was deleted or does not exist.' . PHP_EOL;  
    } else {  
        exit('Error: ' . $keyname . ' was not deleted.' . PHP_EOL);  
    }  
}  
catch (S3Exception $e) {  
    exit('Error: ' . $e->getAwsErrorMessage() . PHP_EOL);  
}  
  
// 2. Check to see if the object was deleted.  
try  
{  
    echo 'Checking to see if ' . $keyname . ' still exists...' . PHP_EOL;  
  
    $result = $s3->getObject([  
        'Bucket' => $bucket,  
        'Key'     => $keyname  
    ]);  
  
    echo 'Error: ' . $keyname . ' still exists.';  
}  
catch (S3Exception $e) {  
    exit($e->getAwsErrorMessage());  
}
```

Javascript

Este exemplo mostra como usar a versão 3 do AWS SDK for JavaScript para excluir um objeto. Para obter mais informações sobre o AWS SDK for JavaScript, consulte [Usar a AWS SDK for JavaScript \(p. 1271\)](#).

```
import { DeleteObjectCommand } from "@aws-sdk/client-s3";  
import { s3Client } from "./libs/s3Client.js" // Helper function that creates Amazon S3 service client module.  
  
export const bucketParams = { Bucket: "BUCKET_NAME", Key: "KEY" };  
  
export const run = async () => {  
    try {  
        const data = await s3Client.send(new DeleteObjectCommand(bucketParams));  
        console.log("Success. Object deleted.", data);  
        return data; // For unit tests.  
    } catch (err) {  
        console.log("Error", err);  
    }  
};  
run();
```

Usar a AWS CLI

Para excluir um objeto por solicitação, use a API `DELETE`. Para obter mais informações, consulte [DELETE Object](#). Para obter mais informações sobre como usar a CLI para excluir um objeto, consulte [delete-object](#).

Uso dos REST API

Você pode usar os AWS SDKs para excluir um objeto. Contudo, se o seu aplicativo exigir, você pode enviar solicitações REST diretamente. Para obter mais informações, consulte [DELETE Object](#) na Referência de APIs do Amazon Simple Storage Service.

Excluir vários objetos

Como todos os objetos em seu bucket do S3 incorrem em custos de armazenamento, você deve excluir os objetos de que não precisa mais. Por exemplo, se estiver coletando arquivos de log, é uma boa ideia excluí-los quando não forem mais necessários. Você pode definir uma regra de ciclo de vida para excluir automaticamente objetos como arquivos de log. Para obter mais informações, consulte [the section called "Definir a configuração do ciclo de vida"](#) (p. 753).

Para obter informações sobre os recursos e a definição de preço do Amazon S3, consulte a [definição de preço do Amazon S3](#).

Você pode usar o console do Amazon S3 ou a API Multi-Object Delete para excluir vários objetos simultaneamente de um bucket do S3.

Uso do console do S3

Siga etapas a seguir para usar o console do Amazon S3 a fim de excluir vários objetos de um bucket.

Para excluir objetos

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
 2. Navegue até o bucket ou pasta do Amazon S3 que contém os objetos que você deseja excluir.
 3. Marque a caixa de seleção à esquerda dos nomes dos objetos que você deseja excluir.
 4. Escolha Actions (Ações) e escolha Delete (Excluir) na lista de opções exibida.
- Como alternativa, escolha Delete (Excluir) nas opções no canto superior direito.
5. Digite **delete** se solicitado para confirmar que você deseja excluir esses objetos.
 6. Escolha Delete objects (Excluir objetos) no canto inferior direito e o Amazon S3 excluirá os objetos especificados.

Warning

- A exclusão dos objetos especificados não poderá ser desfeita.
- Essa ação exclui todos os objetos especificados. Ao excluir pastas, aguarde a conclusão da ação de exclusão antes de adicionar objetos novos à pasta. Caso contrário, os objetos novos também podem ser excluídos.
- Para excluir um objeto em um bucket com versionamento de buckets desativado ou suspenso, o Amazon S3 criará um marcador de exclusão. Para desfazer a ação de exclusão, exclua esse marcador de exclusão. Para confirmar essa ação, digite **permanently delete**.
- Para excluir uma versão de objeto em um bucket com o versionamento ativado, o Amazon S3 excluirá permanentemente a versão do objeto. Para confirmar essa ação, digite **delete**.

Uso de AWS SDKs

O Amazon S3 fornece a API Multi-Object Delete , que você pode usar para 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 obter mais informações, consulte [Delete - Multi-Object Delete](#).

Para saber mais sobre exclusão de objetos, consulte [Excluir objetos do Amazon S3 \(p. 237\)](#).

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 adiciona um marcador de exclusão ao objeto.
- Especifique o nome da chave do objeto e o ID de versão a serem excluídos. O Amazon S3 exclui 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 instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.DeleteObjectsRequest;
import com.amazonaws.services.s3.model.DeleteObjectsRequest.KeyVersion;
import com.amazonaws.services.s3.model.DeleteObjectsResult;

import java.io.IOException;
import java.util.ArrayList;

public class DeleteMultipleObjectsNonVersionedBucket {

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

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .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.");
        } catch (AmazonServiceException | SdkClientException e) {
            System.out.println("There was an error deleting the objects: " + e.getMessage());
        }
    }
}
```

```
// 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 AWS Management Console.
3. Remove 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 AWS Management Console.

```
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import com.amazonaws.auth.profile.ProfileCredentialsProvider;  
import com.amazonaws.regions.Regions;  
import com.amazonaws.services.s3.AmazonS3;  
import com.amazonaws.services.s3.AmazonS3ClientBuilder;  
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;  
import com.amazonaws.services.s3.model.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;  
  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.List;  
  
public class DeleteMultipleObjectsVersionEnabledBucket {  
    private static AmazonS3 S3_CLIENT;  
    private static String VERSIONED_BUCKET_NAME;
```

```
public static void main(String[] args) throws IOException {
    Regions clientRegion = Regions.DEFAULT_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");
}
}
```

.NET

O AWS SDK for .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 do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
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 são exibidos 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 do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;  
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;
}
}
```

PHP

Estes exemplos mostram como usar as classes da versão 3 do AWS SDK for PHP para excluir vários objetos de buckets do Amazon S3 com e sem versionamento. Para obter mais informações sobre versionamento, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

Os exemplos consideram que você já está seguindo as instruções para [Usar o AWS SDK for PHP e executar exemplos do PHP \(p. 1268\)](#) e tem o AWS SDK for 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 \(p. 1269\)](#).

```
<?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
]);

// 3. Delete the objects.
foreach ($keys['Contents'] as $key)
{
    $s3->deleteObjects([
        'Bucket'  => $bucket,
        'Delete'  => [
            'Objects' => [
                [
                    'Key' => $key['Key']
                ]
            ]
        ]
    ]);
}
```

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 \(p. 1269\)](#).

```
<?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,
    'VersioningConfiguration' => [
        '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]);

// 4. Delete the object versions.
$deletedResults = 'The following objects were deleted successfully:' . PHP_EOL;
$deleted = false;
$errorResults = 'The following objects could not be deleted:' . PHP_EOL;
$errors = false;

foreach ($versions['Versions'] as $version)
{
    $result = $s3->deleteObjects([
        'Bucket'  => $bucket,
        'Delete'  => [
            'Objects' => [
                [
                    'Key'    => $version['Key'],
                    'VersionId' => $version['VersionId']
                ]
            ]
        ]
    ]);

    if (isset($result['Deleted']))
    {
        $deleted = true;

        $deletedResults .= "Key: {$result['Deleted'][0]['Key']}, " .
                           "VersionId: {$result['Deleted'][0]['VersionId']} " . PHP_EOL;
    }

    if (isset($result['Errors']))
    {
        $errors = true;

        $errorResults .= "Key: {$result['Errors'][0]['Key']}, " .
                           "VersionId: {$result['Errors'][0]['VersionId']}, " .

```

```
        "Message: {$result['Errors'][0]['Message']}" . PHP_EOL;
    }

    if ($deleted)
    {
        echo $deletedResults;
    }

    if ($errors)
    {
        echo $errorResults;
    }

    // 5. Suspend object versioning for the bucket.
    $s3->putBucketVersioning([
        'Bucket' => $bucket,
        'VersioningConfiguration' => [
            'Status' => 'Suspended'
        ]
    ]);
}
```

Uso dos REST API

Você pode usar os AWS SDKs para excluir vários objetos usando a API Multi-Object Delete. Contudo, se o seu aplicativo exigir, você pode enviar solicitações REST diretamente.

Para obter mais informações, consulte [Excluir vários objetos](#) na Referência da API do Amazon Simple Storage Service.

Organizar, listar e trabalhar com seus objetos

No Amazon S3, você pode usar prefixos para organizar seu armazenamento. Um prefixo é um agrupamento lógico dos objetos em um bucket. O valor do prefixo é semelhante a um nome de diretório que permite que você armazene dados semelhantes em um bucket no mesmo diretório. Quando você faz upload de objetos de maneira programática, é possível usar prefixos para organizar seus dados.

No console do Amazon S3, os prefixos são chamados de pastas. Você pode exibir todos os objetos e pastas no console do S3 navegando até um bucket. Você também pode exibir informações sobre cada objeto incluindo propriedades do objeto.

Para obter mais informações sobre como listar e organizar seus dados no Amazon S3, consulte os tópicos a seguir.

Tópicos

- [Organizar objetos usando prefixos \(p. 258\)](#)
- [Listar chaves de objeto programaticamente \(p. 260\)](#)
- [Organizar objetos no console do Amazon S3 usando pastas \(p. 269\)](#)
- [Exibir uma visão geral do objeto no console do Amazon S3 \(p. 271\)](#)
- [Exibir propriedades do objeto no console do Amazon S3 \(p. 272\)](#)

Organizar objetos usando prefixos

Você pode usar prefixos para organizar os dados armazenados nos buckets do Amazon S3. Um prefixo é uma string no início do nome da chave do objeto. Um prefixo pode ter qualquer comprimento, sujeito ao

comprimento máximo do nome da chave do objeto (1.024 bytes). Você pode pensar nos prefixos como uma forma de organizar seus dados de forma semelhante aos diretórios. No entanto, os prefixos não são diretórios.

Pesquisar por prefixo limita os resultados a somente essas chaves que começam com o prefixo especificado. O delimitador faz com que uma operação de lista acumule todas as chaves que compartilham um prefixo comum em um único resultado da lista de resumo.

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. Você pode usar outro caractere como um delimitador. Não há nada de especial no caractere barra (/), mas esse é um delimitador de prefixo muito comum. 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 usar a barra (/) como delimitador. Os seguintes exemplos usam uma barra (/) como delimitador.

- Europa/França/Nova Aquitânia/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 os dados. Por exemplo, para listar todos os estados nos EUA, defina `Delimiter='/'` e `Prefix='North America/USA/'`. Para listar todas as províncias no Canadá para as quais você tem dados, defina `Delimiter='/'` e `Prefix='North America/Canada/'`.

Listar objetos usando prefixos e delimitadores

Se emitir uma solicitação de lista com um delimitador, você poderá pesquisar a hierarquia apenas em um nível, pulando e resumindo as chaves aninhadas (possivelmente milhões) em níveis mais profundos. Por exemplo, suponha que você tenha um bucket ([DOC-EXAMPLE-BUCKET](#)) com 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 barra (/). Em resposta, o Simple Storage Service (Amazon S3) retorna a chave do objeto `sample.jpg` porque não contém o caractere delimitador /. Todas as outras chaves contêm o caractere delimitador. O Simple Storage Service (Amazon S3) agrupa essas chaves e retorna um único elemento `CommonPrefixes` com valor de prefixo `photos/`, que é uma substring que vai do início dessas chaves até a primeira ocorrência do delimitador especificado.

Example

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

```
<Name>DOC-EXAMPLE-BUCKET</Name>
<Prefix></Prefix>
<Marker></Marker>
<MaxKeys>1000</MaxKeys>
<Delimiter></Delimiter>
<IsTruncated>false</IsTruncated>
<Contents>
    <Key>sample.jpg</Key>
    <LastModified>2011-07-24T19:39:30.000Z</LastModified>
    <ETag>"d1a7fb5eab1c16cb4f7cf341cf188c3d"</ETag>
    <Size>6</Size>
    <Owner>
        <ID>75cc57f09aa0c8caeab4f8c24e99d10f8e7faeefbf76c078efc7c6caea54ba06a</ID>
        <DisplayName>displayname</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
</Contents>
<CommonPrefixes>
    <Prefix>photos/</Prefix>
</CommonPrefixes>
</ListBucketResult>
```

Para obter mais informações sobre como listar chaves de objeto programaticamente, consulte [Listar chaves de objeto programaticamente \(p. 260\)](#).

Listar chaves de objeto programaticamente

No Amazon S3, as chaves podem ser listadas por prefixo. Você pode escolher um prefixo comum para os nomes das chaves relacionadas e marcar essas chaves com um caractere especial que delimita a hierarquia. Em seguida, você pode usar a operação de lista para selecionar e procurar 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 a enumeração das chaves contidas em um bucket. As chaves são selecionadas para a listagem pelo bucket e pelo prefixo. Por exemplo, considere um bucket chamado “*dictionary*” 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 a SOAP via HTTP está obsoleto, mas o SOAP continua disponível via HTTPS.

Os novos recursos do Amazon S3 não são compatíveis com SOAP. Em vez de SOAP, recomendamos usar a API REST ou os AWS SDKs.

Grupos de chaves que compartilham um prefixo terminado por um delimitador especial podem ser rolados para acima 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 “French/ logical”. 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 [Organizar objetos usando prefixes \(p. 258\)](#).

API REST

Se a sua aplicação 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, consulte [GET Bucket \(listar objetos\) versão 2](#) na Referência da API do Amazon Simple Storage Service.

Implementação eficiente de lista

A performance da lista não é substancialmente afetada pelo número total de chaves no bucket. Também não é afetada pela presença ou ausência dos argumentos `prefix`, `marker`, `maxkeys` ou `delimiter`.

Iterar em 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 é compatível com a 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 solicitações de chaves de lista até receber todas elas. As bibliotecas wrapper do SDK fornecem a mesma paginação.

Exemplos

Os exemplos de código a seguir mostram como listar objetos em um bucket do S3.

.NET

AWS SDK for .NET

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
/// <summary>
/// Shows how to list the objects in an Amazon S3 bucket.
/// </summary>
/// <param name="client">An initialized Amazon S3 client object.</param>
/// <param name="bucketName">The name of the bucket for which to list
/// the contents.</param>
/// <returns>A boolean value indicating the success or failure of the
/// copy operation.</returns>
public static async Task<bool> ListBucketContentsAsync(IAmazonS3 client,
string bucketName)
{
    try
    {
        var request = new ListObjectsV2Request
        {
            BucketName = bucketName,
            MaxKeys = 5,
        };

        Console.WriteLine("-----");
        Console.WriteLine($"Listing the contents of {bucketName}:");
        Console.WriteLine("-----");

        var response = new ListObjectsV2Response();

        do
        {
            response = await client.ListObjectsV2Async(request);

            response.S3Objects
```

```
.ForEach(obj => Console.WriteLine($"{obj.Key,-35} {  
{obj.LastModified.ToShortDateString(),10}{obj.Size,10}}"));

        // If the response is truncated, set the request  
ContinuationToken  
        // from the NextContinuationToken property of the response.  
        request.ContinuationToken = response.NextContinuationToken;  
    }  
    while (response.IsTruncated);  
  
    return true;  
}  
catch (AmazonS3Exception ex)  
{  
    Console.WriteLine($"Error encountered on server.  
Message:{ex.Message}' getting list of objects.");  
    return false;  
}  
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK for .NET.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
bool AwsDoc::S3::ListObjects(const Aws::String& bucketName,  
    const Aws::String& region)  
{  
    Aws::Client::ClientConfiguration config;  
  
    if (!region.empty())  
    {  
        config.region = region;  
    }  
  
    Aws::S3::S3Client s3_client(config);  
  
    Aws::S3::Model::ListObjectsRequest request;  
    request.WithBucket(bucketName);  
  
    auto outcome = s3_client.ListObjects(request);  
  
    if (outcome.IsSuccess())  
    {  
        std::cout << "Objects in bucket '" << bucketName << "' :"  
        << std::endl << std::endl;  
  
        Aws::Vector<Aws::S3::Model::Object> objects =  
            outcome.GetResult().GetContents();  
  
        for (Aws::S3::Model::Object& object : objects)  
        {  
            std::cout << object.GetKey() << std::endl;  
        }  
  
    }  
  
    return true;  
}
```

```
    }
    else
    {
        std::cout << "Error: ListObjects: " <<
        outcome.GetError().GetMessage() << std::endl;

        return false;
    }
}

int main()
{
    Aws::SDKOptions options;
    Aws::InitAPI(options);
    {
        //TODO: Name of a bucket in your account.
        //The bucket must have at least one object in it. One way to achieve
        //this is to configure and run put_object.cpp's executable first.
        const Aws::String bucket_name = "DOC-EXAMPLE-BUCKET";

        //TODO: Set to the AWS Region in which the bucket was created.
        Aws::String region = "us-east-1";

        if (!AwsDoc::S3::ListObjects(bucket_name, region))
        {
            return 1;
        }

        Aws::ShutdownAPI(options);

        return 0;
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK for C++.

Go

SDK for Go V2

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
// List objects in the bucket.
// n.b. object keys in Amazon S3 do not begin with '/'. You do not need to lead
// your
// prefix with it.
fmt.Println("Listing the objects in the bucket:")
listObjsResponse, err := client.ListObjectsV2(context.TODO(),
&s3.ListObjectsV2Input{
    Bucket: aws.String(name),
    Prefix: aws.String(""),
})

if err != nil {
    panic("Couldn't list bucket contents")
}

for _, object := range listObjsResponse.Contents {
    fmt.Printf("%s (%d bytes, class %v) \n", *object.Key, object.Size,
object.StorageClass)
```

```
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK for Go.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static void listBucketObjects(S3Client s3, String bucketName ) {  
  
    try {  
        ListObjectsRequest listObjects = ListObjectsRequest  
            .builder()  
            .bucket(bucketName)  
            .build();  
  
        ListObjectsResponse res = s3.listObjects(listObjects);  
        List<S3Object> objects = res.contents();  
        for (S3Object myValue : objects) {  
            System.out.print("\n The name of the key is " + myValue.key());  
            System.out.print("\n The object is " + calKb(myValue.size()) + "  
KBs");  
            System.out.print("\n The owner is " + myValue.owner());  
        }  
  
    } catch (S3Exception e) {  
        System.err.println(e.awsErrorDetails().errorMessage());  
        System.exit(1);  
    }  
}  
  
//convert bytes to kbs.  
private static long calKb(Long val) {  
    return val/1024;  
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.  
import { S3Client } from "@aws-sdk/client-s3";  
// Set the AWS Region.  
const REGION = "us-east-1";  
// Create an Amazon S3 service client object.  
const s3Client = new S3Client({ region: REGION });
```

```
export { s3Client };
```

Liste os objetos.

```
// Import required AWS SDK clients and commands for Node.js.
import { ListObjectsCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
Amazon S3 service client module.

// Create the parameters for the bucket
export const bucketParams = { Bucket: "BUCKET_NAME" };

export const run = async () => {
  try {
    const data = await s3Client.send(new ListObjectsCommand(bucketParams));
    console.log("Success", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Liste mil ou mais objetos.

```
// Import required AWS SDK clients and commands for Node.js.
import { ListObjectsCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
Amazon S3 service client module.

// Create the parameters for the bucket
export const bucketParams = { Bucket: "BUCKET_NAME" };

export async function run() {
  // Declare truncated as a flag that the while loop is based on.
  let truncated = true;
  // Declare a variable to which the key of the last element is assigned to in the
  response.
  let pageMarker;
  // while loop that runs until 'response.truncated' is false.
  while (truncated) {
    try {
      const response = await s3Client.send(new ListObjectsCommand(bucketParams));
      // return response; //For unit tests
      response.Contents.forEach((item) => {
        console.log(item.Key);
      });
      // Log the key of every item in the response to standard output.
      truncated = response.IsTruncated;
      // If truncated is true, assign the key of the last element in the response
      to the pageMarker variable.
      if (truncated) {
        pageMarker = response.Contents.slice(-1)[0].Key;
        // Assign the pageMarker value to bucketParams so that the next iteration
        starts from the new pageMarker.
        bucketParams.Marker = pageMarker;
      }
      // At end of the list, response.truncated is false, and the function exits
      the while loop.
    } catch (err) {
      console.log("Error", err);
    }
  }
}
```

```
        truncated = false;
    }
}
run();
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

Note

Essa documentação é de pré-lançamento para um recurso em versão de pré-visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
suspend fun listBucketObjects(bucketName: String) {

    val request = ListObjectsRequest {
        bucket = bucketName
    }

    S3Client { region = "us-east-1" }.use { s3 ->

        val response = s3.listObjects(request)
        response.contents?.forEach { myObject ->
            println("The name of the key is ${myObject.key}")
            println("The object is ${calKb(myObject.size)} KBs")
            println("The owner is ${myObject.owner}")
        }
    }
}

private fun calKb(intValue: Long): Long {
    return intValue / 1024
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK para Kotlin.

PHP

SDK for PHP

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Liste objetos em um bucket.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2', 'version' => 'latest']);
```

```
try {
    $contents = $s3client->listObjects([
        'Bucket' => $bucket_name,
    ]);
    echo "The contents of your bucket are: \n";
    foreach ($contents['Contents'] as $content) {
        echo $content['Key'] . "\n";
    }
} catch (Exception $exception) {
    echo "Failed to list objects in $bucket_name with error: " . $exception-
>getMessage();
    exit("Please fix error with listing objects before continuing.");
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK for PHP.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""
    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource in
Boto3
                           that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

    @staticmethod
    def list(bucket, prefix=None):
        """
        Lists the objects in a bucket, optionally filtered by a prefix.

        :param bucket: The bucket to query. This is a Boto3 Bucket resource.
        :param prefix: When specified, only objects that start with this prefix are
listed.
        :return: The list of objects.
        """
        try:
            if not prefix:
                objects = list(bucket.objects.all())
            else:
                objects = list(bucket.objects.filter(Prefix=prefix))
                logger.info("Got objects %s from bucket '%s'",
                            [o.key for o in objects], bucket.name)
        except ClientError:
            logger.exception("Couldn't get objects for bucket '%s'.", bucket.name)
            raise
        else:
            return objects
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK para Python (Boto3).

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
require "aws-sdk-s3"

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

  # @param bucket [Aws::S3::Bucket] An existing Amazon S3 bucket.
  def initialize(bucket)
    @bucket = bucket
  end

  # Lists object in a bucket.
  #
  # @param max_objects [Integer] The maximum number of objects to list.
  # @return [Integer] The number of objects listed.
  def list_objects(max_objects)
    count = 0
    puts "The objects in #{@bucket.name} are:"
    @bucket.objects.each do |obj|
      puts "\t#{obj.key}"
      count += 1
      break if count == max_objects
    end
    count
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't list objects in bucket #{@bucket.name}. Here's why: #{e.message}"
    0
  end
end

def run_demo
  bucket_name = "doc-example-bucket"

  wrapper = BucketListObjectsWrapper.new(Aws::S3::Bucket.new(bucket_name))
  count = wrapper.list_objects(25)
  puts "Listed #{count} objects."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK for Ruby.

Rust

SDK for Rust

Note

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
pub async fn list_objects(client: &Client, bucket_name: &str) -> Result<(), Error> {
    let objects = client.list_objects_v2().bucket(bucket_name).send().await?;
    println!("Objects in bucket:");
    for obj in objects.contents().unwrap_or_default() {
        println!("{}:{}", obj.key().unwrap());
    }
    Ok(())
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK para Rust.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public func listBucketFiles(bucket: String) async throws -> [String] {
    let input = ListObjectsV2Input(
        bucket: bucket
    )
    let output = try await client.listObjectsV2(input: input)
    var names: [String] = []

    guard let objList = output.contents else {
        return []
    }

    for obj in objList {
        if let objName = obj.key {
            names.append(objName)
        }
    }

    return names
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK para Swift.

Organizar objetos no console do Amazon S3 usando pastas

Nos buckets e objetos do Amazon S3 estão os recursos primários e os objetos são armazenados em buckets. O Amazon S3 tem uma estrutura plana em vez de uma hierarquia como você normalmente veria em um sistema de arquivos. No entanto, para fins de simplicidade organizacional, o console do Amazon S3 oferece suporte ao conceito de pastas como meio de agrupar objetos. Ele faz isso usando um prefixo

de nome compartilhado para os objetos (ou seja, objetos com nomes que começam com uma string em comum). Os nomes de objeto também são chamados de nomes de chave.

Por exemplo, é possível criar uma pasta no console chamada `photos`, e armazenar um objeto chamado `myphoto.jpg` nela. Em seguida, o objeto é armazenado como o nome da chave `photos/myphoto.jpg`, em que `photos/` é o prefixo.

Estes são mais dois exemplos:

- Se você tiver três objetos no bucket — `logs/date1.txt`, `logs/date2.txt` e `logs/date3.txt` — o console mostrará uma pasta chamada `logs`. Se você abrir a pasta no console, verá três objetos: `date1.txt`, `date2.txt` e `date3.txt`.
- Se você tiver um objeto chamado `photos/2017/example.jpg`, o console mostrará uma pasta chamada `photos` que contém a pasta `2017`. A pasta `2017` conterá o objeto `example.jpg`.

Você pode ter pastas dentro de pastas, mas não buckets dentro de buckets. Você pode carregar e copiar objetos diretamente em uma pasta. As pastas podem ser criadas, excluídas e tornadas públicas, mas não podem ser renomeadas. Os objetos podem ser copiados de uma pasta para outra.

Important

Quando você usa o console do Amazon S3 para criar uma pasta, o Amazon S3 cria um objeto de 0 byte com uma chave definida como o nome da pasta que você forneceu. Por exemplo, se você criar uma pasta chamada `photos` no bucket, o console do Amazon S3 criará um objeto de 0 byte com a chave `photos/`. O console cria esse objeto para ser compatível com a ideia de pastas.

O console do Amazon S3 trata todos os objetos que têm um caractere de barra (/) como o último caractere (final) no nome da chave como uma pasta (por exemplo, `examplekeyname/`). Não é possível carregar um objeto que tenha um nome de chave com caractere de / no final usando o console do Amazon S3. No entanto, é possível carregar objetos nomeados com uma / no final com a API do Amazon S3 usando a AWS CLI, os AWS SDKs ou a API REST.

Os objetos nomeados com uma / no final são exibidos como pastas no console do Amazon S3. O console do Amazon S3 não exibe o conteúdo e os metadados desse objeto. Quando você usa o console para copiar um objeto nomeado com uma / no final, uma pasta é criada no local de destino, mas os dados e os metadados do objeto não são copiados.

Tópicos

- [Criar uma pasta \(p. 270\)](#)
- [Tornar as pastas públicas \(p. 271\)](#)
- [Excluir pastas \(p. 271\)](#)

Criar uma pasta

Esta seção descreve como usar o console do Amazon S3 para criar uma pasta.

Important

Se sua política de bucket não permitir carregar objetos para esse bucket sem as etiquetas, os metadados ou os donatários das listas de controle de acesso (ACL), você não poderá criar uma pasta usando essa configuração. Em vez disso, carregue uma pasta vazia e especifique essas configurações na configuração de upload.

Para criar uma pasta

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, selecione o nome do bucket no qual você deseja criar uma pasta.

3. Se a política de bucket não permitir carregar objetos para esse bucket sem criptografia, você deverá escolher Enable (Habilitar) em Server-side encryption (Criptografia no servidor).
4. Selecione Create folder (Criar pasta).
5. Insira um nome para a pasta (por exemplo, **favorite-pics**). Em seguida, escolha Create folder (Criar pasta).

Tornar as pastas públicas

Recomendamos bloquear todo o acesso público às pastas e aos buckets do Amazon S3, a menos que você exija especificamente uma pasta ou um bucket públicos. Ao tornar uma pasta pública, qualquer pessoa na Internet pode visualizar todos os objetos agrupados nessa pasta.

No console do Amazon S3, você pode tornar uma pasta pública. Também é possível tornar uma pasta pública criando uma política de bucket que limita o acesso por prefixo. Para obter mais informações, consulte [Identity and Access Management no Amazon S3 \(p. 419\)](#).

Warning

Após tornar uma pasta pública no console do Amazon S3, não será possível torná-la privada novamente. Em vez disso, é necessário definir permissões em cada objeto individual na pasta pública para que o objeto não tenha acesso público. Para obter mais informações, consulte [Configurar ACLs \(p. 596\)](#).

Excluir pastas

Esta seção explica como usar o console do Amazon S3 para excluir pastas de um bucket do S3.

Para obter informações sobre os recursos e a definição de preço do Amazon S3, consulte [Amazon S3](#).

Para excluir pastas de um bucket do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket do qual você deseja excluir pastas.
3. Na lista Objects (Objetos), marque a caixa de seleção ao lado das pastas e objetos que você deseja excluir.
4. Escolha Delete (Excluir).
5. Na página Delete objects (Excluir objetos) verifique se os nomes das pastas selecionadas para exclusão estão listados.
6. Na caixa Delete objects (Excluir objetos), insira **delete** e escolha Delete objects (Excluir objetos).

Warning

Essa ação exclui todos os objetos especificados. Ao excluir pastas, aguarde a conclusão da ação de exclusão antes de adicionar objetos novos à pasta. Caso contrário, os objetos novos também podem ser excluídos.

Exibir uma visão geral do objeto no console do Amazon S3

Você pode usar o console do Amazon S3 para exibir uma visão geral de um objeto. A visão geral do objeto no console fornece todas as informações essenciais de um objeto em um só lugar.

Para abrir o painel de visão geral de um objeto

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
 2. Na lista Buckets, escolha o nome do bucket que contém o objeto.
 3. Na lista Objects (Objetos), escolha o nome do objeto do qual deseja uma visão geral.
- A visão geral do objeto é aberta.
4. Para fazer download do objeto, escolha Object actions (Ações de objeto)e, em seguida, escolha Download. Para copiar o caminho do objeto para a área de transferência, em Object URL (URL do objeto), escolha o URL.
 5. Se o versionamento estiver habilitado no bucket, escolha Versions (Versões) para listar as versões do objeto.
 - Para fazer download de uma versão do objeto, marque a caixa de seleção ao lado do ID da versão, escolha Actions (Ações) e, em seguida, escolha Download.
 - Para excluir uma versão de objeto, marque a caixa de seleção ao lado do ID da versão e escolha Delete (Excluir).

Important

Você pode cancelar a exclusão de um objeto somente se ele foi excluído como a versão mais recente (atual). Não é possível cancelar a exclusão de uma versão anterior de um objeto que foi excluído.

Exibir propriedades do objeto no console do Amazon S3

Você pode usar o console do Amazon S3 para exibir as propriedades de um objeto, como a classe de armazenamento, as configurações de criptografia, as etiquetas e os metadados.

Para visualizar as propriedades de um objeto

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket que contém o objeto.
3. Na lista Objects (Objetos), escolha o nome do objeto do qual você deseja visualizar as propriedades.

A Object overview (Visão geral do objeto) de seu objeto é aberta. Você pode rolar para baixo para exibir as propriedades do objeto.

4. Na página Object overview (Visão geral do objeto), você pode configurar as propriedades a seguir para o objeto.

Note

Se você alterar as propriedades Storage Class (Classe de armazenamento), Encryption (Criptografia) ou Metadata (Metadados), um novo objeto será criado para substituir o antigo.

Se o versionamento do S3 estiver habilitado, uma nova versão do objeto será criada e o objeto existente se tornará uma versão mais antiga. A função que altera a propriedade também se torna o proprietário do novo objeto ou (versão do objeto).

- a. Storage class (Classe de armazenamento) – cada objeto no Amazon S3 tem uma classe de armazenamento associada a ela. A classe de armazenamento que você decide usar depende

da frequência com que acessa o objeto. A classe de armazenamento padrão de objetos do S3 é STANDARD. Escolha qual classe de armazenamento usar ao fazer upload de um objeto. Para obter mais informações sobre classes de armazenamento, consulte [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#).

Para alterar a classe de armazenamento depois de fazer upload de um objeto, escolha Storage class (Classe de armazenamento). Escolha a classe de armazenamento desejada e Save (Salvar).

- b. Server-side encryption settings (Configurações de criptografia no lado do servidor): você pode usar a criptografia no lado do servidor para criptografar seus objetos do S3. Para obter mais informações, consulte [Especificando criptografia no lado do servidor com o AWS KMS \(SSE-KMS\) \(p. 364\)](#) ou [Especificação de criptografia do Amazon S3 \(p. 380\)](#).
- c. Metadata (Metadados) – cada objeto no Amazon S3 tem um conjunto de pares de nome–valor que representa seus metadados. Para obter informações sobre como adicionar metadados a um objeto do S3, consulte [Editar metadados de objeto no console do Amazon S3 \(p. 169\)](#).
- d. Tags: você categoriza o armazenamento adicionando tags a um objeto do S3. Para obter mais informações, consulte [Categorizando seu armazenamento usando tags \(p. 878\)](#).
- e. Object lock legal hold and retention (Retenção legal e retenção do Bloqueio de objetos): você pode impedir que um objeto seja excluído. Para obter mais informações, consulte [Usar o bloqueio de objetos do S3 \(p. 722\)](#).

Usar pre-signed URLs

Por padrão, todos os objetos e buckets são privados. No entanto, você pode usar um URL pré-assinado para compartilhar objetos de forma opcional ou permitir que seus clientes/usuários carreguem objetos em buckets sem credenciais ou permissões de segurança da AWS.

É possível usar URLs pré-assinados para gerar um URL que pode ser usado para acessar seus buckets do S3. Ao criar um pre-signed URL, você o associa a uma ação específica. É possível compartilhar o URL, e qualquer pessoa com acesso a ele pode executar a ação incorporada no URL como se fosse o usuário de assinatura original. O URL expirará e não funcionará mais quando atingir seu tempo de expiração.

Important

Para todas as regiões lançadas após 20 de março de 2019, se uma solicitação chega no local errado do Amazon S3, o Amazon S3 retorna um erro de solicitação incorreta HTTP 400.

Limitar recursos de pre-signed URLs

Os recursos de um URL pré-assinado são limitados pelas permissões do usuário que o criou. Em essência, os URLs pré-assinados são um token ao portador que concede acesso aos clientes que os têm. Dessa forma, recomendamos que você os proteja adequadamente.

Se quiser restringir o uso de URLs pré-assinados e todo o acesso do S3 a caminhos de rede específicos, você poderá criar políticas do AWS Identity and Access Management (IAM) que exijam um caminho de rede específico. Essas políticas podem ser definidas na entidade principal do IAM que faz a chamada, no bucket do Amazon S3 ou em ambos.

Uma restrição de caminho de rede no principal exige que o usuário dessas credenciais faça solicitações na rede especificada. Uma restrição no bucket limita o acesso a ele apenas às solicitações originadas na rede especificada. Perceba que essas restrições também se aplicam fora do cenário de pre-signed URL.

A condição global do IAM que você usa depende do tipo de endpoint. Se estiver usando o endpoint público para o Amazon S3, use `aws:SourceIp`. Se estiver usando um VPC endpoint para o Amazon S3, use `aws:SourceVpc` ou `aws:SourceVpce`.

A declaração de política do IAM a seguir requer que a entidade principal só acesse a AWS do intervalo de rede especificado. Com essa declaração de política em vigor, exige-se que todo acesso seja originário desse intervalo. Isso inclui o caso de alguém usando um pre-signed URL para o S3.

```
{  
    "Sid": "NetworkRestrictionForIAMPrincipal",  
    "Effect": "Deny",  
    "Action": "",  
    "Resource": "",  
    "Condition": {  
        "NotIpAddressIfExists": { "aws:SourceIp": "IP-address" },  
        "BoolIfExists": { "aws:ViaAWSService": "false" }  
    }  
}
```

Quem pode criar um URL pré-assinado

Qualquer um com credenciais de segurança válidas pode criar um pre-signed URL. Mas para que alguém acesse com êxito um objeto, o URL pré-assinado deve ter sido criado por alguém que tenha permissão para executar a operação na qual o URL pré-assinado é baseado.

As credenciais que podem ser usadas para criar um URL pré-assinado são as seguintes:

- Perfil da instância do IAM: válido por até 6 horas.
- AWS Security Token Service: válido por até 36 horas quando assinado com credenciais permanentes, como as credenciais do usuário raiz da Conta da AWS ou um usuário do IAM.
- Usuário do IAM: válido por até 7 dias quando o AWS Signature versão 4 é usado.

Para criar um URL pré-assinado que seja válido por até 7 dias, primeiro designe credenciais de usuário do IAM (a chave de acesso e a chave secreta) para o SDK que você está usando. Depois, gere um URL pré-assinado usando o AWS Signature versão 4.

Note

- Se você criou um URL pré-assinado usando um token temporário, o URL expirará quando o token expirar, mesmo que o URL tenha sido criado com uma hora de validade posterior.
- Como os URLs pré-assinados concedem acesso aos buckets do Amazon S3 a quem tem o URL, recomendamos protegê-los adequadamente.

Quando o Amazon S3 confere a data e hora de validade de um URL pré-assinado?

O Amazon S3 confere a data e hora de validade de um URL assinado no momento da solicitação HTTP. Por exemplo, se um cliente começar a baixar um arquivo grande imediatamente antes da hora de validade, o download será concluído mesmo se a hora de validade passar durante o download. Se a conexão cair e o cliente tentar reiniciar o download após a hora de validade, ocorrerá uma falha no download.

Para obter mais informações sobre como usar um URL pré-assinado para compartilhar ou carregar objetos, consulte os tópicos a seguir.

Tópicos

- [Compartilhar objetos usando URLs pré-assinados \(p. 275\)](#)
- [Gerar um URL pré-assinado para carregar um objeto \(p. 278\)](#)

- Excluir um objeto usando um URL pré-assinado com o AWS SDK for JavaScript (p. 287)

Compartilhar objetos usando URLs pré-assinados

Por padrão, todos os objetos do S3 são privados. Somente o proprietário do objeto tem permissão para acessá-lo. 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 criar um URL pré-assinado para o objeto, você deverá fornecer suas credenciais de segurança, especificar um nome de bucket, uma chave de objeto, um método HTTP (GET para baixar o objeto) e uma data e hora de validade. Os pre-signed URLs são válidos apenas pela duração especificada. Se você criou um pre-signed URL usando um token temporário, o URL expirará quando o token expirar, mesmo se o URL tiver sido criado com uma hora de expiração posterior.

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. Como os URLs pré-assinados concedem acesso aos buckets do Amazon S3 a quem tem o URL, recomendamos protegê-los adequadamente. Para obter mais detalhes sobre como proteger pre-signed URLs, consulte [Limitar recursos de pre-signed URLs \(p. 273\)](#).

Para obter mais informações sobre quem pode criar um URL pré-assinado, consulte [Quem pode criar um URL pré-assinado \(p. 274\)](#).

Gerar um URL pré-assinado para compartilhar um objeto

Você pode gerar um URL pré-assinado para um objeto sem escrever nenhum código, usando o console do S3 ou o AWS Explorer for Visual Studio. Você também pode gerar um URL pré-assinado de modo programático, usando os AWS SDK para Java, .NET, [Ruby](#), [PHP](#), [Node.js](#), [Python](#) e [Go](#).

Uso do console do S3

Você pode usar o AWS Management Console para gerar um URL pré-assinado para um objeto seguindo estas etapas.

Note

No console do Amazon S3, o tempo máximo de validade de um URL pré-assinado é de 12 horas a partir do momento da criação.

Para gerar um URL pré-assinado usando o AWS Management Console

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, selecione o nome do bucket que contém os objetos para os quais você deseja obter um URL pré-assinado.
3. Na lista Objects (Objetos), selecione o objeto para o qual deseja criar um URL pré-assinado.
4. No menu Actions (Ações), escolha Share with a presigned URL (Compartilhar com um URL pré-assinado).
5. Especifique por quanto tempo deseja que o URL pré-assinado seja válido.
6. Escolha Create presigned URL (Criar URL pré-assinado).
7. Quando uma confirmação é exibida, o URL é copiado automaticamente para a área de transferência. Você verá um botão para copiar o URL pré-assinado se precisar copiá-lo novamente.

Usar o AWS Explorer for Visual Studio

Se você estiver usando o Visual Studio, poderá gerar um URL pré-assinado 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](#).

Note

Com o AWS Explorer for Visual Studio, o tempo máximo de validade de um URL pré-assinado é de sete dias a partir do momento da criação.

Para obter instruções sobre como instalar o AWS Explorer, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers \(p. 1258\)](#).

Uso de AWS SDKs

O exemplo a seguir gera um URL pré-assinado que você pode compartilhar para recuperar um objeto.

Note

Com os AWS SDKs, o tempo máximo de validade de um URL pré-assinado é de sete dias a partir do momento da criação.

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 [Compartilhar objetos usando URLs pré-assinados \(p. 275\)](#).

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

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

import java.io.IOException;
import java.net.URL;
import java.time.Instant;

public class GeneratePresignedURL {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_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();
            expiration.setTime(expiration.getTime() + (1000 * 60 * 60));
            GeneratePresignedUrlRequest generatePresignedUrlRequest =
                new GeneratePresignedUrlRequest(bucketName, objectKey)
                    .withMethod(HttpMethod.GET)
                    .withExpiration(expiration);
            URL url = s3Client.generatePresignedUrl(generatePresignedUrlRequest);
            System.out.println("Presigned URL: " + url);
        } catch (AmazonServiceException ase) {
            System.out.println("Caught an AmazonServiceException, which means your request failed");
            System.out.println("HTTP Status Code: " + ase.getStatusCode());
            System.out.println("Error Code: " + ase.getErrorCode());
            System.out.println("Error Message: " + ase.getMessage());
            System.out.println("Request ID: " + ase.getRequestId());
        }
    }
}
```

```
long expTimeMillis = Instant.now().toEpochMilli();
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();
}
}
```

.NET

Example

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
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 how long the presigned URL lasts, in hours
        private const double timeoutDuration = 12;
        // 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(timeoutDuration);
        }
        static string GeneratePreSignedURL(double duration)
        {
            string urlString = "";
            try
            {
                GetPreSignedUrlRequest request1 = new GetPreSignedUrlRequest
                {
                    BucketName = bucketName,
                    Key = objectKey,
```

```
        Expires = DateTime.UtcNow.AddHours(duration)
    };
    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;
}
}
```

PHP

Para obter informações sobre o uso do AWS SDK for PHP versão 3 para gerar um URL pré-assinado, consulte [URL pré-assinado do Amazon S3 com o AWS SDK for PHP versão 3](#) no Guia do desenvolvedor do AWS SDK for PHP.

Python

Gere uma URL pré-assinada para compartilhar um objeto usando o SDK for Python (Boto3). Por exemplo, use um cliente Boto3 e a `generate_presigned_url` função para gerar uma URL pré-assinada que obtém um objeto.

```
import boto3
url = boto3.client('s3').generate_presigned_url(
    ClientMethod='get_object',
    Params={'Bucket': 'BUCKET_NAME', 'Key': 'OBJECT_KEY'},
    ExpiresIn=3600)
```

Para obter mais informações sobre como usar o SDK for Python (Boto3) para gerar um URL pré-assinado, consulte [Python](#) na Referência da API do AWS SDK for Python (Boto).

Uso do AWS CLI

Para gerar um URL pré-assinado para compartilhar um objeto usando o AWS CLI, consulte [presign](#) na Referência de comando da AWS CLI.

Note

Com a AWS CLI, o tempo máximo de validade de um URL pré-assinado é de sete dias a partir do momento da criação.

Gerar um URL pré-assinado para carregar um objeto

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 URLs pré-assinados 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. Ou seja, é necessário iniciar a ação antes da data e a hora de expiração.

Se a ação consistir em várias etapas, como um carregamento fracionado, todas as etapas devem ser iniciadas dentro da validade. Do contrário, você receberá um erro quando o Simple Storage Service (Amazon S3) tentar iniciar uma etapa com um URL expirado.

É possível usar o pre-signed URL várias vezes, até a data e a hora de expiração.

Acesso a URLs pré-assinados

Como os URLs pré-assinados concedem acesso aos buckets do Amazon S3 a quem tem o URL, recomendamos protegê-los adequadamente. Para obter mais informações sobre como proteger URLs pré-assinados, consulte [Limitar recursos de pre-signed URLs \(p. 273\)](#).

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. Para obter mais informações, consulte [Quem pode criar um URL pré-assinado \(p. 274\)](#).

Gerar um URL pré-assinado para carregar objetos

Você pode gerar um URL pré-assinado de modo programático usando os AWS SDKs para .NET, Java Ruby, [JavaScript](#), PHP e [Python](#).

Você pode usar o AWS SDK para gerar um URL pré-assinado que você ou qualquer pessoa a quem der o URL podem usar para carregar um objeto para o Simple Storage Service (Amazon S3). Quando você usar o URL para fazer 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 obtido por upload.

Usar o AWS Explorer for Visual Studio

Se estiver usando o Microsoft Visual Studio, você também poderá gerar um URL pré-assinado de objeto sem escrever nenhum código usando o AWS Explorer no AWS Toolkit for Visual Studio. Qualquer pessoa que receber um pre-signed URL válido poderá fazer upload de um objeto programaticamente. Para obter mais informações, consulte [Usar o Amazon S3 do AWS Explorer](#) no Guia do usuário do AWS Toolkit for Visual Studio.

Para obter instruções sobre como instalar o AWS Explorer, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers \(p. 1258\)](#).

Uso de AWS SDKs

Os exemplos a seguir mostram como fazer upload de objetos usando URLs pré-assinados.

Os exemplos de código a seguir mostram como criar um URL pré-assinado para o S3 e carregar um objeto.

Go

SDK for Go V2

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
// Get a presigned URL for the object.  
// In order to get a presigned URL for an object, you must  
// create a Presignclient  
fmt.Println("Create Presign client")  
presignClient := s3.NewPresignClient(&client)  
  
presignResult, err := presignClient.PresignGetObject(context.TODO(),  
&s3.GetObjectInput{  
    Bucket: aws.String(name),  
    Key:    aws.String("path/myfile.jpg"),  
})  
  
if err != nil {  
    panic("Couldn't get presigned URL for GetObject")  
}  
  
fmt.Printf("Presigned URL For object: %s\n", presignResult.URL)
```

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static void signBucket(S3Presigner presigner, String bucketName, String  
keyName) {  
  
    try {  
        PutObjectRequest objectRequest = PutObjectRequest.builder()  
            .bucket(bucketName)  
            .key(keyName)  
            .contentType("text/plain")  
            .build();  
  
        PutObjectPresignRequest presignRequest =  
PutObjectPresignRequest.builder()  
            .signatureDuration(Duration.ofMinutes(10))  
            .putObjectRequest(objectRequest)  
            .build();  
  
        PresignedPutObjectRequest presignedRequest =  
presigner.presignPutObject(presignRequest);  
        String myURL = presignedRequest.url().toString();  
        System.out.println("Presigned URL to upload a file to: " +myURL);  
        System.out.println("Which HTTP method needs to be used when uploading a  
file: " + presignedRequest.httpRequest().method());  
  
        // Upload content to the Amazon S3 bucket by using this URL.  
        URL url = presignedRequest.url();  
  
        // Create the connection and use it to upload the new object by using  
the presigned URL.  
        HttpURLConnection connection = (HttpURLConnection)  
url.openConnection();  
        connection.setDoOutput(true);  
        connection.setRequestProperty("Content-Type", "text/plain");  
        connection.setRequestMethod("PUT");
```

```
        OutputStreamWriter out = new
OutputStreamWriter(connection.getOutputStream());
        out.write("This text was uploaded as an object by using a presigned
URL.");
        out.close();

        connection.getResponseCode();
        System.out.println("HTTP response code is " +
connection.getResponseCode());

    } catch (S3Exception | IOException e) {
        e.printStackTrace();
    }
}
```

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.
import { S3Client } from "@aws-sdk/client-s3";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an Amazon S3 service client object.
const s3Client = new S3Client({ region: REGION });
export { s3Client };
```

Crie um URL pré-assinado para carregar um objeto em um bucket.

```
// Import the required AWS SDK clients and commands for Node.js
import {
    CreateBucketCommand,
    DeleteObjectCommand,
    PutObjectCommand,
    DeleteBucketCommand
} from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
// Amazon S3 service client module.
import { getSignedUrl } from "@aws-sdk/s3-request-presigner";
import fetch from "node-fetch";

// Set parameters
// Create a random name for the Amazon Simple Storage Service (Amazon S3) bucket
// and key
export const bucketParams = {
    Bucket: `test-bucket-${Math.ceil(Math.random() * 10 ** 10)}`,
    Key: `test-object-${Math.ceil(Math.random() * 10 ** 10)}`,
    Body: "BODY"
};
export const run = async () => {
    try {
        // Create an S3 bucket.
        console.log(`Creating bucket ${bucketParams.Bucket}`);
    }
```

```
    await s3Client.send(new CreateBucketCommand({ Bucket: bucketParams.Bucket }));
    console.log(`Waiting for "${bucketParams.Bucket}" bucket creation...`);
} catch (err) {
    console.log("Error creating bucket", err);
}
try {
    // Create a command to put the object in the S3 bucket.
    const command = new PutObjectCommand(bucketParams);
    // Create the presigned URL.
    const signedUrl = await getSignedUrl(s3Client, command, {
        expiresIn: 3600,
    });
    console.log(
        `\nPutting "${bucketParams.Key}" using signedUrl with body
"${bucketParams.Body}" in v3`
    );
    console.log(signedUrl);
    const response = await fetch(signedUrl, {method: 'PUT', body:
bucketParams.Body});
    console.log(
        `\nResponse returned by signed URL: ${await response.text()}\n`
    );
} catch (err) {
    console.log("Error creating presigned URL", err);
}
try {
    // Delete the object.
    console.log(`\nDeleting object "${bucketParams.Key}" from bucket`);
    await s3Client.send(
        new DeleteObjectCommand({ Bucket: bucketParams.Bucket, Key:
bucketParams.Key })
    );
} catch (err) {
    console.log("Error deleting object", err);
}
try {
    // Delete the S3 bucket.
    console.log(`\nDeleting bucket ${bucketParams.Bucket}`);
    await s3Client.send(
        new DeleteBucketCommand({ Bucket: bucketParams.Bucket })
    );
} catch (err) {
    console.log("Error deleting bucket", err);
}
};

run();
```

Crie um URL pré-assinado para baixar um objeto de um bucket.

```
// Import the required AWS SDK clients and commands for Node.js
import {
    CreateBucketCommand,
    PutObjectCommand,
    GetObjectCommand,
    DeleteObjectCommand,
    DeleteBucketCommand
} from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
// Amazon S3 service client module.
import { getSignedUrl } from "@aws-sdk/s3-request-presigner";
const fetch = require("node-fetch");

// Set parameters
// Create a random names for the S3 bucket and key.
```

```
export const bucketParams = {
  Bucket: `test-bucket-${Math.ceil(Math.random() * 10 ** 10)}`,
  Key: `test-object-${Math.ceil(Math.random() * 10 ** 10)}`,
  Body: "BODY"
};

export const run = async () => {
  // Create an S3 bucket.
  try {
    console.log(`Creating bucket ${bucketParams.Bucket}`);
    const data = await s3Client.send(
      new CreateBucketCommand({ Bucket: bucketParams.Bucket })
    );
    return data; // For unit tests.
    console.log(`Waiting for "${bucketParams.Bucket}" bucket creation...\n`);
  } catch (err) {
    console.log("Error creating bucket", err);
  }
  // Put the object in the S3 bucket.
  try {
    console.log(`Putting object "${bucketParams.Key}" in bucket`);
    const data = await s3Client.send(
      new PutObjectCommand({
        Bucket: bucketParams.Bucket,
        Key: bucketParams.Key,
        Body: bucketParams.Body,
      })
    );
    return data; // For unit tests.
  } catch (err) {
    console.log("Error putting object", err);
  }
  // Create a presigned URL.
  try {
    // Create the command.
    const command = new GetObjectCommand(bucketParams);

    // Create the presigned URL.
    const signedUrl = await getSignedUrl(s3Client, command, {
      expiresIn: 3600,
    });
    console.log(
      `\nGetting "${bucketParams.Key}" using signedUrl with body
"${bucketParams.Body}" in v3`
    );
    console.log(signedUrl);
    const response = await fetch(signedUrl);
    console.log(
      `\nResponse returned by signed URL: ${await response.text()}\n`
    );
  } catch (err) {
    console.log("Error creating presigned URL", err);
  }
  // Delete the object.
  try {
    console.log(`\nDeleting object "${bucketParams.Key}" from bucket`);
    const data = await s3Client.send(
      new DeleteObjectCommand({ Bucket: bucketParams.Bucket, Key:
        bucketParams.Key })
    );
    return data; // For unit tests.
  } catch (err) {
    console.log("Error deleting object", err);
  }
  // Delete the S3 bucket.
  try {
```

```
    console.log(`\nDeleting bucket ${bucketParams.Bucket}`);
    const data = await s3Client.send(
      new DeleteBucketCommand({ Bucket: bucketParams.Bucket, Key:
        bucketParams.Key })
    );
    return data; // For unit tests.
  } catch (err) {
    console.log("Error deleting object", err);
  }
};

run();

```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Gere um URL pré-assinado que possa realizar uma ação do S3 por tempo limitado. Use o pacote Requests (Solicitações) para fazer uma solicitação com o URL.

```
import argparse
import logging
import boto3
from botocore.exceptions import ClientError
import requests

logger = logging.getLogger(__name__)

def generate_presigned_url(s3_client, client_method, method_parameters,
                           expires_in):
    """
    Generate a presigned Amazon S3 URL that can be used to perform an action.

    :param s3_client: A Boto3 Amazon S3 client.
    :param client_method: The name of the client method that the URL performs.
    :param method_parameters: The parameters of the specified client method.
    :param expires_in: The number of seconds the presigned URL is valid for.
    :return: The presigned URL.
    """
    try:
        url = s3_client.generate_presigned_url(
            ClientMethod=client_method,
            Params=method_parameters,
            ExpiresIn=expires_in
        )
        logger.info("Got presigned URL: %s", url)
    except ClientError:
        logger.exception(
            "Couldn't get a presigned URL for client method '%s'.", client_method
        )
        raise
    return url

def usage_demo():
    logging.basicConfig(level=logging.INFO, format='%(levelname)s: %(message)s')
```

```
print('*'*88)
print("Welcome to the Amazon S3 presigned URL demo.")
print('*'*88)

parser = argparse.ArgumentParser()
parser.add_argument('bucket', help="The name of the bucket.")
parser.add_argument(
    'key', help="For a GET operation, the key of the object in Amazon S3. For a "
    "PUT operation, the name of a file to upload.")
parser.add_argument(
    'action', choices=('get', 'put'), help="The action to perform.")
args = parser.parse_args()

s3_client = boto3.client('s3')
client_action = 'get_object' if args.action == 'get' else 'put_object'
url = generate_presigned_url(
    s3_client, client_action, {'Bucket': args.bucket, 'Key': args.key}, 1000)

print("Using the Requests package to send a request to the URL.")
response = None
if args.action == 'get':
    response = requests.get(url)
elif args.action == 'put':
    print("Putting data to the URL.")
    try:
        with open(args.key, 'r') as object_file:
            object_text = object_file.read()
        response = requests.put(url, data=object_text)
    except FileNotFoundError:
        print(f"Couldn't find {args.key}. For a PUT operation, the key must be"
              "the "
              f"name of a file that exists on your computer.")

if response is not None:
    print("Got response:")
    print(f"Status: {response.status_code}")
    print(response.text)

print('*'*88)

if __name__ == '__main__':
    usage_demo()
```

Gere uma solicitação POST pré-assinada para carregar um arquivo.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""
    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                    that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def generate_presigned_post(self, object_key, expires_in):
        """
        Generate a presigned Amazon S3 POST request to upload a file.
        A presigned POST can be used for a limited time to let someone without an
        AWS
        account upload a file to a bucket.
```

```
:param object_key: The object key to identify the uploaded object.
:param expires_in: The number of seconds the presigned POST is valid.
:return: A dictionary that contains the URL and form fields that contain
        required access data.
"""
try:
    response = self.bucket.meta.client.generate_presigned_post(
        Bucket=self.bucket.name, Key=object_key, ExpiresIn=expires_in)
    logger.info("Got presigned POST URL: %s", response['url'])
except ClientError:
    logger.exception(
        "Couldn't get a presigned POST URL for bucket '%s' and object
'%s'", self.bucket.name, object_key)
    raise
return response
```

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
require "aws-sdk-s3"
require "net/http"

# Creates a presigned URL that can be used to upload content to an object.
#
# @param bucket [Aws::S3::Bucket] An existing Amazon S3 bucket.
# @param object_key [String] The key to give the uploaded object.
# @return [URI, nil] The parsed URI if successful; otherwise nil.
def get_presigned_url(bucket, object_key)
    url = bucket.object(object_key).presigned_url(:put)
    puts "Created presigned URL: #{url}."
    URI(url)
rescue Aws::Errors::ServiceError => e
    puts "Couldn't create presigned URL for #{bucket.name}:#{object_key}. Here's why:
#{e.message}"
end

def run_demo
    bucket_name = "doc-example-bucket"
    object_key = "my-file.txt"
    object_content = "This is the content of my-file.txt."

    bucket = Aws::S3::Bucket.new(bucket_name)
    presigned_url = get_presigned_url(bucket, object_key)
    return unless presigned_url

    response = Net::HTTP.start(presigned_url.host) do |http|
        http.send_request("PUT", presigned_url.request_uri, object_content,
"content_type" => "")
    end

    case response
    when Net::HTTPSuccess
        puts "Content uploaded!"
    else
```

```
    puts response.value
  end
end

run_demo if $PROGRAM_NAME == __FILE__
```

Excluir um objeto usando um URL pré-assinado com o AWS SDK for JavaScript

Example

O exemplo do AWS SDK for JavaScript a seguir utiliza um URL pré-assinado para excluir um objeto.

Para obter mais informações sobre como usar URLs pré-assinados, consulte [Usar pre-signed URLs \(p. 273\)](#).

```
// Import the required AWS SDK clients and commands for Node.js
import {
  CreateBucketCommand,
  DeleteObjectCommand,
  PutObjectCommand,
  DeleteBucketCommand
} from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an Amazon S3
// service client module.
import { getSignedUrl } from "@aws-sdk/s3-request-presigner";
import fetch from "node-fetch";

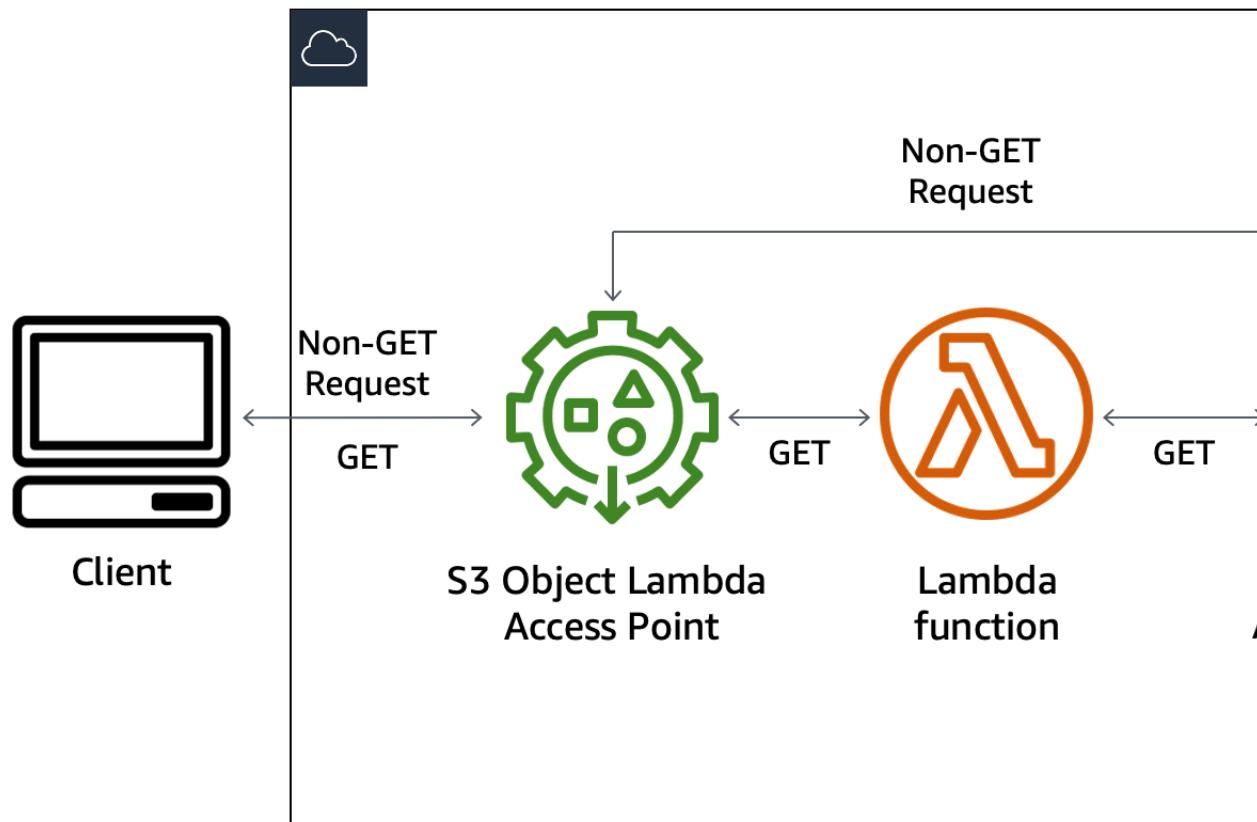
// Set parameters
// Create a random name for the Amazon Simple Storage Service (Amazon S3) bucket and key
export const bucketParams = {
  Bucket: `test-bucket-${Math.ceil(Math.random() * 10 ** 10)}`,
  Key: `test-object-${Math.ceil(Math.random() * 10 ** 10)}`,
  Body: "BODY"
};
export const run = async () => {
  try {
    // Create an S3 bucket.
    console.log(`Creating bucket ${bucketParams.Bucket}`);
    await s3Client.send(new CreateBucketCommand({ Bucket: bucketParams.Bucket }));
    console.log(`Waiting for "${bucketParams.Bucket}" bucket creation...`);
  } catch (err) {
    console.log("Error creating bucket", err);
  }
  try {
    // Create a command to put the object in the S3 bucket.
    const command = new PutObjectCommand(bucketParams);
    // Create the presigned URL.
    const signedUrl = await getSignedUrl(s3Client, command, {
      expiresIn: 3600,
    });
    console.log(
      `\nPutting "${bucketParams.Key}" using signedUrl with body "${bucketParams.Body}" in
v3`;
    );
    console.log(signedUrl);
    const response = await fetch(signedUrl, {method: 'PUT', body: bucketParams.Body});
    console.log(`
```

```
    `\\nResponse returned by signed URL: ${await response.text()}\\n`  
);  
} catch (err) {  
    console.log("Error creating presigned URL", err);  
}  
try {  
    // Delete the object.  
    console.log(`\\nDeleting object "${bucketParams.Key}" from bucket`);  
    await s3Client.send(  
        new DeleteObjectCommand({ Bucket: bucketParams.Bucket, Key: bucketParams.Key })  
    );  
} catch (err) {  
    console.log("Error deleting object", err);  
}  
try {  
    // Delete the S3 bucket.  
    console.log(`\\nDeleting bucket ${bucketParams.Bucket}`);  
    await s3Client.send(  
        new DeleteBucketCommand({ Bucket: bucketParams.Bucket })  
    );  
} catch (err) {  
    console.log("Error deleting bucket", err);  
}  
};  
run();
```

Transformar objetos com o S3 Object Lambda

Com o S3 Object Lambda, você pode adicionar seu próprio código às solicitações GET do Amazon S3 para modificar e processar dados conforme eles são retornados para uma aplicação. Você pode usar o código personalizado para modificar os dados retornados por solicitações S3 GET padrão para filtrar linhas, redimensionar imagens dinamicamente, editar dados confidenciais e muito mais. Desenvolvido pelas funções do AWS Lambda, seu código é executado em uma infraestrutura totalmente gerenciada pela AWS, eliminando a necessidade de criar e armazenar cópias derivadas de seus dados ou de executar proxies, tudo sem a necessidade de alterações nas aplicações.

O S3 Object Lambda usa funções do AWS Lambda para processar automaticamente a saída de uma solicitação GET padrão do S3. O AWS Lambda é um serviço de computação sem servidor que executa código definido pelo cliente sem exigir o gerenciamento de recursos de computação subjacentes. Você pode criar e executar suas próprias funções do Lambda personalizadas, adaptando a transformação de dados para seus casos de uso específicos. Você pode configurar uma função do Lambda e anexá-la a um endpoint de serviço S3 Object Lambda, e o S3 chamará automaticamente sua função. Em seguida, qualquer dado recuperado usando uma solicitação GET do S3 por meio do endpoint do S3 Object Lambda retornará um resultado transformado de volta à aplicação. Todas as outras solicitações serão processadas como normais, conforme ilustrado no diagrama a seguir.



Os tópicos nesta seção descrevem como trabalhar com pontos de acesso do Object Lambda.

Tópicos

- [Criar pontos de acesso do Object Lambda \(p. 290\)](#)
- [Usar pontos de acesso do Amazon S3 Object Lambda \(p. 295\)](#)
- [Usar um template do AWS CloudFormation para automatizar a configuração do S3 Object Lambda \(p. 296\)](#)
- [Configurando políticas do IAM para pontos de acesso do Object Lambda \(p. 299\)](#)
- [Escrever e depurar funções do AWS Lambda para pontos de acesso do S3 Object Lambda \(p. 302\)](#)
- [Usar funções do Lambda criadas pela AWS \(p. 314\)](#)
- [Práticas recomendadas e diretrizes para o S3 Object Lambda \(p. 316\)](#)
- [Considerações sobre segurança para pontos de acesso do S3 Object Lambda \(p. 317\)](#)
- [Tutoriais do S3 Object Lambda \(p. 318\)](#)

Criar pontos de acesso do Object Lambda

Um ponto de acesso do Object Lambda está associado a exatamente um ponto de acesso padrão e, portanto, a um bucket do Amazon S3. Para criar um ponto de acesso do Object Lambda, você precisa dos seguintes recursos:

- Uma política do IAM
- Um bucket do Amazon S3
- Um ponto de acesso do S3 padrão
- Uma função AWS Lambda

As seções a seguir descrevem como criar um ponto de acesso do Object Lambda usando o AWS Management Console e a AWS CLI.

Criar um ponto de acesso do Object Lambda

Para obter informações sobre como criar pontos de acesso Object Lambda usando a API REST, consulte [CreateAccessPointForObjectLambda](#) na Referência da API do Amazon Simple Storage Service.

Uso do console do S3

Para criar um ponto de acesso do Object Lambda usando o console

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação no lado esquerdo do console, escolha Object Lambda access points (Pontos de acesso do Object Lambda).
3. Na página Object Lambda access points (Pontos de acesso do Object Lambda), escolha Create Object Lambda access point (Criar ponto de acesso do Object Lambda).
4. Em Object Lambda access point name (Nome do ponto de acesso do Object Lambda), insira o nome que deseja usar para o ponto de acesso.

Tal como acontece com pontos de acesso padrão, existem regras para nomeação. Para obter mais informações, consulte [Regras para nomear pontos de acesso do Amazon S3 \(p. 325\)](#).

5. Para Supporting Access Point (Ponto de acesso de suporte), insira ou navegue até o ponto de acesso padrão que você deseja usar. O ponto de acesso deve estar na mesma Região da AWS que os objetos que você deseja transformar.
6. Para a função Invoke Lambda, você pode optar por usar uma função pré-criada ou inserir o nome do recurso da Amazon (ARN) de uma função do AWS Lambda em sua Conta da AWS.

Para obter mais informações sobre funções pré-construídas, consulte [Usar funções do Lambda criadas pela AWS \(p. 314\)](#).

7. (Opcional) Para Range and part number (Intervalo e número de peça), você deve habilitar essa opção a fim processar solicitações GET com cabeçalhos de intervalo e número de peça. Selecionar essa opção confirma que sua função do Lambda é capaz de reconhecer e processar essas solicitações. Para obter mais informações sobre cabeçalhos de intervalo e números de peça, consulte [Trabalhar com cabeçalhos Range e partNumber \(p. 311\)](#).
8. (Opcional) Em Payload (Carga), adicione um texto JSON para fornecer informações adicionais à sua função do Lambda. Uma carga é um JSON opcional que você pode fornecer à sua função do Lambda como entrada. Você pode configurar cargas com parâmetros diferentes para pontos de acesso do Object Lambda diferentes que invocam a mesma função do Lambda, estendendo, assim, a flexibilidade da sua função do Lambda.

9. (Opcional) Para Request metrics (Métricas de solicitação), escolha Enable (Habilitar) ou Disable (Desabilitar) para adicionar o monitoramento do Amazon S3 ao seu ponto de acesso do Object Lambda. As métricas de solicitação são cobradas na taxa padrão do CloudWatch.
10. (Opcional) Em Object Lambda access point policy (Política de ponto de acesso do Object Lambda), defina uma política de recursos. Essa política de recurso concede permissão GetObject para o ponto de acesso do Object Lambda especificado.
11. Escolha Create Object Lambda access point (Criar ponto de acesso do Object Lambda).

Uso do AWS CLI

Para criar um ponto de acesso do Object Lambda usando um template do AWS CloudFormation

1. Baixe o pacote de implantação da função s3objectlambda_deployment_package.zip do AWS Lambda em [S3 Object Lambda default configuration](#) (Configuração padrão do S3 Object Lambda).
2. Carregue o pacote para um bucket do Amazon S3 que esteja com o versionamento de objetos habilitado.

```
aws s3api put-object --bucket Amazon S3 bucket name --key  
s3objectlambda_deployment_package.zip --body release/  
s3objectlambda_deployment_package.zip
```

3. Baixe o modelo s3objectlambda_defaultconfig.yaml do AWS CloudFormation em [S3 Object Lambda default configuration](#) (Configuração padrão do S3 Object Lambda).
4. Implante o template em sua conta da AWS.

```
aws cloudformation deploy --template-file s3objectlambda_defaultconfig.yaml \  
--stack-name AWS CloudFormation stack name --parameter-overrides  
ObjectLambdaAccessPointName=Object Lambda access point name \  
SupportingAccessPointName=Amazon S3 access point S3BucketName=Amazon S3 bucket \  
LambdaFunctionS3BucketName=Amazon S3 bucket containing your Lambda package  
LambdaFunctionS3Key=Lambda object key LambdaFunctionS3ObjectVersion=Lambda object  
version LambdaFunctionRuntime=Lambda function runtime --capabilities capability_IAM
```

Para obter mais informações sobre como modificar a configuração padrão do template do AWS CloudFormation, consulte [the section called “Conceitos básicos de um template do AWS CloudFormation” \(p. 296\)](#).

Para criar um ponto de acesso do Object Lambda usando a AWS CLI

O exemplo a seguir cria um ponto de acesso do Object Lambda denominado *my-object-lambda-ap* para o bucket *DOC-EXAMPLE-BUCKET1* na conta *111122223333*. Esse exemplo pressupõe que um ponto de acesso padrão denominado *example-ap* já tenha sido criado. Para obter informações sobre como criar um ponto de acesso padrão, consulte [the section called “Criar pontos de acesso” \(p. 325\)](#).

Este exemplo usa a função pré-criada compress da AWS. Para obter funções do AWS Lambda de exemplo, consulte [the section called “Usar funções criadas pela AWS” \(p. 314\)](#).

1. Crie um bucket. Neste exemplo, usaremos *DOC-EXAMPLE-BUCKET1*. Para obter mais informações sobre a criação de buckets, consulte [the section called “Criação de um bucket” \(p. 129\)](#).
2. Crie um ponto de acesso padrão e anexe-o ao seu bucket. Neste exemplo, usaremos *example-ap*. Para obter informações sobre a criação de pontos de acesso padrão, consulte [the section called “Criar pontos de acesso” \(p. 325\)](#).
3. Crie na conta uma função do Lambda que você deseja usar para transformar o objeto do Simple Storage Service (Amazon S3). Consulte [Uso do Lambda com a AWS CLI](#) no Guia do desenvolvedor do AWS Lambda. Você também pode usar uma função do Lambda pré-criada pela AWS.

- Crie um arquivo de configuração JSON chamado `my-olap-configuration.json`. Nessa configuração, forneça o ponto de acesso de suporte e o ARN da função Lambda criados nas etapas anteriores.

Example

```
{  
    "SupportingAccessPoint" : "arn:aws:s3:us-east-1:111122223333:accesspoint/example-ap",  
    "TransformationConfigurations": [{  
        "Actions" : ["GetObject"],  
        "ContentTransformation" : {  
            "AwsLambda": {  
                "FunctionPayload" : "{\"compressionType\":\"gzip\"}",  
                "FunctionArn" : "arn:aws:lambda:us-east-1:111122223333:function/compress"  
            }  
        }  
    }]  
}
```

- Execute `create-access-point-for-object-lambda` para criar seu ponto de acesso do Object Lambda.

```
aws s3control create-access-point-for-object-lambda --account-id 111122223333 --  
name my-object-lambda-ap --configuration file://my-olap-configuration.json
```

- (Opcional) Crie um arquivo de política JSON chamado `my-olap-policy.json`.

Essa política de recursos concede a permissão de `GetObject` para a conta `444455556666` ao ponto de acesso do Object Lambda especificado.

Example

```
{  
    "Version": "2008-10-17",  
    "Statement": [  
        {  
            "Sid": "Grant account 444455556666 GetObject access",  
            "Effect": "Allow",  
            "Action": "s3-object-lambda:GetObject",  
            "Principal": {  
                "AWS": "arn:aws:iam::444455556666:root"  
            },  
            "Resource": "your-object-lambda-access-point-arn"  
        }  
    ]  
}
```

- (Opcional) Execute `put-access-point-policy-for-object-lambda` para definir sua política de recursos.

```
aws s3control put-access-point-policy-for-object-lambda --account-id 111122223333 --  
name my-object-lambda-ap --policy file://my-olap-policy.json
```

- (Opcional) Especifique uma carga útil.

Uma carga útil é um JSON opcional que você pode fornecer à sua função do AWS Lambda como entrada. Você pode configurar cargas com parâmetros diferentes para pontos de acesso do Object

Lambda diferentes que invocam a mesma função do Lambda, estendendo, assim, a flexibilidade da sua função do Lambda.

A configuração do ponto de acesso do Object Lambda a seguir mostra uma carga com dois parâmetros.

```
{  
    "SupportingAccessPoint": "AccessPointArn",  
    "CloudWatchMetricsEnabled": false,  
    "TransformationConfigurations": [  
        {  
            "Actions": ["GetObject"],  
            "ContentTransformation": {  
                "AwsLambda": {  
                    "FunctionArn": "FunctionArn",  
                    "FunctionPayload": "{\"res-x\": \"100\", \"res-y\": \"100\"}"  
                }  
            }  
        }  
    ]  
}
```

A configuração de ponto de acesso do Object Lambda a seguir mostra uma carga útil com um único parâmetro e com GetObject-Range e GetObject-PartNumber habilitados.

```
{  
    "SupportingAccessPoint": "AccessPointArn",  
    "CloudWatchMetricsEnabled": false,  
    "AllowedFeatures": ["GetObject-Range", "GetObject-PartNumber"],  
    "TransformationConfigurations": [  
        {  
            "Actions": ["GetObject"],  
            "ContentTransformation": {  
                "AwsLambda": {  
                    "FunctionArn": "FunctionArn",  
                    "FunctionPayload": "{\"compression-amount\": \"5\"}"  
                }  
            }  
        }  
    ]  
}
```

Important

Ao usar os pontos de acesso do Object Lambda, certifique-se de que a carga útil não contenha informações sigilosas.

Usar o console e o template do AWS CloudFormation

É possível criar um ponto de acesso do Object Lambda usando a configuração padrão fornecida pelo Amazon S3. Você pode baixar um template do AWS CloudFormation e o código-fonte da função Lambda do [repositório GitHub](#) e implantar esses recursos para configurar um ponto de acesso funcional do Object Lambda.

Para carregar o pacote de implantação da função Lambda

1. Baixe o pacote de implantação da função do AWS Lambda s3objectlambda_deployment_package.zip em [S3 Object Lambda default configuration](#) (Configuração padrão do S3 Object Lambda).
2. Carregue o pacote para um bucket do Amazon S3 que esteja com o versionamento de objetos habilitado.

Para criar um ponto de acesso do Object Lambda usando o console do AWS CloudFormation

1. Baixe o template do AWS CloudFormation s3objectlambda_defaultconfig.yaml em [S3 Object Lambda default configuration](#) (Configuração padrão do S3 Object Lambda).
2. Faça login no Console de Gerenciamento da AWS e abra o console do AWS CloudFormation em <https://console.aws.amazon.com/cloudformation>.
3. Na página do AWS CloudFormation, selecione Create stack (Criar pilha).
4. Em Prerequisite - Prepare template (Pré-requisito - Preparar template), escolha Template is ready (O template está pronto).
5. Em Specify template (Especificar template), escolha Upload a template file (Carregar arquivo de template) e carregue s3objectlambda_defaultconfig.yaml.
6. Escolha Next (Próximo).
7. Na página Specify stack details (Especificar detalhes da pilha), insira um nome para a pilha.
8. Escolha Next (Próximo).
9. Na seção Parameters (Parâmetros), especifique os parâmetros que são definidos no template da pilha:
 - a. Insira o nome do bucket do Amazon S3 para o qual você carregou o pacote de implantação em LambdaFunctionS3BucketName.
 - b. Insira o nome da chave de objeto do Amazon S3 para o qual você carregou o pacote de implantação em LambdaFunctionS3Key.
 - c. Insira a versão do objeto do Amazon S3 no qual você carregou o pacote de implantação para LambdaFunctionS3Key.
 - d. Insira seu tempo de execução preferido para a função do Lambda em LambdaFunctionRuntime. As opções disponíveis são nodejs14.x, python3.9 e java11.
 - e. Insira um nome para o ponto de acesso do Object Lambda em ObjectLambdaAccessPointName.
 - f. Insira o nome do bucket do Amazon S3 que será associado a seu ponto de acesso do Object Lambda em S3BucketName.
 - g. Insira o nome de seu ponto de acesso de suporte em SupportingAccessPointName.

Note

Trata-se de um ponto de acesso associado ao bucket do Amazon S3 que você escolheu na etapa anterior. Caso não tenha nenhum ponto de acesso associado ao bucket do Amazon S3, você poderá configurar o template para criar um selecionando true em CreateNewSupportingAccessPoint.

10. Escolha Next (Próximo).
11. Na página Configure stack options (Configurar opções de pilha), selecione Next (Próximo).

Para obter mais informações sobre as configurações opcionais nesta página, consulte [Configurar opções de pilha do AWS CloudFormation](#) no Guia do usuário do AWS CloudFormation.

12. Na página Review (Revisar), selecione Create stack (Criar pilha).

Para obter mais informações sobre como modificar a configuração padrão do template do AWS CloudFormation, consulte [the section called “Conceitos básicos de um template do AWS CloudFormation” \(p. 296\)](#).

Para obter mais informações sobre como configurar pontos de acesso do Object Lambda usando o AWS CloudFormation sem o template, consulte [AWS::S3ObjectLambda::AccessPoint](#) no Guia do usuário do AWS CloudFormation.

Usar o AWS Cloud Development Kit (AWS CDK)

Para obter mais informações sobre como configurar pontos de acesso do Object Lambda usando o AWS CDK, consulte [a biblioteca de construções do AWS#S3ObjectLambda](#) na Referência da API do AWS Cloud Development Kit (AWS CDK).

Usar pontos de acesso do Amazon S3 Object Lambda

Fazer solicitações via pontos de acesso do S3 Object Lambda funciona da mesma forma que fazer solicitações por meio de outros pontos de acesso. Para obter mais informações sobre como fazer solicitações via ponto de acesso, consulte [Usar pontos de acesso \(p. 329\)](#). Você pode fazer solicitações via pontos de acesso do Object Lambda usando o AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDKs ou API REST do Amazon S3.

Important

Os nomes de recursos da Amazon (ARNs) para pontos de acesso do Object Lambda usam um nome de serviço s3-object-lambda. Assim, os ARNs dos pontos de acesso do Object Lambda começam com arn:aws::s3-object-lambda, em vez do prefixo arn:aws ::s3 usado com outros pontos de acesso.

Como encontrar o ARN do seu ponto de acesso do Object Lambda

Para usar um ponto de acesso do Object Lambda com a AWS CLI ou AWS SDKs, é necessário saber o nome do recurso da Amazon (ARN) do ponto de acesso do Object Lambda. Os exemplos a seguir mostram como encontrar o ARN para um ponto de acesso do Object Lambda usando o AWS Management Console ou a AWS CLI.

AWS Management Console

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Object Lambda access points (Pontos de acesso do Object Lambda).
3. Selecione o botão de opção próximo ao ponto de acesso do Object Lambda cujo ARN deseja copiar.
4. Escolha Copy ARN (Copiar ARN).

AWS CLI

1. Para recuperar uma lista dos pontos de acesso do Object Lambda associados à Conta da AWS, execute o comando a seguir. Antes de executar o comando, substitua o ID da conta **111122223333** pelo ID da sua Conta da AWS.

```
aws s3control list-access-points-for-object-lambda --account-id 111122223333
```

2. Revise a saída do comando para localizar o ARN do ponto de acesso do Object Lambda que deseja usar. A saída do comando anterior será semelhante à saída do exemplo a seguir.

```
{
    "ObjectLambdaAccessPointList": [
        {
            "Name": "my-object-lambda-ap",
            "ObjectLambdaAccessPointArn": "arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/my-object-lambda-ap"
```

```
    },
    ...
}
```

Usar um template do AWS CloudFormation para automatizar a configuração do S3 Object Lambda

Você pode usar um template do AWS CloudFormation para criar rapidamente um ponto de acesso do Object Lambda. O template do AWS CloudFormation cria automaticamente recursos relevantes, configura funções do IAM e configura uma função Lambda que lida automaticamente com solicitações por meio do ponto de acesso do Object Lambda. Com o template do AWS CloudFormation, é possível implementar práticas recomendadas, melhorar seu procedimento de segurança e reduzir erros causados por processos manuais.

Esse [repositório GitHub](#) contém o template do AWS CloudFormation e o código-fonte da função do AWS Lambda. Para obter instruções sobre como usar o template, consulte [Criar pontos de acesso do Object Lambda](#).

A função Lambda fornecida no template não executa nenhuma transformação e retorna seus objetos como estão do bucket do Amazon S3. Você pode clonar a função e adicionar seu próprio código de transformação para modificar e processar dados à medida que eles são retornados a uma aplicação.

Modificar o template

Criar um novo ponto de acesso de suporte

É possível criar um novo ponto de acesso de suporte transmitindo o seguinte parâmetro como parte do comando `aws cloudformation deploy` para criar ao implantar o template.

```
CreateNewSupportingAccessPoint=true
```

Configurar uma carga útil de função

Você pode configurar uma carga útil para fornecer dados complementares à função Lambda transmitindo o seguinte parâmetro como parte do comando `aws cloudformation deploy` ao implantar o template.

```
LambdaFunctionPayload="format=csv"
```

Habilitar o monitoramento do Amazon CloudWatch

Você pode habilitar o monitoramento do CloudWatch transmitindo o seguinte parâmetro como parte do comando `aws cloudformation deploy` ao implantar o template.

```
EnableCloudWatchMonitoring=true
```

Isso habilitará o ponto de acesso do Object Lambda para métricas de solicitação do Amazon S3 e criará dois alarmes do CloudWatch para monitorar erros no lado do cliente e do lado do servidor.

Note

O uso do Amazon CloudWatch incorrerá em custos adicionais. Para obter mais informações sobre métricas de solicitação do Amazon S3, consulte [Monitorar e registrar pontos de acesso](#)

Configurar a simultaneidade provisionada

Para otimizar a latência, é possível configurar a simultaneidade provisionada para a função Lambda com base no ponto de acesso do Object Lambda editando o template para incluir as seguintes linhas em Resources.

```
LambdaFunctionVersion:  
  Type: AWS::Lambda::Version  
  Properties:  
    FunctionName: !Ref LambdaFunction  
    ProvisionedConcurrencyConfig:  
      ProvisionedConcurrentExecutions: Integer
```

Note

Haverá cobranças adicionais pela simultaneidade de provisionamento. Para obter mais informações, consulte [Gerenciar a simultaneidade provisionada do Lambda](#) no Guia do desenvolvedor do AWS Lambda.

Modificar a função Lambda

Alterar valores de cabeçalho

Por padrão, a função Lambda encaminha todos os cabeçalhos, exceto Content-Length e ETag, da solicitação de URL pré-assinado ao cliente do GetObject. Com base em seu código de transformação na função Lambda, é possível escolher enviar novos valores de cabeçalho ao cliente do GetObject.

Você pode atualizar sua função Lambda para enviar novos valores de cabeçalho, transmitindo-os na API `WriteGetObjectResponse`.

Por exemplo, se a função Lambda traduzir texto em objetos do Amazon S3 para um idioma diferente, você poderá transmitir um novo valor no cabeçalho Content-Language. Faça isso modificando a função `writeResponse`, conforme o exemplo abaixo.

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,  
  transformedObject: Buffer,  
  headers: Headers): Promise<PromiseResult<{}, AWSError>> {  
  const { algorithm, digest } = getChecksum(transformedObject);  
  
  return s3Client.writeGetObjectResponse({  
    RequestRoute: requestContext.outputRoute,  
    RequestToken: requestContext.outputToken,  
    Body: transformedObject,  
    Metadata: {  
      'body-checksum-algorithm': algorithm,  
      'body-checksum-digest': digest  
    },  
    ...headers,  
    ContentLanguage: 'my-new-language'  
  }).promise();  
}
```

Para obter uma lista completa dos cabeçalhos compatíveis, consulte [WriteGetObjectResponse](#) na Referência da API do Amazon Simple Storage Service.

Retornar cabeçalhos de metadados

Você pode atualizar sua função Lambda para enviar novos valores de cabeçalho, transmitindo-os na API `WriteGetObjectResponse`.

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,  
  transformedObject: Buffer,  
  headers: Headers): Promise<PromiseResult<{}, AWSError>> {
```

```
const { algorithm, digest } = getChecksum(transformedObject);

return s3Client.writeGetObjectResponse({
  RequestRoute: requestContext.outputRoute,
  RequestToken: requestContext.outputToken,
  Body: transformedObject,
  Metadata: {
    'body-checksum-algorithm': algorithm,
    'body-checksum-digest': digest,
    'my-new-header': 'my-new-value'
  },
  ...headers
}).promise();
}
```

Retornar um novo código de status

É possível retornar um código de status personalizado ao cliente do `GetObject` transmitindo-o na API `WriteGetObjectResponse`.

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,
  transformedObject: Buffer,
  headers: Headers): Promise<PromiseResult<{}, AWSError>> {
  const { algorithm, digest } = getChecksum(transformedObject);

  return s3Client.writeGetObjectResponse({
    RequestRoute: requestContext.outputRoute,
    RequestToken: requestContext.outputToken,
    Body: transformedObject,
    Metadata: {
      'body-checksum-algorithm': algorithm,
      'body-checksum-digest': digest
    },
    ...headers,
    StatusCode: Integer
  }).promise();
}
```

Para obter uma lista completa dos códigos de status compatíveis, consulte [WriteGetObjectResponse](#) na Referência da API do Amazon Simple Storage Service.

Aplicar Range e PartNumber ao objeto de origem

Por padrão, o ponto de acesso do Object Lambda criado pelo template do AWS CloudFormation pode lidar com parâmetros de alcance e número de parte. A função Lambda aplica o intervalo ou o número da parte solicitado ao objeto transformado. Para isso, ele precisa baixar todo o objeto e executar a transformação. Em alguns casos, os intervalos de objetos transformados podem mapear exatamente para seus intervalos de objetos de origem. Isso significa que solicitar o intervalo de bytes A-B no objeto de origem e executar a transformação poderá produzir o mesmo resultado que solicitar o objeto inteiro, executar a transformação e retornar o intervalo de bytes A-B no objeto transformado.

Nesses casos, é possível alterar a implementação da função Lambda para aplicar o intervalo ou o número da parte diretamente no objeto de origem. Isso melhora a latência geral da função e a memória necessárias. Para obter mais informações, consulte [Trabalhar com cabeçalhos Range e partNumber](#).

Desabilitar o manuseio de Range e partNumber

Por padrão, o ponto de acesso do Object Lambda criado pelo template do AWS CloudFormation pode lidar com parâmetros de alcance e número de parte. Se você não precisar, poderá desabilitá-lo removendo as linhas do template a seguir.

```
AllowedFeatures:
```

- GetObject-Range
- GetObject-PartNumber

Transformar objetos grandes

Por padrão, a função Lambda processa todo o objeto na memória antes que ele possa começar a transmitir a resposta ao S3 Object Lambda. É possível modificar a função para transmitir a resposta à medida que ela executa a transformação. Isso ajuda a reduzir a latência de transformação e o tamanho da memória da função Lambda. Para obter um exemplo de implementação, consulte o [Exemplo de conteúdo compactado de transmissão](#).

Configurando políticas do IAM para pontos de acesso do Object Lambda

Os pontos de acesso do Simple Storage Service (Amazon S3) são compatíveis com as políticas de recursos do AWS Identity and Access Management (IAM) que você pode usar para controlar o uso do ponto de acesso por recurso, usuário ou outras condições. Para obter um exemplo detalhado, consulte [Tutorial: Como transformar dados para sua aplicação com o S3 Object Lambda \(p. 30\)](#) e [Tutorial: Detectar e editar dados PII com o S3 Object Lambda e o Amazon Comprehend \(p. 45\)](#).

No caso de uma única Conta da AWS, estes quatro recursos devem ter permissões concedidas para trabalhar com pontos de acesso do Object Lambda:

- O usuário ou a função do IAM
- O bucket e o ponto de acesso padrão associado
- O ponto de acesso do Object Lambda
- A função do AWS Lambda

Solucionar avisos de segurança, erros, avisos gerais e sugestões de AWS Identity and Access Management Access Analyzer antes de salvar a política. O IAM Access Analyzer executa verificações de política para validar sua política em relação à [gramática das políticas](#) e às [práticas recomendadas](#) do IAM. Essas verificações geram descobertas e fornecem recomendações práticas que ajudam a criar políticas que sejam funcionais e estejam em conformidade com as práticas recomendadas de segurança. Para saber mais sobre a validação de políticas usando o IAM Access Analyzer, consulte [Validação de políticas do IAM Access Analyzer](#) no Guia do usuário IAM. Para visualizar uma lista de avisos, erros e sugestões retornados pelo IAM Access Analyzer, consulte [Referência de verificação de políticas do IAM Access Analyzer](#).

Esses exemplos assumem que você tem os seguintes recursos:

- Um bucket do Amazon S3 com o seguinte Nome do recurso da Amazon (ARN):

`arn:aws:s3:::DOC-EXAMPLE-BUCKET1`

O exemplo de política de buckets do S3 a seguir delega o controle de acesso para um bucket aos pontos de acesso do bucket. Essa política permite total acesso a todos os pontos de acesso pertencentes à conta do proprietário do bucket. Assim, todo o acesso a esse bucket é controlado pelas políticas anexadas aos seus pontos de acesso. Os usuários só podem ler no bucket por meio do ponto de acesso do S3, o que significa que as operações só podem ser invocadas por meio dos pontos de acesso. Para obter mais informações, consulte [Delegar controle de acesso a pontos de acesso \(p. 321\)](#).

Example : política de buckets para delegar controle de acesso a pontos de acesso

```
{  
    "Version": "2012-10-17",  
    "Statement" : [
```

```
{
    "Effect": "Allow",
    "Principal" : { "AWS":"account-ARN" },
    "Action" : "*",
    "Resource" : [
        "arn:aws:s3::::DOC-EXAMPLE-BUCKET1",
        "arn:aws:s3::::DOC-EXAMPLE-BUCKET1/*"
    ],
    "Condition": {
        "StringEquals" : { "s3:DataAccessPointAccount" : "Bucket owner's account
ID" }
    }
}
```

- Um ponto de acesso do Amazon S3 Standard nesse bucket com o seguinte ARN:

`arn:aws:s3:us-east-1:111122223333:accesspoint/my-access-point`

- Um ponto de acesso do Object Lambda com o seguinte ARN:

`arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/my-object-lambda-
ap`

- Uma função do AWS Lambda com o seguinte ARN:

`arn:aws:lambda:us-east-1:111122223333:function:MyObjectLambdaFunction`

Note

Se estiver usando uma função do Lambda da sua conta, você deverá incluir a versão da função em sua declaração de política. Por exemplo, `arn:aws:lambda:us-
east-1:111122223333:function:MyObjectLambdaFunction:$LATEST`

A seguinte política do IAM concede a um usuário permissão para a função Lambda, o ponto de acesso padrão e o ponto de acesso do S3 Object Lambda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLambdaInvocation",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:lambda:us-east-1:111122223333:function:MyObjectLambdaFunction:
$LATEST",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "s3-object-lambda.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AllowStandardAccessPointAccess",
            "Action": [
                "s3:Get*",
                "s3>List*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:us-east-1:111122223333:accesspoint/my-access-point/*",
        }
    ]
}
```

```
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:CalledVia": [
                "s3-object-lambda.amazonaws.com"
            ]
        }
    },
{
    "Sid": "AllowObjectLambdaAccess",
    "Action": [
        "s3-object-lambda:Get*",
        "s3-object-lambda>List*"
    ],
    "Effect": "Allow",
    "Resource": "arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/my-object-lambda-ap"
}
]
```

Função de execução do Lambda

Sua função Lambda precisa de permissão para enviar dados para o S3 Object Lambda quando solicitações são feitas para um ponto de acesso do Object Lambda. Isso é fornecido ativando a permissão `s3-object-lambda:WriteGetObjectResponse` na função de execução da sua função Lambda. Crie uma nova função de execução ou atualize uma existente.

Como criar uma função de execução no console do IAM

1. Abra a página [Roles \(Funções\)](#) no console do IAM.
2. Selecione Create role (Criar função).
3. Em Common use cases (Casos de uso comuns), selecione Lambda.
4. Escolha Next: Permissions (Próximo: permissões).
5. Em Attach permissions policies (Anexar políticas de permissões), escolha a política gerenciada da AWS [AmazonS3ObjectLambdaExecutionRolePolicy](#).
6. Escolha Next: Tags (Próximo: tags).
7. Escolha Next: Review (Próximo: revisar).
8. Em Role name (Nome da função), insira **s3-object-lambda-role**.
9. Selecione Create role (Criar função).
10. Aplique a **s3-object-lambda-role** recém-criada como a função de execução de sua função Lambda.

Para obter instruções, consulte [Criar uma função para um serviço da AWS \(console\)](#) no Guia do usuário do IAM.

Para atualizar a função de execução de sua função Lambda

Adicione a instrução a seguir à função de execução usada pela função Lambda.

```
{
    "Sid": "AllowObjectLambdaAccess",
    "Action": [ "s3-object-lambda:WriteGetObjectResponse" ],
    "Effect": "Allow",
    "Resource": "*"
}
```

Para obter informações sobre funções de execução, consulte [Função de execução do Lambda no Guia do desenvolvedor do AWS Lambda](#).

Usar chaves de contexto com pontos de acesso do Object Lambda

Com o S3 Object Lambda, as solicitações GET invocarão automaticamente as funções do Lambda, e todas as outras solicitações serão encaminhadas para o Simple Storage Service (Amazon S3). O S3 Object Lambda avaliará chaves de contexto, como `s3-object-lambda:TlsVersion` ou `s3-object-lambda:AuthType`, relacionadas à conexão ou assinatura da solicitação. Todas as outras chaves de contexto, como `s3:prefix`, são avaliadas pelo Simple Storage Service (Amazon S3).

Escrever e depurar funções do AWS Lambda para pontos de acesso do S3 Object Lambda

Essa seção apresenta detalhes de como escrever e depurar funções do Lambda para uso com pontos de acesso do Object Lambda.

Tópicos

- [Trabalhar com WriteGetObjectResponse \(p. 302\)](#)
- [Depuração do S3 Object Lambda \(p. 310\)](#)
- [Trabalhar com cabeçalhos Range e partNumber \(p. 311\)](#)
- [Formato e uso de contexto de evento \(p. 313\)](#)

Trabalhar com WriteGetObjectResponse

O S3 Object Lambda inclui uma nova operação da API do Amazon S3, `WriteGetObjectResponse`, que permite que a função do Lambda apresente dados e cabeçalhos de resposta personalizados para o autor da chamada `GetObject`. O `WriteGetObjectResponse` oferece ao autor do Lambda amplo controle sobre o código de status, cabeçalhos da resposta e corpo da resposta com base em suas necessidades de processamento. Você pode usar `WriteGetObjectResponse` para responder com todo o objeto transformado, partes do objeto transformado ou outras respostas com base no contexto da sua aplicação. A seção a seguir mostra exemplos exclusivos do uso da operação da API `WriteGetObjectResponse`.

- Exemplo 1: responder com um código de status HTTP 403 (Proibido)
- Exemplo 2: responda com uma imagem transformada
- Exemplo 3: Transmitir conteúdo compactado

Exemplo 1:

Você pode usar `WriteGetObjectResponse` para responder com o código de status HTTP 403 (Proibido) com base no conteúdo do objeto.

Java

```
package com.amazonaws.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;
```

```
import java.io.ByteArrayInputStream;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example1 {

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();

        // Check to see if the request contains all of the necessary information.
        // If it does not, send a 4XX response and a custom error code and message.
        // Otherwise, retrieve the object from S3 and stream it
        // to the client unchanged.
        var tokenIsNotPresent = !
event.getUserRequest().getHeaders().containsKey("requiredToken");
        if (tokenIsNotPresent) {
            s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
                .withRequestRoute(event.outputRoute())
                .withRequestToken(event.outputToken())
                .withStatusCode(403)
                .withContentLength(0L).withInputStream(new ByteArrayInputStream(new
byte[0]))
                .withErrorCode("MissingRequiredToken")
                .withErrorMessage("The required token was not present in the
request."));
            return;
        }

        // Prepare the presigned URL for use and make the request to S3.
        HttpClient httpClient = HttpClient.newBuilder().build();
        var presignedResponse = httpClient.send(
            HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
            HttpResponse.BodyHandlers.ofInputStream());

        // Stream the original bytes back to the caller.
        s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
            .withRequestRoute(event.outputRoute())
            .withRequestToken(event.outputToken())
            .withInputStream(presignedResponse.body()));
    }
}
```

Python

```
import boto3
import requests

def handler(event, context):
    s3 = boto3.client('s3')

    """
    Retrieve the operation context object from the event. This object indicates where
    the WriteGetObjectResponse request
    should be delivered and contains a presigned URL in 'inputS3Url' where we can
    download the requested object from.
    The 'userRequest' object has information related to the user who made this
    'GetObject' request to
    S3 Object Lambda.
    """
    get_context = event["getContext"]
```

```
user_request_headers = event["userRequest"]["headers"]

route = get_context["outputRoute"]
token = get_context["outputToken"]
s3_url = get_context["inputS3Url"]

# Check for the presence of a 'CustomHeader' header and deny or allow based on that
# header.
is_token_present = "SuperSecretToken" in user_request_headers

if is_token_present:
    # If the user presented our custom 'SuperSecretToken' header, we send the
    requested object back to the user.
    response = requests.get(s3_url)
    s3.write_get_object_response(RequestRoute=route, RequestToken=token,
Body=response.content)
else:
    # If the token is not present, we send an error back to the user.
    s3.write_get_object_response(RequestRoute=route, RequestToken=token,
StatusCode=403,
ErrorCode="NoSuperSecretTokenFound", ErrorMessage="The request was not secret
enough.")

# Gracefully exit the Lambda function.
return { 'status_code': 200 }
```

Node.js

```
const { S3 } = require('aws-sdk');
const axios = require('axios').default;

exports.handler = async (event) => {
    const s3 = new S3();

    // Retrieve the operation context object from the event. This object indicates
    where the WriteGetObjectResponse request
    // should be delivered and contains a presigned URL in 'inputS3Url' where we can
    download the requested object from.
    // The 'userRequest' object has information related to the user who made this
    'GetObject' request to S3 Object Lambda.
    const { userRequest, getObjectContext } = event;
    const { outputRoute, outputToken, inputS3Url } = getObjectContext;

    // Check for the presence of a 'CustomHeader' header and deny or allow based on
    that header.
    const isTokenPresent = Object
        .keys(userRequest.headers)
        .includes("SuperSecretToken");

    if (!isTokenPresent) {
        // If the token is not present, we send an error back to the user. The 'await'
        in front of the request
        // indicates that we want to wait for this request to finish sending before
        moving on.
        await s3.writeGetObjectResponse({
            RequestRoute: outputRoute,
            RequestToken: outputToken,
            StatusCode: 403,
            ErrorCode: "NoSuperSecretTokenFound",
            ErrorMessage: "The request was not secret enough.",
        }).promise();
    } else {
        // If the user presented our custom 'SuperSecretToken' header, we send the
        requested object back to the user.
```

```
// Again, note the presence of 'await'.
const presignedResponse = await axios.get(inputs3Url);
await s3.writeGetObjectResponse({
    RequestRoute: outputRoute,
    RequestToken: outputToken,
    Body: presignedResponse.data,
}).promise();
}

// Gracefully exit the Lambda function.
return { statusCode: 200 };
}
```

Exemplo 2:

Ao executar uma transformação de imagem, talvez você ache que precisa de todos os bytes do objeto de origem antes de começar a processá-los. Nesse caso, a solicitação `WriteGetObjectResponse` retorna todo o objeto para a aplicação solicitante em uma chamada.

Java

```
package com.amazonaws.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import javax.imageio.ImageIO;
import java.awt.image.BufferedImage;
import java.awt.Image;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example2 {

    private static final int HEIGHT = 250;
    private static final int WIDTH = 250;

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();
        HttpClient httpClient = HttpClient.newBuilder().build();

        // Prepare the presigned URL for use and make the request to S3.
        var presignedResponse = httpClient.send(
            HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
            HttpResponse.BodyHandlers.ofInputStream());

        // The entire image is loaded into memory here so that we can resize it.
        // Once the resizing is completed, we write the bytes into the body
        // of the WriteGetObjectResponse.
        var originalImage = ImageIO.read(presignedResponse.body());
        var resizingImage = originalImage.getScaledInstance(WIDTH, HEIGHT,
Image.SCALE_DEFAULT);
        var resizedImage = new BufferedImage(WIDTH, HEIGHT,
BufferedImage.TYPE_INT_RGB);
```

```
        resizedImage.createGraphics().drawImage(resizingImage, 0, 0, WIDTH, HEIGHT,
null);

        var baos = new ByteArrayOutputStream();
        ImageIO.write(resizedImage, "png", baos);

        // Stream the bytes back to the caller.
        s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
            .withRequestRoute(event.outputRoute())
            .withRequestToken(event.outputToken())
            .withInputStream(new ByteArrayInputStream(baos.toByteArray())));
    }
}
```

Python

```
import boto3
import requests
import io
from PIL import Image

def handler(event, context):
    """
        Retrieve the operation context object from the event. This object indicates where
        the WriteGetObjectResponse request
        should be delivered and has a presigned URL in 'inputS3Url' where we can download
        the requested object from.
        The 'userRequest' object has information related to the user who made this
        'GetObject' request to
        S3 Object Lambda.
    """
    get_context = event["getObjectContext"]
    route = get_context["outputRoute"]
    token = get_context["outputToken"]
    s3_url = get_context["inputS3Url"]

    """
        In this case, we're resizing '.png' images that are stored in S3 and are accessible
        through the presigned URL
        'inputS3Url'.
    """
    image_request = requests.get(s3_url)
    image = Image.open(io.BytesIO(image_request.content))
    image.thumbnail((256,256), Image.ANTIALIAS)

    transformed = io.BytesIO()
    image.save(transformed, "png")

    # Send the resized image back to the client.
    s3 = boto3.client('s3')
    s3.write_get_object_response(Body=transformed.getvalue(), RequestRoute=route,
    RequestToken=token)

    # Gracefully exit the Lambda function.
    return { 'status_code': 200 }
```

Node.js

```
const { S3 } = require('aws-sdk');
const axios = require('axios').default;
const sharp = require('sharp');
```

```
exports.handler = async (event) => {
    const s3 = new S3();

    // Retrieve the operation context object from event. This object indicates where
    // the WriteGetObjectResponse request
    // should be delivered and has a presigned URL in 'inputS3Url' where we can
    // download the requested object from.
    const { getObjectContext } = event;
    const { outputRoute, outputToken, inputS3Url } = getObjectContext;

    // In this case, we're resizing '.png' images that are stored in S3 and are
    // accessible through the presigned URL
    // 'inputS3Url'.
    const { data } = await axios.get(inputS3Url, { responseType: 'arraybuffer' });

    // Resize the image.
    const resized = await sharp(data)
        .resize({ width: 256, height: 256 })
        .toBuffer();

    // Send the resized image back to the client.
    await s3.writeGetObjectResponse({
        RequestRoute: outputRoute,
        RequestToken: outputToken,
        Body: resized,
    }).promise();

    // Gracefully exit the Lambda function.
    return { statusCode: 200 };
}
```

Exemplo 3:

Quando objetos são comprimidos, os dados compactados são produzidos de forma incremental. Consequentemente, você pode usar sua solicitação WriteGetObjectResponse para retornar os dados compactados assim que eles estiverem prontos. Como mostrado neste exemplo, não é necessário saber o tamanho da transformação concluída.

Java

```
package com.amazonaws.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example3 {

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();
        HttpClient httpClient = HttpClient.newBuilder().build();

        // Request the original object from S3.
        var presignedResponse = httpClient.send(

```

```
    HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
    HttpResponse.BodyHandlers.ofInputStream());

    // Consume the incoming response body from the presigned request,
    // apply our transformation on that data, and emit the transformed bytes
    // into the body of the WriteGetObjectResponse request as soon as they're
    ready.
    // This example compresses the data from S3, but any processing pertinent
    // to your application can be performed here.
    var bodyStream = new GZIPCompressingInputStream(presignedResponse.body());

    // Stream the bytes back to the caller.
    s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
        .withRequestRoute(event.outputRoute())
        .withRequestToken(event.outputToken())
        .withInputStream(bodyStream));
}

}
```

Python

```
import boto3
import requests
import zlib
from botocore.config import Config

"""
A helper class to work with content iterators. Takes an interator and compresses the
bytes that come from it. It
implements 'read' and '__iter__' so that the SDK can stream the response.
"""
class Compress:
    def __init__(self, content_iter):
        self.content = content_iter
        self.compressed_obj = zlib.compressobj()

    def read(self, _size):
        for data in self.__iter__():
            return data

    def __iter__(self):
        while True:
            data = next(self.content)
            chunk = self.compressed_obj.compress(data)
            if not chunk:
                break

            yield chunk

        yield self.compressed_obj.flush()

def handler(event, context):
    """
    Setting the 'payload_signing_enabled' property to False allows us to send a
    streamed response back to the client.
    in this scenario, a streamed response means that the bytes are not buffered into
    memory as we're compressing them,
    but instead are sent straight to the user.
    """
    my_config = Config(
        region_name='eu-west-1',
```

```
signature_version='s3v4',
s3={
    "payload_signing_enabled": False
}
)
s3 = boto3.client('s3', config=my_config)

"""
Retrieve the operation context object from the event. This object indicates where
the WriteGetObjectResponse request
should be delivered and has a presigned URL in 'inputS3Url' where we can download
the requested object from.
The 'userRequest' object has information related to the user who made this
'GetObject' request to S3 Object Lambda.
"""
get_context = event["getObjectContext"]
route = get_context["outputRoute"]
token = get_context["outputToken"]
s3_url = get_context["inputS3Url"]

# Compress the 'get' request stream.
with requests.get(s3_url, stream=True) as r:
    compressed = Compress(r.iter_content())

# Send the stream back to the client.
s3.writeGetObjectResponse(Body=compressed, RequestRoute=route,
RequestToken=token, ContentType="text/plain",
ContentEncoding="gzip")

# Gracefully exit the Lambda function.
return {'status_code': 200}
```

Node.js

```
const { S3 } = require('aws-sdk');
const axios = require('axios').default;
const zlib = require('zlib');

exports.handler = async (event) => {
    const s3 = new S3();

    // Retrieve the operation context object from the event. This object indicates
    // where the WriteGetObjectResponse request
    // should be delivered and has a presigned URL in 'inputS3Url' where we can
    // download the requested object from.
    const { getObjectContext } = event;
    const { outputRoute, outputToken, inputS3Url } = getObjectContext;

    // Download the object from S3 and process it as a stream, because it might be a
    // huge object and we don't want to
    // buffer it in memory. Note the use of 'await' because we want to wait for
    // 'writeGetObjectResponse' to finish
    // before we can exit the Lambda function.
    await axios({
        method: 'GET',
        url: inputS3Url,
        responseType: 'stream',
    }).then(
        // Gzip the stream.
        response => response.data.pipe(zlib.createGzip())
    ).then(
        // Finally send the gzip-ed stream back to the client.
        stream => s3.writeGetObjectResponse({
            RequestRoute: outputRoute,
```

```
    RequestToken: outputToken,
    Body: stream,
    ContentType: "text/plain",
    ContentEncoding: "gzip",
  }).promise()
);

// Gracefully exit the Lambda function.
return { statusCode: 200 };
}
```

Note

Embora o S3 Object Lambda permita enviar uma resposta completa ao autor da chamada por meio da solicitação `WriteGetObjectResponse`, a quantidade real de tempo disponível pode ser menor. Por exemplo, o tempo limite da função do Lambda pode ser inferior a 60 segundos. Em outros casos, o autor da chamada pode ter tempos limites mais rigorosos.

Para que o autor da chamada original receba uma resposta diferente de 500 (Erro interno do servidor), a chamada `WriteGetObjectResponse` deve ser concluída. Se a função do Lambda retornar, excepcionalmente ou de outra forma, antes que a operação da API `WriteGetObjectResponse` seja chamada, o autor da chamada original receberá uma resposta 500. Exceções lançadas durante o tempo necessário para concluir a resposta resultam em respostas truncadas ao autor da chamada. Se a função do Lambda recebe uma resposta 200 (OK) da chamada da API `WriteGetObjectResponse`, o autor da chamada original enviou a solicitação completa. A resposta da função do Lambda, independentemente de uma exceção ser lançada ou não, é ignorada pelo S3 Object Lambda.

Ao chamar essa API, o Amazon S3 exige a rota e o token de solicitação do contexto do evento. Para obter mais informações, consulte [Formato e uso de contexto de evento \(p. 313\)](#).

Esses parâmetros são necessários para conectar a resposta `WriteGetObjectResult` com o autor da chamada original. Ainda que sempre seja apropriado repetir respostas 500, observe que, porque o token de solicitação é um token de uso único, tentativas subsequentes de usá-lo podem resultar em respostas 400 (Solicitação inválida). Embora a chamada para `WriteGetObjectResponse` com os tokens de rota e solicitação não precise ser feita a partir da função do Lambda invocada, ela deve ser feita por uma identidade na mesma conta. A chamada também deverá ser concluída antes que a função do Lambda termine a execução.

Depuração do S3 Object Lambda

As solicitações de `GetObject` para pontos de acesso do S3 Object Lambda podem resultar em novas respostas de erro quando algo der errado com a invocação ou execução da função do Lambda. Esses erros seguem o mesmo formato que os erros padrão do Amazon S3. Para obter informações sobre erros do S3 Object Lambda, consulte [Lista de códigos de erro do S3 Object Lambda](#) na Referência da API do Amazon Simple Storage Service.

Para obter mais informações sobre a depuração geral da função do Lambda, consulte [Monitoramento e solução de problemas de aplicações do Lambda](#) no Guia do desenvolvedor do AWS Lambda.

Para obter informações sobre erros padrão do Amazon S3, consulte [Respostas de erro](#) na Referência da API do Amazon Simple Storage Service.

Você pode habilitar métricas de solicitação no Amazon CloudWatch para seus pontos de acesso do Object Lambda. Essas métricas podem ser usadas para monitorar a performance operacional do seu ponto de acesso.

Para obter registros em log mais granulares sobre solicitações feitas aos seus pontos de acesso do Object Lambda, é possível habilitar eventos de dados do AWS CloudTrail. Para obter mais informações, consulte [Registro eventos de dados em logs para trilhas](#) no Guia do usuário do AWS CloudTrail.

Trabalhar com cabeçalhos Range e partNumber

Ao trabalhar com objetos grandes, você pode usar o cabeçalho HTTP `Range` para baixar um intervalo de bytes especificado de um objeto. É possível usar conexões simultâneas ao Amazon S3 para buscar diferentes escalas de bytes no mesmo objeto. Você também pode usar o parâmetro `partNumber` (inteiro entre 1 e 10.000) que efetivamente executa uma solicitação GET "em um intervalo" para a parte especificada do objeto. Para obter mais informações, consulte [Sintaxe da solicitação GetObject](#) na Referência da API do Amazon Simple Storage Service.

Como há várias maneiras pelas quais você pode querer lidar com uma solicitação que inclua os parâmetros `Range` ou `partNumber`, o S3 Object Lambda não aplica esses parâmetros ao objeto transformado. Em vez disso, sua função do Lambda deve implementar essa funcionalidade conforme necessário para a aplicação.

Para usar os parâmetros `Range` e `partNumber` com o S3 Object Lambda, é necessário habilitar esses parâmetros para o ponto de acesso do Object Lambda e, em seguida, escrever uma função do Lambda capaz de lidar com solicitações que incluem esses parâmetros. As etapas a seguir descrevem como fazer isso.

Etapa 1: Configurar seu ponto de acesso do Object Lambda

Por padrão, os pontos de acesso do Object Lambda respondem com um erro HTTP 501 (Não implementado) a qualquer solicitação `GetObject` que contém um parâmetro `Range` ou `partNumber` nos cabeçalhos ou nos parâmetros de consulta. Para permitir que um ponto de acesso do Object Lambda aceite tais solicitações, é necessário atualizar a configuração do ponto de acesso do Object Lambda usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) ou AWS SDKs. Para obter mais informações sobre como atualizar a configuração do ponto de acesso do Object Lambda, consulte [Criar pontos de acesso do Object Lambda](#) (p. 290).

Etapa 2: Implementar o manuseio de intervalos ou partNumber na função do Lambda

Quando seu ponto de acesso do Object Lambda invoca a função do Lambda com uma solicitação `GetObject` em um intervalo, o parâmetro `Range` ou `partNumber` é incluído no contexto do evento. A localização do parâmetro no contexto do evento depende de qual parâmetro foi usado e de como ele foi incluído na solicitação original para o ponto de acesso do Object Lambda, conforme explicado na tabela a seguir.

Parâmetro	Localização do contexto do evento
<code>Range</code> (cabeçalho)	<code>userRequest.headers.Range</code>
<code>Range</code> (parâmetro de consulta)	<code>userRequest.url</code> (parâmetro de consulta <code>Range</code>)
<code>partNumber</code>	<code>userRequest.url</code> (parâmetro de consulta <code>partNumber</code>)

Important

O URL pré-assinado fornecido não contém o parâmetro `Range` ou `partNumber` da solicitação original. Se você desejar recuperar somente o intervalo especificado do Amazon S3, será necessário adicionar o parâmetro ao URL pré-assinado.

Depois de extrair o valor `Range` ou `partNumber`, você poderá adotar uma das seguintes abordagens com base nas necessidades do aplicativo:

A. Mapear o intervalo ou partNumber solicitado para o objeto transformado (recomendado)

A maneira mais confiável de lidar com solicitações Range ou partNumber é recuperar o objeto completo do S3, transformar o objeto e, em seguida, aplicar os parâmetros Range ou partNumber solicitados ao objeto transformado. Para fazer isso, use o URL pré-assinado fornecido para buscar todo o objeto do Amazon S3 e, em seguida, processe o objeto conforme necessário. Para obter um exemplo de função do Lambda que processa o parâmetro Range dessa forma, consulte [este exemplo](#) no repositório de exemplos da AWS no GitHub.

B. Mapear o intervalo ou partNumber solicitado no URL pré-assinado

Em alguns casos, sua função do Lambda pode mapear o intervalo ou o partNumber solicitado diretamente no URL pré-assinado para recuperar apenas parte do objeto do Amazon S3. Essa abordagem será adequada somente se sua transformação atender a ambos os seguintes critérios:

1. Sua função de transformação pode ser aplicada a intervalos de objetos parciais.
2. O parâmetro Range ou partNumber atua nos mesmos dados tanto no objeto original quanto no objeto transformado.

Por exemplo, uma função de transformação que converte todos os caracteres em um objeto codificado em ASCII em maiúsculas atende a ambos os critérios anteriores. A transformação pode ser aplicada a parte de um objeto, e aplicar o parâmetro Range ou partNumber antes da transformação produz mesmo resultado que a aplicação do parâmetro após a transformação.

Por outro lado, uma função que inverte os caracteres em um objeto codificado em ASCII não atende a esses critérios. Essa função atende ao critério 1, pois pode ser aplicada a intervalos de objetos parciais. No entanto, ela não atende ao critério 2, porque aplicar o parâmetro Range antes da transformação produz resultados diferentes do que a aplicação do parâmetro após a transformação.

Por exemplo, considere uma solicitação para aplicar a função aos três primeiros caracteres de um objeto com o conteúdo abcdefg. Aplicar o parâmetro Range antes da transformação recupera somente abc e, em seguida, reverte os dados, retornando cba. Mas se o parâmetro for aplicado após a transformação, a função recuperará todo o objeto, o reverte e, em seguida, aplicará o parâmetro Range, retornando gfe. Como esses resultados são diferentes, essa função não deve aplicar o parâmetro Range ao recuperar o objeto do Amazon S3. Em vez disso, ela deve recuperar todo o objeto, executar a transformação e, somente então, aplicar o parâmetro Range.

Warning

Em muitos casos, aplicar o parâmetro Range ou partNumber ao URL pré-assinado resultará em um comportamento inesperado da função do Lambda ou do cliente solicitante. A menos que você tenha certeza de que seu aplicativo funcionará corretamente ao recuperar apenas um objeto parcial do Amazon S3, recomendamos recuperar e transformar objetos completos, conforme descrito anteriormente na abordagem A.

Se a aplicação atender aos critérios descritos anteriormente, você poderá simplificar a função AWS Lambda e minimizar os custos de transferência de dados buscando apenas o intervalo de objetos solicitado e, em seguida, executando sua transformação nesse intervalo.

O exemplo de código a seguir demonstra como recuperar o cabeçalho Range da solicitação GetObject e adicioná-lo ao URL pré-assinado que o Lambda pode usar para recuperar o intervalo solicitado do Amazon S3.

```
private HttpRequest.Builder applyRangeHeader(ObjectLambdaEvent event, HttpRequest.Builder presignedRequest) {
    var header = event.getUserRequest().getHeaders().entrySet().stream()
        .filter(e -> e.getKey().toLowerCase(Locale.ROOT).equals("range"))
        .findFirst();

    // Add check in the query string itself.
    header.ifPresent(entry -> presignedRequest.header(entry.getKey(), entry.getValue()));
}
```

```
    return presignedRequest;
}
```

Formato e uso de contexto de evento

O S3 Object Lambda fornece contexto sobre a solicitação que está sendo feita no evento passado para a função do Lambda. O exemplo a seguir mostra uma solicitação e descrições de campo.

```
{
    "xAmzRequestId": "requestId",
    "getObjectContext": {
        "inputS3Url": "https://my-s3-ap-111122223333.s3-accesspoint.us-east-1.amazonaws.com/example?X-Amz-Security-Token=<snip>",
        "outputRoute": "io-use1-001",
        "outputToken": "OutputToken"
    },
    "configuration": {
        "accessPointArn": "arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/example-object-lambda-ap",
        "supportingAccessPointArn": "arn:aws:s3:us-east-1:111122223333:accesspoint/example-ap",
        "payload": "{}"
    },
    "userRequest": {
        "url": "https://object-lambda-111122223333.s3-object-lambda.us-east-1.amazonaws.com/example",
        "headers": {
            "Host": "object-lambda-111122223333.s3-object-lambda.us-east-1.amazonaws.com",
            "Accept-Encoding": "identity",
            "X-Amz-Content-SHA256": "e3b0c44298fc1example"
        }
    },
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "principalId",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
        "accountId": "111122223333",
        "accessKeyId": "accessKeyId",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "principalId",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            }
        }
    },
    "protocolVersion": "1.00"
}
```

- **xAmzRequestId**: o ID da solicitação do Amazon S3 para essa solicitação. Recomendamos que você registre em log esse valor para ajudar na depuração.
- **getObjectContext**: os detalhes de entrada e saída para conexões com o Amazon S3 e o S3 Object Lambda.
- **inputS3Url**: um URL pré-designado que pode ser usado para buscar o objeto original do Amazon S3. O URL é assinado usando a identidade do autor da chamada original, e suas permissões serão

aplicadas quando o URL for usado. Se houver cabeçalhos assinados no URL, a função do Lambda deverá incluí-los na chamada para o Amazon S3, exceto para o cabeçalho Host.

- **outputRoute:** um token de roteamento que é adicionado ao URL do S3 Object Lambda quando a função do Lambda chama `WriteGetObjectResponse`.
- **outputToken:** um token opaco usado pelo S3 Object Lambda para corresponder à chamada de `WriteGetObjectResponse` com o autor da chamada original.
- **configuration:** informações de configuração sobre o ponto de acesso do Object Lambda.
 - **accessPointArn:** o nome do recurso da Amazon (ARN) do ponto de acesso do Object Lambda que recebeu essa solicitação.
 - **supportingAccessPointArn:** o ARN do ponto de acesso de suporte especificado na configuração do ponto de acesso do Object Lambda.
 - **payload:** dados personalizados que são aplicados à configuração do ponto de acesso do Object Lambda. O S3 Object Lambda trata esses dados como uma string opaca, portanto, ela pode precisar ser decodificada antes do uso.
- **userRequest:** informações sobre a chamada original para o S3 Object Lambda.
 - **url:** o URL decodificado da solicitação, conforme recebido pelo S3 Object Lambda, excluindo qualquer parâmetro de consulta relacionado à autorização.
 - **headers:** um mapa de string para strings contendo os cabeçalhos HTTP e seus valores da chamada original, excluindo qualquer cabeçalho relacionado à autorização. Se o mesmo cabeçalho aparecer várias vezes, os valores de cada instância do mesmo cabeçalho serão combinados em uma lista delimitada por vírgulas. O caso dos cabeçalhos originais é retido neste mapa.
- **userIdentity:** detalhes sobre a identidade que fez a chamada para o S3 Object Lambda. Para obter mais informações, consulte [Registro de eventos de dados em logs para trilhas](#) no Guia do usuário do AWS CloudTrail.
 - **type:** o tipo da identidade.
 - **accountId:** a Conta da AWS à qual a identidade pertence.
 - **userName:** o nome amigável da identidade que fez a chamada.
 - **principalId:** o identificador exclusivo da identidade que fez a chamada.
 - **arn:** o ARN da entidade principal que fez a chamada. A última seção do ARN contém o usuário ou função que fez a chamada.
 - **sessionContext:** se a solicitação foi feita com credenciais de segurança temporárias, esse elemento fornece informações sobre a sessão que foi criada para essas credenciais.
 - **invokedBy:** o nome do serviço da AWS service (Serviço da AWS) que fez a solicitação, como o Amazon EC2 Auto Scaling ou AWS Elastic Beanstalk.
 - **sessionIssuer:** se a solicitação foi feita com credenciais de segurança temporárias, esse elemento fornece informações sobre como as credenciais foram obtidas.
- **protocolVersion:** o ID da versão do contexto fornecido. O formato desse campo é `{Major Version}.{Minor Version}`. Os números de versão menores são sempre números de dois dígitos. Qualquer remoção ou alteração na semântica de um campo exigirá o aumento da versão principal e adesão ativa. O Amazon S3 pode adicionar novos campos a qualquer momento, no qual você pode experimentar um aumento da versão secundária. Devido à natureza das implementações de software, é possível que você veja várias versões secundárias em uso ao mesmo tempo.

Usar funções do Lambda criadas pela AWS

A AWS fornece algumas funções pré-criadas do Lambda que você pode usar com o S3 Object Lambda para detectar e remover informações de identificação pessoal (PII) e descompactar objetos do S3. Essas funções do Lambda estão disponíveis no AWS Serverless Application Repository e podem ser selecionadas por meio do AWS Management Console ao criar o ponto de acesso do Object Lambda.

Para obter mais informações sobre como implantar aplicações sem servidor pelo AWS Serverless Application Repository, consulte [Implantação de aplicações](#) no Guia do desenvolvedor do AWS Serverless Application Repository.

Exemplo 1: Controle de Acesso da PII

Essa função do Lambda usa o Amazon Comprehend, um Natural Language Processing (NLP – Serviço de processamento de linguagem natural) que usa machine learning para encontrar insights e relações no texto. Ele detecta automaticamente informações de identificação pessoal (PII), como nomes, endereços, datas, números de cartão de crédito e números de previdência social de documentos em seu bucket do Amazon S3. Se você tiver documentos no bucket que incluam PII, poderá configurar a função do S3 Object Lambda de controle de acesso da PII para detectar esses tipos de entidade da PII e restringir o acesso a usuários não autorizados.

Para começar, basta implantar a função do Lambda a seguir em sua conta e adicionar o ARN na configuração do ponto de acesso do Object Lambda.

ARN:

```
arn:aws:serverlessrepo:us-east-1:839782855223:applications/  
ComprehendPiiAccessControls3ObjectLambda
```

Você pode adicionar a exibição dessa função no AWS Management Console usando o seguinte link do SAR: [ComprehendPiiAccessControls3ObjectLambda](#).

Para visualizar essa função no GitHub, consulte [S3 Object Lambda do Amazon Comprehend](#).

Exemplo 2: edição de PII

Essa função do Lambda usa o Amazon Comprehend, um Natural Language Processing (NLP – Serviço de processamento de linguagem natural) que usa machine learning para encontrar insights e relações no texto. Ele remove automaticamente informações de identificação pessoal (PII), como nomes, endereços, datas, números de cartão de crédito e números de previdência social, de documentos em seu bucket do Amazon S3. Se você tiver documentos em seu bucket que incluam informações como números de cartão de crédito ou informações de contas bancárias, poderá configurar a função do S3 Object Lambda de edição de PII para detectar PII e, em seguida, retornar uma cópia desses documentos nas quais os tipos de entidade de PII estejam editados.

Para começar, basta implantar a função do Lambda a seguir em sua conta e adicionar o ARN na configuração do ponto de acesso do Object Lambda.

ARN:

```
arn:aws:serverlessrepo:us-east-1:839782855223:applications/  
ComprehendPiiRedaction3ObjectLambda
```

Você pode adicionar a exibição dessa função no AWS Management Console usando o seguinte link do SAR: [ComprehendPiiRedaction3ObjectLambda](#).

Para visualizar essa função no GitHub, consulte [S3 Object Lambda do Amazon Comprehend](#).

Exemplo 3: descompressão

A função do Lambda S3ObjectLambdaDecompression está equipada para descomprimir objetos armazenados no S3 em um dos seis formatos de arquivo compactados, incluindo bzip2, gzip, snappy, zlib,

zstandard e ZIP. Para começar, basta implantar a função do Lambda a seguir em sua conta e adicionar o ARN na configuração do ponto de acesso do Object Lambda.

ARN:

```
arn:aws:serverlessrepo:eu-west-1:123065155563:applications/S3ObjectLambdaDecompression
```

Você pode adicionar a exibição dessa função no AWS Management Console usando o seguinte link do SAR: [S3ObjectLambdaDecompression](#).

Para visualizar essa função no GitHub, consulte [Descompressão do S3 Object Lambda](#).

Práticas recomendadas e diretrizes para o S3 Object Lambda

Ao usar o S3 Object Lambda, siga essas práticas recomendadas e diretrizes para otimizar operações e performance.

Tópicos

- [Trabalhar com o S3 Object Lambda \(p. 316\)](#)
- [Produtos da AWS usados em conexão com o S3 Object Lambda \(p. 316\)](#)
- [Trabalhar com cabeçalhos GET Range e partNumber \(p. 317\)](#)
- [Como trabalhar com a AWS CLI e os SDKs \(p. 317\)](#)

Trabalhar com o S3 Object Lambda

O S3 Object Lambda é compatível apenas com o processamento de solicitações GetObject. Qualquer solicitação que não seja GET, como ListObjects ou HeadObject, não invocará o Lambda e retornará respostas de API padrão e não transformadas. Você pode criar no máximo 1.000 pontos de acesso do Object Lambda por Conta da AWS por região. A função do AWS Lambda que você usa deve estar nas mesmas Conta da AWS e região que o ponto de acesso do Object Lambda.

O S3 Object Lambda permite transmitir uma resposta completa ao autor da chamada em até 60 segundos. Sua função também está sujeita a cotas padrão do Lambda. Para obter mais informações, consulte [Cotas do Lambda](#) no Guia do desenvolvedor do AWS Lambda. O uso do S3 Object Lambda invoca sua função do Lambda especificada, e você é responsável por garantir que todos os dados substituídos ou excluídos do S3 pela função ou aplicação do Lambda especificada sejam intencionais e corretos.

Você só pode usar o S3 Object Lambda para executar operações em objetos. Você não pode usá-lo para realizar outras operações do Amazon S3, como modificar ou excluir buckets. Para obter uma lista completa das operações do S3 que oferecem suporte a pontos de acesso, consulte [Compatibilidade com ponto de acesso com produtos da AWS \(p. 335\)](#).

Além dessa lista, os pontos de acesso do S3 Object Lambda não suportam [POST Object](#), [Copiar](#) (como origem) ou [Selecionar conteúdo do objeto](#).

Produtos da AWS usados em conexão com o S3 Object Lambda

O S3 Object Lambda conecta Amazon S3, o AWS Lambda e, opcionalmente, outros produtos da AWS de sua escolha para fornecer objetos relevantes para as aplicações solicitadas. Todos os produtos da AWS usados em conexão com o S3 Object Lambda continuarão a ser regidos por seus respectivos Acordos de nível de serviço (SLA). Por exemplo, no caso de qualquer produto da AWS não cumprir seu compromisso de serviço, você estará qualificado para receber um crédito de serviço conforme documentado no SLA do produto.

Trabalhar com cabeçalhos GET Range e partNumber

Ao trabalhar com objetos grandes, você pode usar o cabeçalho Range HTTP para fazer download de um byte-range especificado de um objeto buscando apenas a parte especificada. É possível usar conexões simultâneas ao Amazon S3 para buscar diferentes escalas de bytes no mesmo objeto. Você também pode usar partNumber (inteiro entre 1 e 10.000) que efetivamente executa uma solicitação GET “intervalada” para a parte especificada do objeto. Para obter mais informações, consulte [Sintaxe de solicitaçãoGetObject](#) na Referência da API do Amazon Simple Storage Service.

Ao receber uma solicitação GET, o S3 Object Lambda invoca a função do Lambda especificada primeiro, portanto, se sua solicitação GET contiver parâmetros de intervalo ou número de peça, você deve garantir que a função do Lambda esteja equipada para reconhecer e gerenciar esses parâmetros. Como pode haver várias entidades conectadas em tal configuração (solicitando clientes e serviços como Lambda, S3, outros), é aconselhável que todas as entidades envolvidas interpretem o intervalo solicitado (ou partNumber) de forma uniforme. Isso garante que os intervalos esperados pela aplicação correspondam aos intervalos que a função do Lambda está processando. Ao criar uma função para lidar com solicitações com cabeçalhos de intervalo, teste todas as combinações de tamanhos de resposta, tamanhos de objetos originais e tamanhos de intervalo de solicitações que sua aplicação planeja usar.

Por padrão, os pontos de acesso do S3 Object Lambda responderão com um 501 a qualquer solicitação GetObject que contenha um parâmetro de intervalo ou número de peça, seja nos cabeçalhos ou parâmetros de consulta. Você pode confirmar que sua função do Lambda está preparada para lidar com solicitações parciais ou de intervalo atualizando sua configuração de ponto de acesso do Object Lambda por meio do AWS Management Console ou da AWS CLI.

Como trabalhar com a AWS CLI e os SDKs

Os subcomandos do S3 (cp, mv e sync) da AWS CLI e o uso do gerenciador de transferência não são aceitos em combinações com o S3 Object Lambda.

Considerações sobre segurança para pontos de acesso do S3 Object Lambda

O S3 Object Lambda permite aos clientes a capacidade de realizar transformações personalizadas em dados à medida que deixa o S3 usando a escala e a flexibilidade do AWS Lambda Lambda como uma plataforma de computação. O S3 e o Lambda permanecem seguros por padrão, mas é necessária uma consideração especial do autor do Lambda para manter essa segurança. O S3 Object Lambda exige que todo o acesso seja feito por entidades autenticadas (sem acesso anônimo) e por HTTPS.

Para mitigar esse risco, recomendamos que a função de execução do Lambda seja cuidadosamente definida para o menor conjunto de privilégios possível. Além disso, o Lambda deve fazer seus acessos ao S3 por meio do URL pré-assinado fornecido sempre que possível.

Configurar políticas do IAM

Os pontos de acesso do S3 são compatíveis com as políticas de recursos do AWS Identity and Access Management (IAM) que permitem controlar o uso do ponto de acesso por recurso, usuário ou por outras condições. Para obter mais informações, consulte [Configurando políticas do IAM para pontos de acesso do Object Lambda \(p. 299\)](#).

Comportamento de criptografia

Como o ponto de acesso do Object Lambda usa o Amazon S3 e o AWS Lambda Lambda, há diferenças no comportamento de criptografia. Para obter mais informações sobre o comportamento de criptografia

padrão do S3, consulte [Definir o comportamento padrão da criptografia para os buckets do Amazon S3 \(p. 142\)](#).

- Ao usar a criptografia do lado do servidor do S3 com pontos de acesso do Object Lambda, o objeto será descriptografado antes de ser enviado para o AWS Lambda, onde será processado sem criptografia até o autor da chamada original (no caso de um GET).
- Para evitar que a chave seja registrada em log, o S3 rejeitará solicitações GET para objetos criptografados por meio de criptografia no lado do servidor usando chaves fornecidas pelo cliente. A função do Lambda ainda pode recuperar esses objetos desde que tenha acesso à chave fornecida pelo cliente.
- Ao usar a criptografia do lado do cliente do S3 com pontos de acesso do Object Lambda, verifique se o Lambda tem acesso à chave para descriptografar e criptografar novamente o objeto.

Segurança dos pontos de acesso

O S3 Object Lambda usa dois pontos de acesso: um ponto de acesso Object Lambda e um ponto de acesso padrão do S3, referido como o ponto de acesso de suporte. Quando você faz uma solicitação a um ponto de acesso do Object Lambda, o S3 chama o Lambda em seu nome ou delega a solicitação ao ponto de acesso de suporte, de acordo com a configuração do S3 Object Lambda. Quando o Lambda é chamado para o GetObject, o S3 gera um URL pré-assinado para seu objeto em seu nome por meio do ponto de acesso de suporte. Sua função Lambda receberá esse URL como entrada quando chamada.

É possível definir sua função Lambda para usar esse URL ao recuperar o objeto original, em vez de chamar o S3 diretamente. Esse modelo permite aplicar melhores limites de segurança a seus objetos. É possível limitar o acesso direto a objetos por meio de buckets do S3 ou de pontos de acesso do S3 a um conjunto limitado de funções ou usuários do IAM. Isso também protege suas funções Lambda de estarem sujeitas ao problema Confused Deputy, no qual uma função mal configurada com permissões diferentes das permissões de seu chamador GetObject poderia permitir ou negar acesso a objetos quando não deveria.

Acesso público do ponto de acesso do Object Lambda

O S3 Object Lambda não permite acesso anônimo ou público, pois o Amazon S3 precisa autorizar sua identidade para concluir qualquer solicitação do S3 Object Lambda. Ao chamar solicitações do GetObject por um ponto de acesso do Object Lambda, é necessária a permissão `lambda:InvokeFunction` para a função Lambda configurada. Da mesma forma, ao chamar outras APIs por um ponto de acesso do Object Lambda, você precisa ter as permissões `s3>*` necessárias.

Sem essas permissões, as solicitações para chamar o Lambda ou delegar ao S3 apresentarão falha como um erro 403 Forbidden. Todo o acesso deve ser feito por entidades autenticadas. Se precisar de acesso público, você poderá utilizar o `Lambda@Edge` como uma possível alternativa. Para obter mais informações, consulte [Personalizar o conteúdo na borda com o Lambda@Edge](#) no Guia do desenvolvedor do Amazon CloudFront.

IPs de ponto de acesso do Object Lambda

As sub-redes de listas de prefixos gerenciadas por descrição são compatíveis com endpoints da VPC do gateway e estão relacionadas à tabela de roteamento de endpoints da VPC. Os intervalos ausentes pertencem ao Amazon S3, mas não são compatíveis com endpoints da VPC de gateway. Para obter mais informações sobre listas de prefixos gerenciadas por descrição, consulte [DescribePrefixLists](#) na Referência da API do Amazon EC2 e [Intervalos de endereços IPAWS](#) na Referência geral da AWS.

Tutoriais do S3 Object Lambda

Os tutoriais a seguir apresentam procedimentos completos para algumas tarefas do S3 Object Lambda.

- [Tutorial: Como transformar dados para sua aplicação com o S3 Object Lambda \(p. 30\)](#)
- [Tutorial: Detectar e editar dados PII com o S3 Object Lambda e o Amazon Comprehend \(p. 45\)](#)

Para obter os tutoriais sobre o Lambda para objetos do S3, consulte [Tutorial: Como transformar dados para sua aplicação com o S3 Object Lambda \(p. 30\)](#) e [Tutorial: Detectar e editar dados PII com o S3 Object Lambda e o Amazon Comprehend \(p. 45\)](#).

Para obter mais informações sobre pontos de acesso padrão, consulte [Gerenciamento de acesso a dados com pontos de acesso do Amazon S3 \(p. 320\)](#).

Para obter informações sobre como trabalhar com buckets, consulte [Visão geral dos buckets \(p. 123\)](#). Para obter mais informações sobre como trabalhar com objetos, consulte [Visão geral de objetos Amazon S3 \(p. 161\)](#).

Gerenciamento de acesso a dados com pontos de acesso do Amazon S3

Os pontos de acesso do Amazon S3 simplificam o acesso a dados para qualquer produto da AWS ou aplicação de clientes que armazene dados no S3. Os pontos de acesso são nomeados endpoints de rede anexados a buckets que você pode usar para executar operações de objeto do S3, como `GetObject` e `PutObject`. Cada ponto de acesso tem permissões distintas e controles de rede que o S3 aplica para qualquer solicitação feita por meio desse ponto de acesso. Cada ponto de acesso impõe uma política de ponto de acesso personalizada que funciona em conjunto com a política de bucket anexada ao bucket subjacente. Você pode configurar qualquer ponto de acesso para aceitar solicitações somente de uma Virtual Private Cloud (VPC) para restringir o acesso a dados do Amazon S3 a uma rede privada. Você também pode configurar definições personalizadas do Bloqueio de acesso público para cada ponto de acesso.

Note

- Você só pode usar pontos de acesso para executar operações em objetos. Não é possível usar pontos de acesso para executar outras operações do Amazon S3, como modificar ou excluir buckets. Para obter uma lista completa das operações do S3 que oferecem suporte a pontos de acesso, consulte [Compatibilidade com ponto de acesso com produtos da AWS \(p. 335\)](#).
- Os pontos de acesso funcionam com alguns serviços e recursos da AWS, mas não todos. Por exemplo, você não pode configurar a replicação entre regiões para operar por meio de um ponto de acesso. Para obter uma lista completa de serviços da AWS compatíveis com pontos de acesso do S3, consulte [Compatibilidade com ponto de acesso com produtos da AWS \(p. 335\)](#).

Esta seção explica como trabalhar com pontos de acesso do Amazon S3. Para obter informações sobre como trabalhar com buckets, consulte [Visão geral dos buckets \(p. 123\)](#). Para obter mais informações sobre como trabalhar com objetos, consulte [Visão geral de objetos Amazon S3 \(p. 161\)](#).

Tópicos

- [Configurar políticas do IAM para uso de pontos de acesso \(p. 320\)](#)
- [Criar pontos de acesso \(p. 325\)](#)
- [Usar pontos de acesso \(p. 329\)](#)
- [Restrições e limitações de pontos de acesso \(p. 337\)](#)

Configurar políticas do IAM para uso de pontos de acesso

Os pontos de acesso do Amazon S3 são compatíveis com as políticas de recursos do AWS Identity and Access Management (IAM) que permitem controlar o uso do ponto de acesso por recurso, usuário ou por outras condições. Para que um aplicativo ou usuário possa acessar objetos por meio de um ponto de acesso, tanto o ponto de acesso quanto o bucket subjacente devem permitir a solicitação.

Important

Adicionar um ponto de acesso S3 a um bucket não altera o comportamento do bucket quando acessado por meio do nome de bucket existente ou do ARN. Todas as operações existentes no bucket continuarão a funcionar como antes. As restrições que você incluir em uma política de ponto de acesso se aplicam somente a solicitações feitas por meio desse ponto de acesso.

Chaves de condição

Os pontos de acesso do S3 apresentam três novas chaves de condição que podem ser usadas em políticas do IAM para controlar o acesso aos recursos:

s3:DataAccessPointArn

Esta é uma string que você pode usar para corresponder um ARN de ponto de acesso. O exemplo a seguir estabelece a correspondência de todos os pontos de acesso da Conta da AWS 123456789012 na região us-west-2:

```
"Condition" : {
    "StringLike": {
        "s3:DataAccessPointArn": "arn:aws:s3:us-west-2:123456789012:accesspoint/*"
    }
}
```

s3:DataAccessPointAccount

Este é um operador de string que você pode utilizar para corresponder o ID da conta do proprietário de um ponto de acesso. O exemplo a seguir estabelece a correspondência de todos os pontos de acesso pertencentes à Conta da AWS 123456789012.

```
"Condition" : {
    "StringEquals": {
        "s3:DataAccessPointAccount": "123456789012"
    }
}
```

s3:AccessPointNetworkOrigin

Este é um operador de string que você pode utilizar para corresponder a origem da rede, Internet ou VPC. O exemplo a seguir corresponde apenas a pontos de acesso com uma origem de VPC.

```
"Condition" : {
    "StringEquals": {
        "s3:AccessPointNetworkOrigin": "VPC"
    }
}
```

Para obter mais informações sobre o uso de chaves de condição com o Amazon S3, consulte [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#).

Delegar controle de acesso a pontos de acesso

Você pode delegar o controle de acesso de um bucket aos pontos de acesso do bucket. A política de bucket de exemplo a seguir permite acesso total a todos os pontos de acesso pertencentes à conta do proprietário do bucket. Assim, todo o acesso a esse bucket é controlado pelas políticas anexadas aos seus

pontos de acesso. Recomendamos configurar seus buckets dessa maneira para todos os casos de uso que não exigem acesso direto ao bucket.

Example Política de bucket para delegar controle de acesso a pontos de acesso

```
{  
    "Version": "2012-10-17",  
    "Statement" : [  
        {  
            "Effect": "Allow",  
            "Principal" : { "AWS": "*" },  
            "Action" : "*",  
            "Resource" : [ "Bucket ARN", "Bucket ARN/*"],  
            "Condition": {  
                "StringEquals" : { "s3:DataAccessPointAccount" : "Bucket owner's account ID" }  
            }  
        }  
    ]  
}
```

Exemplos de política de ponto de acesso

Os exemplos a seguir demonstram como criar políticas do IAM para controlar solicitações feitas por meio de um ponto de acesso.

Note

As permissões concedidas em uma política de ponto de acesso entram em vigor somente se o bucket subjacente também permitir o mesmo acesso. É possível fazer isso de duas maneiras:

1. (Recomendado) Delegue o controle de acesso do bucket para o ponto de acesso conforme descrito em [Delegar controle de acesso a pontos de acesso \(p. 321\)](#).
2. Adicione as mesmas permissões contidas na política de ponto de acesso à política do bucket subjacente. O primeiro exemplo de política de ponto de acesso demonstra como modificar a política de bucket subjacente para permitir o acesso necessário.

Example Concessão da política de ponto de acesso

A política de ponto de acesso a seguir concede ao usuário do IAM Alice na conta 123456789012 permissões para objetos GET e PUT com o prefixo Alice/ por meio do ponto de acesso my-access-point em uma conta 123456789012.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::123456789012:user/Alice"  
            },  
            "Action": [ "s3:GetObject", "s3:PutObject" ],  
            "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point/object/  
Alice/*"  
        }  
    ]  
}
```

Note

Para que a política de ponto de acesso conceda efetivamente acesso ao usuário Alice, o bucket subjacente também deve permitir o mesmo acesso a Alice. É possível delegar o controle de

acesso do bucket para o ponto de acesso conforme descrito em [Delegar controle de acesso a pontos de acesso \(p. 321\)](#). Ou é possível adicionar a seguinte política ao bucket subjacente para conceder as permissões necessárias à Alice. Observe que a entrada Resource é diferente entre as políticas de ponto de acesso e bucket.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::123456789012:user/Alice"  
            },  
            "Action": ["s3:GetObject", "s3:PutObject"],  
            "Resource": "arn:aws:s3:::awsexamplebucket1/Alice/*"  
        }  
    ]  
}
```

Example Política de ponto de acesso com condição de tag

A política de ponto de acesso a seguir concede ao usuário do IAM Bob na conta 123456789012 permissões para objetos GET por meio do ponto de acesso my-access-point na conta 123456789012 que possui a chave de tag data definida com um valor finance.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::123456789012:user/Bob"  
            },  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point/object/*",  
            "Condition": {  
                "StringEquals": {  
                    "s3:ExistingObjectTag/data": "finance"  
                }  
            }  
        }  
    ]  
}
```

Example Política de ponto de acesso permitindo a listagem de bucket

A política de ponto de acesso a seguir permite que o usuário do IAM Charles na conta 123456789012 visualize os objetos contidos no ponto de acesso subjacente do bucket my-access-point na conta 123456789012.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::123456789012:user/Charles"  
            },  
            "Action": "s3>ListBucket",  
            "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point"  
        }  
    ]  
}
```

Example Política de controle de serviço

A política de controle de serviço a seguir requer que todos os novos pontos de acesso sejam criados com uma origem de rede da VPC. Com essa política em vigor, os usuários da sua organização não podem criar novos pontos de acesso acessíveis da Internet.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3>CreateAccessPoint",  
            "Resource": "*",  
            "Condition": {  
                "StringNotEquals": {  
                    "s3:AccessPointNetworkOrigin": "VPC"  
                }  
            }  
        }  
    ]  
}
```

Example Política de bucket para limitar operações do S3 às origens de rede da VPC

A política de bucket a seguir limita o acesso a todas as operações de objeto do S3 para o bucket `examplebucket` para pontos de acesso com uma origem de rede da VPC.

Important

Antes de usar uma instrução como este exemplo, certifique-se de que você não precisa usar recursos que não são compatíveis com pontos de acesso, como replicação entre regiões.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": [  
                "s3:AbortMultipartUpload",  
                "s3:BypassGovernanceRetention",  
                "s3>DeleteObject",  
                "s3>DeleteObjectTagging",  
                "s3>DeleteObjectVersion",  
                "s3>DeleteObjectVersionTagging",  
                "s3:GetObject",  
                "s3:GetObjectAcl",  
                "s3:GetObjectLegalHold",  
                "s3:GetObjectRetention",  
                "s3:GetObjectTagging",  
                "s3:GetObjectVersion",  
                "s3GetObjectVersionAcl",  
                "s3:GetObjectVersionTagging",  
                "s3>ListMultipartUploadParts",  
                "s3:PutObject",  
                "s3:PutObjectAcl",  
                "s3:PutObjectLegalHold",  
                "s3:PutObjectRetention",  
                "s3:PutObjectTagging",  
                "s3:PutObjectVersionAcl",  
                "s3:PutObjectVersionTagging",  
                "s3:RestoreObject"  
            ],  
            "Resource": "arn:aws:s3:::examplebucket/*",  
            "Condition": {  
                "StringNotEquals": {  
                    "s3:NetworkOrigin": "VPC"  
                }  
            }  
        }  
    ]  
}
```

```
        "Condition": {
            "StringNotEquals": {
                "s3:AccessPointNetworkOrigin": "VPC"
            }
        }
    ]
}
```

Criar pontos de acesso

O Amazon S3 fornece funcionalidade para criar e gerenciar pontos de acesso. Você pode criar pontos de acesso do S3 usando o AWS Management Console, a AWS Command Line Interface (AWS CLI), os AWS SDKs ou a API REST do Amazon S3.

Por padrão, você pode criar até 10 mil pontos de acesso por região para cada uma das suas Contas da AWS. Se você precisar de mais de 10 mil pontos de acesso para uma única conta em uma única região, poderá solicitar um aumento de cota de serviço. Para obter mais informações sobre cotas de serviço e solicitar um aumento, consulte [Service Quotas da AWS](#) na Referência geral da AWS.

Note

Como talvez você queira divulgar o nome do ponto de acesso para permitir que os usuários o utilizem, evite incluir informações sigilosas no nome do ponto de acesso. Os nomes de ponto de acesso são publicados em um banco de dados acessível ao público conhecido como Sistema de Nome de Domínio (DNS).

Regras para nomear pontos de acesso do Amazon S3

Os nomes de pontos de acesso devem atender às seguintes condições:

- Devem ser exclusivos em uma única Conta da AWS e região
- Devem estar em conformidade com as restrições de nomenclatura de DNS
- Devem começar com um número ou uma letra minúscula
- Devem conter entre 3 e 50 caracteres
- Não é possível começar ou terminar com um traço
- Não é possível conter sublinhados, letras maiúsculas ou pontos
- Não é possível terminar com o sufixo `-s3alias`. Esse sufixo se reserva a nomes de alias de ponto de acesso. Para obter mais informações, consulte [Usar um alias em estilo de bucket para seu ponto de acesso \(p. 333\)](#).

Para criar um ponto de acesso, consulte os tópicos a seguir.

Tópicos

- [Criar um ponto de acesso \(p. 325\)](#)
- [Criar pontos de acesso restritos a uma nuvem privada virtual \(p. 327\)](#)
- [Gerenciar o acesso público a pontos de acesso \(p. 329\)](#)

Criar um ponto de acesso

Um ponto de acesso está associado a exatamente um bucket do Amazon S3. Antes de começar, certifique-se de que criou um bucket que pretende usar com esse ponto de acesso. Para obter mais informações sobre a criação de buckets, consulte [Criar, configurar e trabalhar com buckets do Amazon S3](#).

S3 (p. 123). Os pontos de acesso do Amazon S3 são compatíveis com as políticas de recursos do AWS Identity and Access Management (IAM) que permitem controlar o uso do ponto de acesso por recurso, usuário ou por outras condições. Para obter mais informações, consulte [Configurar políticas do IAM para uso de pontos de acesso \(p. 320\)](#).

Solucionar avisos de segurança, erros, avisos gerais e sugestões de AWS Identity and Access Management Access Analyzer antes de salvar a política. O IAM Access Analyzer executa verificações de política para validar sua política em relação à [gramática das políticas](#) e às [práticas recomendadas](#) do IAM. Essas verificações geram descobertas e fornecem recomendações práticas que ajudam a criar políticas que sejam funcionais e estejam em conformidade com as práticas recomendadas de segurança. Para saber mais sobre a validação de políticas usando o IAM Access Analyzer, consulte [Validação de políticas do IAM Access Analyzer](#) no Guia do usuário IAM. Para visualizar uma lista de avisos, erros e sugestões retornados pelo IAM Access Analyzer, consulte [Referência de verificação de políticas do IAM Access Analyzer](#).

Por padrão, você pode criar até 10 mil pontos de acesso por região para cada uma das suas Contas da AWS. Se você precisar de mais de 10 mil pontos de acesso para uma única conta em uma única região, poderá solicitar um aumento de cota de serviço. Para obter mais informações sobre cotas de serviço e solicitar um aumento, consulte [Service Quotas da AWS](#) na Referência geral da AWS.

Os exemplos a seguir demonstram como criar um ponto de acesso com a AWS CLI e o console do S3. Para obter mais informações sobre como criar pontos de acesso usando a API REST, consulte [CreateAccessPoint](#) na Referência da API do Amazon Simple Storage Service.

Uso do console do S3

Como criar um ponto de acesso

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, no lado esquerdo do console, escolha Access points (Pontos de acesso).
3. Na página de pontos de acesso, escolha Create access point (Criar ponto de acesso).
4. No campo Access point name (Nome do ponto de acesso), digite o nome desejado para o ponto de acesso. Para obter mais informações sobre nomenclatura de pontos de acesso, consulte [Regras para nomear pontos de acesso do Amazon S3 \(p. 325\)](#).
5. No campo Bucket name (Nome do bucket) insira o nome de um bucket na sua conta à qual você deseja anexar o ponto de acesso, por exemplo [DOC-EXAMPLE-BUCKET1](#). Como alternativa, você pode escolher Browse S3 (Procurar S3) para navegar e pesquisar buckets na sua conta. Se você escolher Browse S3 (Procurar S3), selecione o bucket desejado e Choose path (Escolher caminho) para preencher o campo Bucket name (Nome do bucket) com o nome desse bucket.
6. (Opcional) Escolha View (Exibir) para exibir o conteúdo do bucket especificado em uma nova janela do navegador.
7. Selecione uma Network origin (Origem de rede). Se você escolher Virtual private cloud (VPC), insira o VPC ID (ID da VPC) que você deseja usar com o ponto de acesso.

Para obter mais informações sobre origens de rede para pontos de acesso, consulte [Criar pontos de acesso restritos a uma nuvem privada virtual \(p. 327\)](#).

8. Em Access point settings for Block Public Access (Configurações do ponto de acesso para o Bloqueio de acesso público), selecione as configurações de bloqueio de acesso público que você deseja aplicar ao ponto de acesso. Todas as configurações de bloqueio de acesso público são ativadas por padrão para novos pontos de acesso; recomendamos que você deixe todas as configurações ativadas, a menos que você saiba que tem uma necessidade específica de desabilitar qualquer uma delas. Atualmente, o Amazon S3 não oferece suporte à alteração das configurações do bloqueio de acesso público após à criação de um ponto de acesso.

Para obter mais informações sobre como usar o Bloqueio de acesso público do Amazon S3 com pontos de acesso, consulte [Gerenciar o acesso público a pontos de acesso \(p. 329\)](#).

9. (Opcional) Em Access point policy - optional (Política de ponto de acesso - opcional), especifique a política de ponto de acesso. Solucione avisos de segurança, erros, avisos gerais e sugestões antes de salvar a política. Para obter mais informações sobre como especificar uma política de ponto de acesso, consulte [Exemplos de política de ponto de acesso \(p. 322\)](#).
10. Selecione Criar ponto de acesso.

Uso do AWS CLI

O exemplo a seguir cria um ponto de acesso chamado `example-ap` para o bucket `example-bucket` na conta `123456789012`. Para criar o ponto de acesso, envie uma solicitação ao Amazon S3 especificando o nome do ponto de acesso, o nome do bucket ao qual você deseja associar o ponto de acesso e o ID da Conta da AWS que é proprietária do bucket. Para obter informações sobre regras de nomeação, consulte [the section called “Regras para nomear pontos de acesso do Amazon S3” \(p. 325\)](#).

```
aws s3control create-access-point --name example-ap --account-id 123456789012 --bucket example-bucket
```

Criar pontos de acesso restritos a uma nuvem privada virtual

Ao criar um ponto de acesso, você pode optar por tornar o ponto de acesso acessível da Internet ou pode especificar que todas as solicitações feitas por meio desse ponto de acesso devem ser originadas de uma Virtual Private Cloud (VPC) específica. Considera-se que um ponto de acesso acessível da Internet tem uma origem de rede da Internet. Ele pode ser usado de qualquer lugar na Internet, sujeito a quaisquer outras restrições de acesso em vigor para o ponto de acesso, o bucket subjacente e os recursos relacionados, como os objetos solicitados. Um ponto de acesso acessível apenas de uma VPC especificada tem uma origem de rede de VPC, e o Amazon S3 rejeita qualquer solicitação feita ao ponto de acesso que não tenha origem nessa VPC.

Important

Você só pode especificar a origem da rede de um ponto de acesso ao criá-lo. Depois de criar o ponto de acesso, não é possível alterar a origem da rede.

Para restringir um ponto de acesso ao acesso somente VPC, inclua o parâmetro `VpcConfiguration` com a solicitação para criar o ponto de acesso. No parâmetro `VpcConfiguration`, você especifica o ID da VPC que deseja usar o ponto de acesso. Se uma solicitação for feita por meio do ponto de acesso, ela deverá ser originada da VPC. Do contrário, o Amazon S3 a rejeitará.

Você pode recuperar a origem da rede de um ponto de acesso usando a AWS CLI, os AWS SDKs ou as APIs REST. Se um ponto de acesso tiver uma configuração de VPC especificada, sua origem de rede será VPC. Caso contrário, a origem da rede do ponto de acesso será Internet.

Example

Exemplo: criar um ponto de acesso restrito ao acesso da VPC

O exemplo a seguir cria um ponto de acesso chamado `example-vpc-ap` para o bucket `example-bucket` na conta `123456789012` que permite acesso somente da VPC `vpc-1a2b3c`. O exemplo verifica se o novo ponto de acesso tem uma origem de rede da VPC.

AWS CLI

```
aws s3control create-access-point --name example-vpc-ap --account-id 123456789012 --bucket example-bucket --vpc-configuration VpcId=vpc-1a2b3c
```

```
aws s3control get-access-point --name example-vpc-ap --account-id 123456789012

{
    "Name": "example-vpc-ap",
    "Bucket": "example-bucket",
    "NetworkOrigin": "VPC",
    "VpcConfiguration": {
        "VpcId": "vpc-1a2b3c"
    },
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": true,
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "RestrictPublicBuckets": true
    },
    "CreationDate": "2019-11-27T00:00:00Z"
}
```

Para usar um ponto de acesso com uma VPC, é necessário modificar a política de acesso do VPC endpoint. Os VPC endpoints permitem que o tráfego fluia da VPC para o Amazon S3. Os pontos de acesso têm políticas de controle de acesso que regulam como os recursos na VPC podem interagir com o Amazon S3. As solicitações da VPC ao Amazon S3 serão bem-sucedidas por meio de um ponto de acesso somente se a política do endpoint da VPC conceder acesso ao ponto de acesso e ao bucket subjacente.

Note

Para tornar os recursos acessíveis apenas em uma VPC, crie uma [zona hospedada privada](#) para o endpoint da VPC. Para usar uma zona hospedada privada, [modifique suas configurações da VPC](#) para que os [atributos da rede de VPC](#) `enableDnsHostnames` e `enableDnsSupport` sejam definidos como `true`.

O exemplo de instrução de política a seguir configura um VPC endpoint para permitir chamadas ao `GetObject` de um bucket denominado `awsexamplebucket1` e um ponto de acesso chamado `example-vpc-ap`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::awsexamplebucket1/*",
                "arn:aws:s3:us-west-2:123456789012:accesspoint/example-vpc-ap/object/*"
            ]
        }
    ]
}
```

Note

A declaração "Resource" neste exemplo usa um nome de recurso da Amazon (ARN) para especificar o ponto de acesso. Para obter mais informações sobre ARNs de ponto de acesso, consulte [Usar pontos de acesso \(p. 329\)](#).

Para obter mais informações sobre políticas do endpoint da VPC, consulte [Políticas de endpoint para o Amazon S3](#) no Guia do usuário da VPC.

Gerenciar o acesso público a pontos de acesso

Os pontos de acesso do Amazon S3 oferecem suporte a configurações independentes do bloqueio de acesso público para cada ponto de acesso. Ao criar um ponto de acesso, você pode especificar configurações do Bloqueio de acesso público que se aplicam a esse ponto de acesso. Para qualquer solicitação feita por meio de um ponto de acesso, o Amazon S3 avalia as configurações de bloqueio de acesso público para esse ponto de acesso, o bucket subjacente e a conta do proprietário do bucket. Se qualquer uma dessas configurações indicar que a solicitação deve ser bloqueada, o Amazon S3 rejeitará a solicitação.

Para obter mais informações sobre o recurso de bloqueio de acesso público do S3, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

Important

- Todas as configurações do Bloqueio de acesso público são habilitadas por padrão para pontos de acesso. Você deve desativar explicitamente todas as configurações que não deseja aplicar a um ponto de acesso.
- Atualmente, o Amazon S3 não oferece suporte à alteração das configurações do bloqueio de acesso público após à criação de um ponto de acesso.

Example

Exemplo: criar um ponto de acesso com configurações personalizadas do Bloqueio de acesso público

Este exemplo cria um ponto de acesso chamado `example-ap` para o bucket `example-bucket` na conta `123456789012` com configurações não padrão do Bloqueio de acesso público. O exemplo recupera a configuração do novo ponto de acesso para verificar as configurações do Bloqueio de acesso público.

AWS CLI

```
aws s3control create-access-point --name example-ap --account-id 123456789012 --bucket example-bucket --public-access-block-configuration BlockPublicAccls=false,IgnorePublicAccls=false,BlockPublicPolicy=true,RestrictPublicBuckets=true
```



```
aws s3control get-access-point --name example-ap --account-id 123456789012

{
    "Name": "example-ap",
    "Bucket": "example-bucket",
    "NetworkOrigin": "Internet",
    "PublicAccessBlockConfiguration": {
        "BlockPublicAccls": false,
        "IgnorePublicAccls": false,
        "BlockPublicPolicy": true,
        "RestrictPublicBuckets": true
    },
    "CreationDate": "2019-11-27T00:00:00Z"
}
```

Usar pontos de acesso

Você pode acessar os objetos em um bucket do Amazon S3 com um ponto de acesso usando o AWS Management Console, a AWS CLI, os AWS SDKs ou as APIs REST do S3.

Os pontos de acesso têm nomes de recurso da Amazon (ARNs). Os ARNs de ponto de acesso são semelhantes aos ARNs de bucket, mas são explicitamente digitados e codificam a região do ponto de

acesso e o ID da Conta da AWS do proprietário do ponto de acesso. Para obter mais informações sobre os ARNs, consulte [Nomes de recurso da Amazon \(ARNs\)](#) na Referência geral da AWS.

Os ARNs de ponto de acesso usam o formato `arn:aws:s3:region:account-id:accesspoint/resource`. Por exemplo:

- `arn:aws:s3:us-west-2:123456789012:accesspoint/test` representa o ponto de acesso nomeado `test`, pertencente à conta `123456789012` na região `us-west-2`.
- `arn:aws:s3:us-west-2:123456789012:accesspoint/*` representa todos os pontos de acesso na conta `123456789012` na região `us-west-2`.

Os ARNs para objetos acessados por meio de um ponto de acesso usam o formato `arn:aws:s3:region:account-id:accesspoint/access-point-name/object/resource`. Por exemplo:

- `arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/unit-01` representa o objeto `unit-01`, acessado por meio do ponto de acesso nomeado `test`, pertencente à conta `123456789012` na região `us-west-2`.
- `arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/*` representa todos os objetos para o ponto de acesso `test`, na conta `123456789012` na região `us-west-2`.
- `arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/unit-01/finance/*` representa todos os objetos no prefixo `unit-01/finance/` para o ponto de acesso `test`, na conta `123456789012` na região `us-west-2`.

Tópicos

- [Monitorar e registrar de pontos de acesso \(p. 330\)](#)
- [Como usar pontos de acesso do Amazon S3 com o console do Amazon S3 \(p. 331\)](#)
- [Usar um alias em estilo de bucket para seu ponto de acesso \(p. 333\)](#)
- [Usar pontos de acesso com operações compatíveis com o Amazon S3 \(p. 335\)](#)

Monitorar e registrar de pontos de acesso

O Amazon S3 registra solicitações feitas por meio de pontos de acesso e solicitações feitas às APIs que gerenciam pontos de acesso, como `CreateAccessPoint` e `GetAccessPointPolicy`. Para monitorar e gerenciar padrões de uso, você também pode configurar métricas de solicitação do Amazon CloudWatch Logs para pontos de acesso.

Tópicos

- [Métricas de solicitação do CloudWatch \(p. 330\)](#)
- [Logs da solicitação \(p. 331\)](#)

Métricas de solicitação do CloudWatch

Para entender e melhorar o desempenho das aplicações que estão usando pontos de acesso, você pode usar as métricas de solicitação do CloudWatch para Amazon S3. Métricas de solicitação ajudam a monitorar as solicitações do Amazon S3 para identificar e atuar rapidamente em problemas operacionais.

Por padrão, as métricas de solicitação estão disponíveis em nível de bucket. Porém, também é possível definir um filtro para as métricas de solicitação usando um prefixo compartilhado, etiquetas de objeto ou um ponto de acesso. Quando você cria um filtro de ponto de acesso, a configuração de métricas de solicitação inclui solicitações para o ponto de acesso especificado. Você pode receber métricas, definir alarmes e acessar painéis para ver operações em tempo real executadas por meio desse ponto de acesso.

Você deve aceitar métricas de solicitação configurando-as no console ou usando a API do Amazon S3. As métricas de solicitação estão disponíveis em intervalos de um minuto após alguma latência para processamento. As métricas de solicitação são cobradas usando a mesma taxa das métricas personalizadas do CloudWatch. Para obter mais informações, consulte [Preço do Amazon CloudWatch](#).

Para criar uma configuração de métricas de solicitação que filtre por ponto de acesso, consulte [Criação de uma configuração de métricas que filtre por prefixo, etiqueta de objeto ou ponto de acesso \(p. 1080\)](#).

Logs da solicitação

Você pode registrar em log as solicitações feitas por meio de pontos de acesso e as solicitações feitas às APIs que gerenciam os pontos de acesso, como `CreateAccessPoint` e `GetAccessPointPolicy`, usando o registro de acesso ao servidor e o AWS CloudTrail.

As entradas de log do CloudTrail para solicitações feitas por meio de pontos de acesso incluirão o ARN do ponto de acesso na seção `resources` do log.

Por exemplo, suponha que você tenha a seguinte configuração:

- Um bucket nomeado `DOC-EXAMPLE-BUCKET1` na região `us-west-2` que contém o objeto nomeado `my-image.jpg`
- Um ponto de acesso chamado `my-bucket-ap` que está associado a `DOC-EXAMPLE-BUCKET1`
- Uma ID da Conta da AWS do `123456789012`

O exemplo a seguir mostra a seção `resources` de uma entrada de log do CloudTrail da configuração anterior:

```
"resources": [
    {"type": "AWS::S3::Object",
     "ARN": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/my-image.jpg"
    },
    {"accountId": "123456789012",
     "type": "AWS::S3::Bucket",
     "ARN": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
    },
    {"accountId": "123456789012",
     "type": "AWS::S3::AccessPoint",
     "ARN": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-bucket-ap"
    }
]
```

Para obter mais informações sobre logs de acesso do servidor do S3, consulte [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#). Para obter mais informações sobre o AWS CloudTrail, consulte [What is AWS CloudTrail? \(O que é?\)](#) no AWS CloudTrail Guia do usuário.

Como usar pontos de acesso do Amazon S3 com o console do Amazon S3

Esta seção explica como gerenciar e usar os pontos de acesso do Amazon S3 usando o AWS Management Console. Antes de começar, navegue até a página de detalhes do ponto de acesso que você deseja gerenciar ou usar, conforme descrito no procedimento a seguir.

Tópicos

- [Listar pontos de acesso para sua conta \(p. 332\)](#)
- [Listar pontos de acesso para um bucket \(p. 332\)](#)

- [Visualizar detalhes de configuração de um ponto de acesso \(p. 332\)](#)
- [Usar um ponto de acesso \(p. 332\)](#)
- [Visualizar configurações de bloqueio de acesso público para um ponto de acesso \(p. 333\)](#)
- [Editar uma política de ponto de acesso \(p. 333\)](#)
- [Excluir um ponto de acesso \(p. 333\)](#)

Listar pontos de acesso para sua conta

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, no lado esquerdo do console, escolha access points (pontos de acesso).
3. Na página access points (pontos de acesso) em access points (pontos de acesso), selecione a Região da AWS que contém os pontos de acesso que você deseja listar.
4. (Opcional) Pesquise pontos de acesso por nome inserindo um nome no campo de texto ao lado do menu suspenso de Região.
5. Escolha o nome do ponto de acesso que você deseja gerenciar ou usar.

Listar pontos de acesso para um bucket

Para listar todos os pontos de acesso de um único bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, no lado esquerdo do console, selecione Buckets.
3. Na página Buckets, selecione o nome do bucket cujos pontos de acesso você deseja listar.
4. Na página de detalhes do bucket, escolha a guia access points (pontos de acesso).
5. Escolha o nome do ponto de acesso que você deseja gerenciar ou usar.

Visualizar detalhes de configuração de um ponto de acesso

1. Navegue até a página de detalhes do ponto de acesso cujos detalhes deseja ver, conforme descrito em [Listar pontos de acesso para sua conta \(p. 332\)](#).
2. Em access point overview (visão geral do ponto de acesso), exiba os detalhes de configuração e as propriedades para o ponto de acesso selecionado.

Usar um ponto de acesso

1. Navegue até a página de detalhes do ponto de acesso cujos detalhes deseja usar, conforme descrito em [Listar pontos de acesso para sua conta \(p. 332\)](#).
2. Na guia Objects (Objetos) escolha o nome de um objeto ou objetos que deseja acessar por meio do ponto de acesso. Nas páginas de operação do objeto, o console exibe um rótulo acima do nome do bucket que mostra o ponto de acesso que você está usando no momento. Enquanto estiver usando o ponto de acesso, você só pode executar as operações de objeto permitidas pelas permissões do ponto de acesso.

Note

- A exibição do console sempre mostra todos os objetos no bucket. O uso de um ponto de acesso conforme descrito neste procedimento restringe as operações que você pode executar nesses objetos, mas não se você puder ver que eles existem no bucket.

- O Console de Gerenciamento do S3 não oferece suporte ao uso de pontos de acesso da Virtual Private Cloud (VPC) para acessar recursos de bucket. Para acessar recursos de bucket em um ponto de acesso da VPC, use a AWS CLI, os AWS SDKs ou as APIs REST do Amazon S3.

Visualizar configurações de bloqueio de acesso público para um ponto de acesso

1. Navegue até a página de detalhes do ponto de acesso para o ponto de acesso cujas configurações deseja exibir, conforme descrito em [Listar pontos de acesso para sua conta \(p. 332\)](#).
2. Escolha Permissions (Permissões).
3. Em access point policy (Política de ponto de acesso), revise as configurações de bloqueio de acesso público do ponto de acesso.

Note

Você não pode alterar as configurações de bloqueio de acesso público para um ponto de acesso após a criação do ponto de acesso.

Editar uma política de ponto de acesso

1. Navegue até a página de detalhes do ponto de acesso cujas políticas deseja editar, conforme descrito em [Listar pontos de acesso para sua conta \(p. 332\)](#).
2. Escolha Permissions (Permissões).
3. Em access point policy (Política do ponto de acesso), escolha Edit (Editar).
4. Insira a política do ponto de acesso no campo de texto. O console exibe automaticamente o nome do recurso da Amazon (ARN) do ponto de acesso, que você pode usar na política.

Excluir um ponto de acesso

1. Navegue até a lista de pontos de acesso para sua conta ou para um bucket específico, conforme descrito em [Listar pontos de acesso para sua conta \(p. 332\)](#).
2. Selecione o botão de opção ao lado do nome do ponto de acesso que você deseja excluir.
3. Escolha Delete (Excluir).
4. Confirme se deseja excluir o ponto de acesso inserindo o nome no campo de texto exibido e escolha Delete (Excluir).

Usar um alias em estilo de bucket para seu ponto de acesso

Quando você cria um ponto de acesso, o Amazon S3 gera automaticamente um alias que poderá ser usado no lugar de um nome de bucket do Amazon S3 para acesso a dados. É possível usar esse alias de ponto de acesso em vez de um nome do recurso da Amazon (ARN) para qualquer operação de plano de dados do ponto de acesso. Para obter uma lista dessas operações, consulte [Compatibilidade com ponto de acesso com produtos da AWS \(p. 335\)](#).

Abaixo, há um exemplo de ARN e de alias de ponto de acesso para um ponto de acesso chamado *my-access-point*.

- ARN — `arn:aws:s3:region:account-id:accesspoint/my-access-point`
- Alias do ponto de acesso: `my-access-point-hrzrlukc5m36ft7okagg1f3gmwluquse1b-s3alias`

Para obter mais informações sobre os ARNs, consulte [Nomes de recurso da Amazon \(ARNs\)](#) na Referência geral da AWS.

Nomes de alias de ponto de acesso

Cria-se um nome de alias de ponto de acesso dentro do mesmo namespace de um bucket do Amazon S3. Esse nome de alias é gerado automaticamente, e não é possível alterá-lo. O nome de alias de ponto de acesso atende a todos os requisitos de um nome de bucket válido do Amazon S3 e consiste nas seguintes partes:

`[Access point prefix]-[Metadata]-s3alias`

Note

O sufixo `-s3alias` é reservado para nomes de alias de ponto de acesso e não pode ser usado para nomes de bucket ou de ponto de acesso. Para obter mais informações sobre as regras para nomes de bucket do Amazon S3, consulte [Regras de nomeação de bucket \(p. 127\)](#).

Casos de uso e limitações de alias de ponto de acesso

Ao adotar pontos de acesso, é possível usar nomes de alias de ponto de acesso sem exigir alterações extensas de código.

Quando você cria um ponto de acesso, o Amazon S3 gera automaticamente um nome de alias de ponto de acesso, conforme o exemplo a seguir.

```
aws s3control create-access-point --bucket DOC-EXAMPLE-BUCKET1 --name my-access-point --  
account-id 111122223333  
{  
    "AccessPointArn":  
        "arn:aws:s3:region:111122223333:accesspoint/my-access-point",  
    "Alias": "my-access-point-aqfqprnstaefdfbarligizwgyfouse1a-s3alias"  
}
```

É possível usar esse alias de ponto de acesso em vez de um nome de bucket do Amazon S3 para qualquer operação de plano de dados. Para obter uma lista dessas operações, consulte [Compatibilidade com ponto de acesso com produtos da AWS \(p. 335\)](#).

```
aws s3api get-object --bucket my-access-point-aqfqprnstaefdfbarligizwgyfouse1a-s3alias --  
key dir/my_data.rtf my_data.rtf  
{  
    "AcceptRanges": "bytes",  
    "LastModified": "2020-01-08T22:16:28+00:00",  
    "ContentLength": 910,  
    "ETag": "\"00751974dc146b76404bb7290f8f51bb\"",  
    "VersionId": "null",  
    "ContentType": "text/rtf",  
    "Metadata": {}  
}
```

Limitações

- Os aliases não podem ser configurados por clientes.
- Não é possível excluir, modificar ou desabilitar aliases de um ponto de acesso.

- É possível usar esse alias de ponto de acesso em vez de um nome de bucket do Amazon S3 em algumas operações de plano de dados. Para obter uma lista dessas operações, consulte [Compatibilidade de ponto de acesso com operações do S3 \(p. 335\)](#).
- Não é possível usar um nome de alias de ponto de acesso para operações do plano de controle do Amazon S3. Para obter uma lista de operações do plano de controle do Amazon S3, consulte [Controle do Amazon S3](#) na Referência de APIs do Amazon Simple Storage Service.
- Não é possível usar aliases em políticas do IAM.
- Não é possível usar aliases como destino de log para logs de acesso do servidor S3.
- Não é possível usar aliases como destino de log para logs do AWS CloudTrail.
- O Amazon SageMaker GroundTruth não aceita alias de ponto de acesso.

Usar pontos de acesso com operações compatíveis com o Amazon S3

Os exemplos a seguir demonstram como usar pontos de acesso com operações compatíveis no Amazon S3.

Tópicos

- [Compatibilidade com ponto de acesso com produtos da AWS \(p. 335\)](#)
- [Compatibilidade de ponto de acesso com operações do S3 \(p. 335\)](#)
- [Solicitar um objeto por meio de um ponto de acesso \(p. 336\)](#)
- [Fazer upload de um objeto por meio de um alias de ponto de acesso \(p. 336\)](#)
- [Excluir um objeto por meio de um ponto de acesso \(p. 337\)](#)
- [Listar objetos por meio de um alias de ponto de acesso \(p. 337\)](#)
- [Adicionar um conjunto de tags a um objeto por meio de um ponto de acesso \(p. 337\)](#)
- [Conceder permissões de acesso por meio de um ponto de acesso usando uma ACL \(p. 337\)](#)

Compatibilidade com ponto de acesso com produtos da AWS

Os aliases de pontos de acesso do Amazon S3 permitem que qualquer aplicação que exija um nome de bucket do S3 use um ponto de acesso com facilidade. É possível usar aliases de ponto de acesso do S3 em qualquer lugar em que você use nomes de bucket do S3 para acessar dados no S3.

Compatibilidade de ponto de acesso com operações do S3

Você pode usar pontos de acesso para acessar um bucket usando o seguinte subconjunto de APIs do Amazon S3. Todas as operações listadas abaixo podem aceitar ARNs de ponto de acesso ou aliases de ponto de acesso:

Operações do S3

- [AbortMultipartUpload](#)
- [CompleteMultipartUpload](#)
- [CopyObject](#) (somente cópias da mesma região)
- [CreateMultipartUpload](#)
- [DeleteObject](#)
- [DeleteObjectTagging](#)
- [GetBucketAcl](#)

- [GetBucketCors](#)
- [GetBucketLocation](#)
- [GetBucketNotificationConfiguration](#)
- [GetBucketPolicy](#)
- [GetObject](#)
- [GetObjectAcl](#)
- [GetObjectLegalHold](#)
- [GetObjectRetention](#)
- [GetObjectTagging](#)
- [HeadBucket](#)
- [HeadObject](#)
- [ListMultipartUploads](#)
- [ListObjects](#)
- [ListObjectsV2](#)
- [ListParts](#)
- [Presign](#)
- [PutObject](#)
- [PutObjectLegalHold](#)
- [PutObjectRetention](#)
- [PutObjectAcl](#)
- [PutObjectTagging](#)
- [RestoreObject](#)
- [UploadPart](#)
- [UploadPartCopy](#) (somente cópias da mesma região)

Solicitar um objeto por meio de um ponto de acesso

O exemplo a seguir solicita o objeto `my-image.jpg` por meio do ponto de acesso `prod` pertencente ao ID de conta `123456789012` na região `us-west-2` e salva o arquivo obtido por download como `download.jpg`.

AWS CLI

```
aws s3api get-object --key my-image.jpg --bucket arn:aws:s3:us-west-2:123456789012:accesspoint/prod download.jpg
```

Fazer upload de um objeto por meio de um alias de ponto de acesso

O exemplo a seguir faz upload do objeto `my-image.jpg` por meio do alias de ponto de acesso `my-access-point-hrzrlukc5m36ft7okagg1f3gmwluquuse1b-s3alias` pertencente ao ID de conta `123456789012` na região `us-west-2`.

AWS CLI

```
aws s3api put-object --bucket my-access-point-hrzrlukc5m36ft7okagg1f3gmwluquuse1b-s3alias --key my-image.jpg --body my-image.jpg
```

Excluir um objeto por meio de um ponto de acesso

O exemplo a seguir exclui o objeto `my-image.jpg` por meio do ponto de acesso `prod` pertencente ao ID de conta `123456789012` na região `us-west-2`.

AWS CLI

```
aws s3api delete-object --bucket arn:aws:s3:us-west-2:123456789012:accesspoint/prod --key my-image.jpg
```

Listar objetos por meio de um alias de ponto de acesso

O exemplo a seguir lista objetos por meio do alias de ponto de acesso `my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias` pertencente ao ID de conta `123456789012` na região `us-west-2`.

AWS CLI

```
aws s3api list-objects-v2 --bucket my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias
```

Adicionar um conjunto de tags a um objeto por meio de um ponto de acesso

O exemplo a seguir adiciona um conjunto de tags ao objeto existente `my-image.jpg` por meio do ponto de acesso `prod` pertencente ao ID de conta `123456789012` na região `us-west-2`.

AWS CLI

```
aws s3api put-object-tagging --bucket arn:aws:s3:us-west-2:123456789012:accesspoint/prod --key my-image.jpg --tagging TagSet=[{Key="finance",Value="true"}]
```

Conceder permissões de acesso por meio de um ponto de acesso usando uma ACL

O exemplo a seguir aplica uma ACL a um objeto existente `my-image.jpg` por meio do ponto de acesso `prod` pertencente ao ID de conta `123456789012` na região `us-west-2`.

AWS CLI

```
aws s3api put-object-acl --bucket arn:aws:s3:us-west-2:123456789012:accesspoint/prod --key my-image.jpg --acl private
```

Restrições e limitações de pontos de acesso

Os pontos de acesso do Amazon S3 têm as seguintes restrições e limitações:

- Só é possível criar pontos de acesso para buckets que você possui.

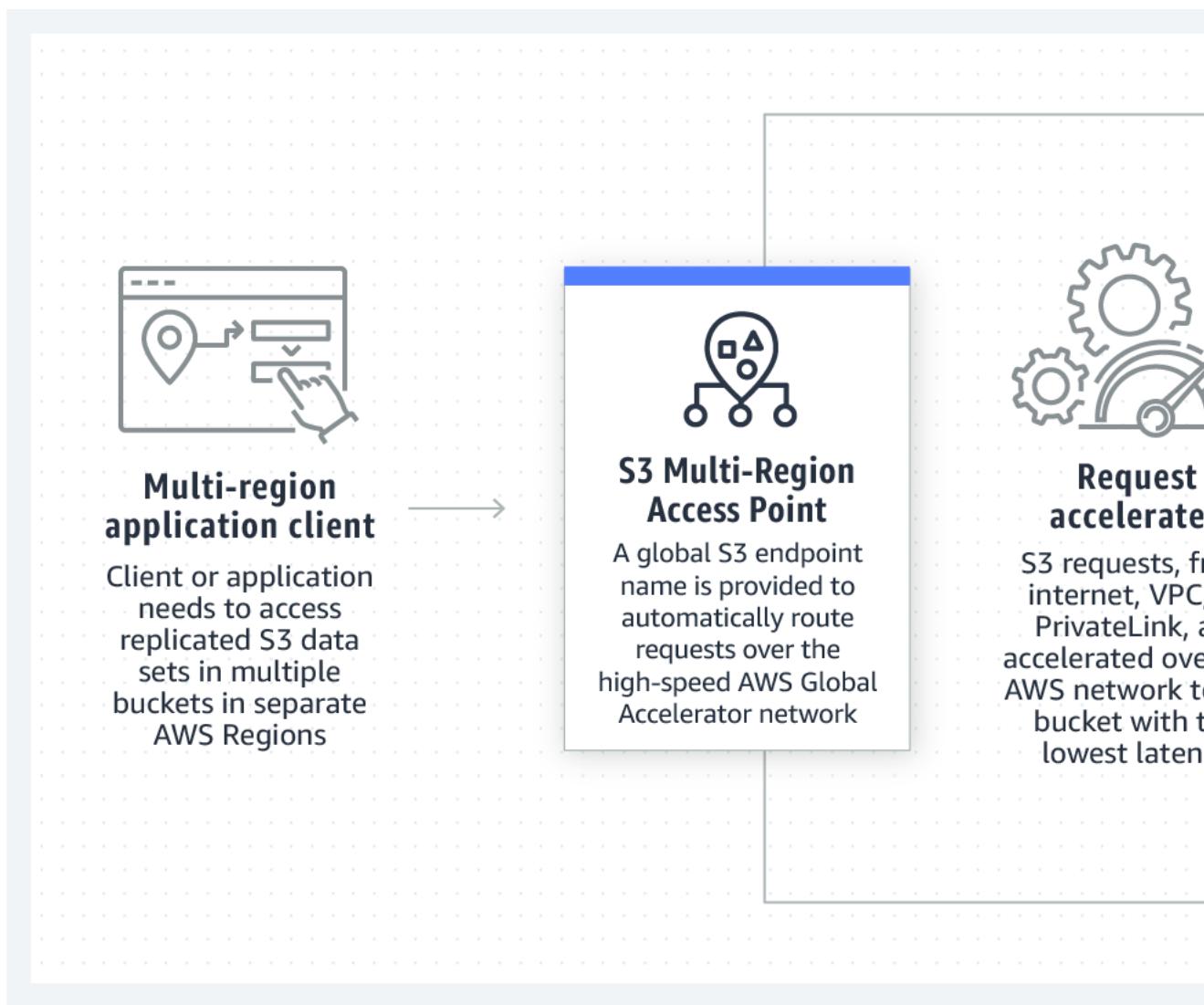
- Cada ponto de acesso é associado a exatamente um bucket, que é necessário especificar ao criar o ponto de acesso. Depois de criar um ponto de acesso, não é possível associá-lo a um bucket diferente. No entanto, você pode excluir um ponto de acesso e criar outro com o mesmo nome associado a um bucket diferente.
- Os nomes dos pontos de acesso devem atender a certas condições. Para obter mais informações sobre nomenclatura de pontos de acesso, consulte [Regras para nomear pontos de acesso do Amazon S3 \(p. 325\)](#).
- Depois de criar um ponto de acesso, não é possível alterar sua configuração de nuvem privada virtual (VPC).
- As políticas de ponto de acesso estão limitadas a 20 KB.
- É possível criar no máximo 10 mil pontos de acesso por Conta da AWS por região. Se você precisar de mais de 10 mil pontos de acesso para uma única conta em uma única região, poderá solicitar um aumento de cota de serviço. Para obter mais informações sobre cotas de serviço e solicitar um aumento, consulte [Service Quotas da AWS](#) na Referência geral da AWS.
- Não é possível usar um ponto de acesso como destino para a replicação do S3. Para obter mais informações sobre a replicação, consulte [Replicação de objetos \(p. 801\)](#).
- Só é possível endereçar pontos de acesso usando URLs em estilo de host virtual. Para obter mais informações sobre o endereçamento no estilo de hospedagem virtual, consulte [Métodos de acesso a um bucket \(p. 135\)](#).
- As APIs que controlam a funcionalidade de ponto de acesso (por exemplo, `PutAccessPoint` e `GetAccessPointPolicy`) não oferecem suporte a chamadas entre contas.
- É necessário usar o AWS Signature Version 4 ao fazer solicitações a um ponto de acesso usando as APIs REST. Para obter mais informações sobre como autenticar solicitações, consulte [Autenticação de solicitações \(AWS Signature Version 4\)](#) na Referência de APIs do Amazon Simple Storage Service.
- Os pontos de acesso só oferecem suporte ao acesso por meio de HTTPS.
- Os pontos de acesso não oferecem suporte ao acesso anônimo.

Pontos de acesso de várias regiões no Amazon S3

Os pontos de acesso de várias regiões do Amazon S3 fornecem um endpoint global que as aplicações podem usar para atender a solicitações de buckets do S3 localizados em várias regiões da AWS. Você pode usar pontos de acesso de várias regiões para criar aplicações de várias regiões com a mesma arquitetura simples usada em uma única região e, em seguida, executar essas aplicações em qualquer lugar do mundo. Em vez de enviar solicitações pela Internet pública congestionada, os pontos de acesso de várias regiões fornecem resiliência de rede integrada com aceleração de solicitações baseadas na Internet para o Amazon S3. Solicitações de aplicações feitas para um endpoint global de várias regiões usam [AWS Global Accelerator](#) para rotear automaticamente na AWS rede global para o bucket S3 com a menor latência de rede.

Ao criar um ponto de acesso de várias regiões, você especifica um conjunto de regiões onde deseja armazenar dados a serem atendidos por meio desse ponto de acesso de várias regiões. Você pode usar [Replicação entre regiões do S3 \(CRR\)](#) para sincronizar dados entre buckets nessas regiões. Em seguida, você pode solicitar ou gravar dados por meio do endpoint global do ponto de acesso de várias regiões. O Amazon S3 atende automaticamente à solicitação para o conjunto de dados replicado da região disponível na rede global da AWS com a menor latência. Os pontos de acesso de várias regiões também são compatíveis com aplicações executadas em nuvens privadas virtuais (VPCs) da Amazon, incluindo aquelas que usam os [AWS PrivateLink for Amazon S3 \(p. 410\)](#).

Veja a seguir uma representação gráfica de um ponto de acesso de várias regiões e como ele roteia solicitações para os buckets.



Tópicos

- [Criação de pontos de acesso multirregionais \(p. 340\)](#)
- [Como fazer solicitações usando um ponto de acesso multirregional \(p. 346\)](#)
- [Gerenciamento de pontos de acesso de várias regiões \(p. 352\)](#)
- [Monitoramento e registro de solicitações feitas por meio de um ponto de acesso multirregional para recursos subjacentes \(p. 353\)](#)
- [Restrições e limitações de pontos de acesso de várias regiões \(p. 355\)](#)

Criação de pontos de acesso multirregionais

Para criar um ponto de acesso multirregional no Amazon S3, especifique o nome, escolha um bucket em cada Região da AWS que você deseja atender a solicitações para o ponto de acesso multirregional e configure as configurações de acesso público de bloco do Amazon S3 para o ponto de acesso multirregional. Você fornece essas informações em uma solicitação de criação, que o Amazon S3 processa de forma assíncrona. O Amazon S3 fornece um token que você pode usar para monitorar o status da solicitação de criação assíncrona.

Solucionar avisos de segurança, erros, avisos gerais e sugestões de AWS Identity and Access Management Access Analyzer antes de salvar a política. O IAM Access Analyzer executa verificações de política para validar sua política em relação à [gramática das políticas](#) e às [práticas recomendadas](#) do IAM. Essas verificações geram descobertas e fornecem recomendações práticas que ajudam a criar políticas que sejam funcionais e estejam em conformidade com as práticas recomendadas de segurança. Para saber mais sobre a validação de políticas usando o IAM Access Analyzer, consulte [Validação de políticas do IAM Access Analyzer](#) no Guia do usuário IAM. Para visualizar uma lista de avisos, erros e sugestões retornados pelo IAM Access Analyzer, consulte [Referência de verificação de políticas do IAM Access Analyzer](#).

Quando você usa a API, a solicitação para criar um ponto de acesso multirregional é assíncrona. Quando você envia uma solicitação para criar um ponto de acesso multirregional, o Amazon S3 autoriza a solicitação de forma síncrona. Em seguida, retorna imediatamente um token que você pode usar para rastrear o progresso da solicitação de criação. Para obter mais informações sobre o rastreamento de solicitações assíncronas para criar e gerenciar pontos de acesso multirregionais, consulte [Gerenciamento de pontos de acesso de várias regiões \(p. 352\)](#).

Depois de criar o ponto de acesso multirregional, você pode criar uma política de controle de acesso para ele. Cada ponto de acesso multirregional pode ter uma política associada. Uma política de pontos de acesso multirregionais é uma política baseada em recursos que permite limitar o uso do ponto de acesso multirregional por recurso, usuário ou outras condições.

Note

Para que uma aplicação ou usuário possa acessar um objeto por meio de um ponto de acesso multirregional, a política de acesso para o ponto de acesso multirregional e a política de acesso para os buckets subjacentes que contêm o objeto devem permitir a solicitação. Quando as duas políticas são diferentes, a política mais restritiva tem precedência.

Usar um bucket com um ponto de acesso multirregional não altera o comportamento do bucket quando o bucket é acessado por meio do nome do bucket existente ou de um nome de recurso da Amazon (ARN). Todas as operações existentes no bucket continuarão a funcionar como antes. As restrições que você incluir em uma política de ponto de acesso multirregional se aplicam somente a solicitações feitas por meio desse ponto de acesso multirregional.

Você pode atualizar a política de um ponto de acesso multirregional depois de criá-lo, mas não pode excluir a política. A aproximação mais próxima possível para excluir uma política é atualizar a política de ponto de acesso multirregional para negar todas as permissões.

Tópicos

- [Regras para nomear pontos de acesso multirregionais do Amazon S3 \(p. 341\)](#)
- [Regras para escolher buckets para pontos de acesso multirregionais do Amazon S3 \(p. 342\)](#)
- [Bloqueio de acesso público de pontos de acesso multirregionais do Amazon S3 \(p. 343\)](#)
- [Criação de pontos de acesso multirregionais do Amazon S3 \(p. 343\)](#)
- [Configurar um ponto de acesso multirregional para uso com AWS PrivateLink \(p. 344\)](#)

Regras para nomear pontos de acesso multirregionais do Amazon S3

Ao criar um ponto de acesso multirregional, atribua a ele um nome, que é uma string escolhida. Não é possível alterar o nome do ponto de acesso multirregional depois que ele for criado. O nome deve ser exclusivo na Conta da AWS, e deve estar em conformidade com os requisitos de nomenclatura listados em [Restrições e limitações de pontos de acesso de várias regiões \(p. 355\)](#). Para ajudá-lo a identificar o ponto de acesso multirregional, use um nome que seja significativo para você, para sua organização ou que reflita o cenário.

Use esse nome ao chamar operações de gerenciamento de pontos de acesso multirregionais, como `GetMultiRegionAccessPoint` e `PutMultiRegionAccessPointPolicy`. O nome não é usado para enviar solicitações ao ponto de acesso multirregional e não precisa ser exposto a clientes que fazem solicitações usando o ponto de acesso multirregional.

Quando o Amazon S3 cria um ponto de acesso multirregional, ele atribui automaticamente um alias a ele. Este alias é uma string alfanumérica exclusiva que termina em `.mrap`. O alias é usado para construir o nome do host e o nome do recurso da Amazon (ARN) para um ponto de acesso multirregional. O nome totalmente qualificado também se baseia no alias do ponto de acesso multirregional.

Não é possível determinar o nome de um ponto de acesso multirregional de seu alias, portanto, você pode divulgar um alias sem o risco de expor o nome, a finalidade ou o proprietário do ponto de acesso multirregional. O Amazon S3 seleciona o alias para cada novo ponto de acesso multirregional, e o alias não pode ser alterado. Para obter mais informações sobre como lidar com um ponto de acesso multirregional, consulte [Como fazer solicitações usando um ponto de acesso multirregional \(p. 346\)](#).

Os aliases de pontos de acesso multirregionais são exclusivos ao longo do tempo e não se baseiam no nome ou na configuração de um ponto de acesso multirregional. Se você criar um ponto de acesso multirregional, excluí-lo e criar outro com o mesmo nome e configuração, o segundo ponto de acesso multirregional terá um alias diferente do primeiro. Novos pontos de acesso multirregionais nunca podem ter o mesmo alias de um ponto de acesso multirregional anterior.

Regras para escolher buckets para pontos de acesso multirregionais do Amazon S3

Cada ponto de acesso multirregional está associado às regiões nas quais você deseja atender a solicitações. O ponto de acesso multirregional deve estar associado a exatamente um bucket em cada uma dessas regiões. Especifique o nome de cada bucket na solicitação para criar o ponto de acesso multirregional. Cada bucket que suporta o ponto de acesso multirregional deve ser de propriedade da mesma Conta da AWS que possui o ponto de acesso multirregional.

Um único bucket pode ser usado por vários pontos de acesso multirregionais.

Important

- Você pode especificar os buckets associados a um ponto de acesso multirregional somente no momento em que o criou. Depois que ele é criado, você não pode adicionar, modificar ou remover buckets da configuração do ponto de acesso multirregional. Para alterar os buckets, você deve excluir todo o ponto de acesso multirregional e criar um novo.
- Não é possível excluir um bucket que faça parte de um ponto de acesso multirregional. Se quiser excluir um bucket anexado a um ponto de acesso multirregional, exclua primeiro o ponto de acesso multirregional.
- A Conta da AWS que possui o ponto de acesso multirregional também deve possuir os buckets associados. Para obter mais informações sobre como usar as permissões com pontos de acesso multirregionais, consulte [Permissões do ponto de acesso multirregional \(p. 348\)](#).
- Nem todas as regiões suportam pontos de acesso multirregionais. Para visualizar a lista de regiões compatíveis, consulte [Restrições e limitações de pontos de acesso de várias regiões \(p. 355\)](#).

Você pode criar regras de replicação para sincronizar dados entre buckets. Essas regras permitem que você copie automaticamente dados de buckets de origem para buckets de destino. Ter buckets conectados a um ponto de acesso multirregional não afeta o funcionamento da replicação. A configuração da replicação com pontos de acesso multirregionais é descrita em uma seção posterior.

É importante perceber que quando você faz uma solicitação para um ponto de acesso multirregional, o ponto de acesso multirregional não faz nenhuma consideração sobre qual bucket pode atender à solicitação. É por isso que a replicação é recomendada. Caso contrário, um dos buckets no ponto de

acesso multirregional pode ter os dados necessários, mas não há como garantir que ele receberá a solicitação. Para obter mais informações, consulte [Como configurar a replicação de bucket para uso com pontos de acesso multirregionais \(p. 351\)](#).

Bloqueio de acesso público de pontos de acesso multirregionais do Amazon S3

Cada ponto de acesso multirregional tem configurações distintas para o bloqueio de acesso público do Amazon S3. Essas configurações operam em conjunto com as configurações de bloqueio de acesso público para os buckets subjacentes ao ponto de acesso multirregional e para a Conta da AWS que possui o ponto de acesso multirregional e os buckets subjacentes.

Quando o Amazon S3 autoriza uma solicitação, ele aplica a combinação mais restritiva dessas configurações. Se as configurações de bloqueio de acesso público para qualquer um desses recursos (o ponto de acesso multirregional, o bucket subjacente ou a conta de proprietário) bloquearem o acesso para a ação ou recurso solicitado, o Amazon S3 rejeitará a solicitação.

Recomendamos que você ative todas as configurações de bloqueio de acesso público, a menos que você tenha uma necessidade específica de desabilitar qualquer uma delas. Por padrão, todas as configurações de bloqueio de acesso público são habilitadas para um ponto de acesso multirregional. Saiba que, se o bloqueio de acesso público estiver ativado, o ponto de acesso multirregional não poderá aceitar solicitações baseadas na Internet.

Important

No momento, o Amazon S3 não oferece suporte à alteração das configurações de bloqueio de acesso público para um ponto de acesso multirregional após ter sido criado.

Para obter mais informações sobre o bloqueio de acesso público do Amazon S3, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

Criação de pontos de acesso multirregionais do Amazon S3

Os exemplos a seguir demonstram como criar um ponto de acesso multirregional usando a AWS Management Console.

Uso do console do S3

Para criar um ponto de acesso multirregional

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Multi-Region Access Points (Pontos de acesso multirregionais).
3. No campo Multi-Region Access Point name (Nome do ponto de acesso multirregional), forneça um nome para o ponto de acesso multirregional.
4. Para selecionar os buckets que serão associados a esse ponto de acesso multirregional, escolha Add buckets (Adicionar buckets).

Para criar um novo bucket, escolha Create Bucket (Criar bucket). Depois de criar o bucket, escolha Add buckets (Adicionar buckets) para adicionar o bucket ao ponto de acesso multirregional.

Para obter mais informações sobre a criação de buckets, consulte [Criação de um bucket \(p. 129\)](#).

5. Em Block Public Access settings for this Multi-Region Access Point (Configurações do bloqueio de acesso público para este ponto de acesso multirregional), selecione as configurações de bloqueio de acesso público que você deseja aplicar ao ponto de acesso multirregional. Por padrão, todas

as configurações de bloqueio de acesso público são habilitadas para novos pontos de acesso multirregionais. Recomendamos que você deixe todas as configurações ativadas, a menos que saiba que tem uma necessidade específica de desabilitar qualquer uma delas.

Note

Atualmente, o Amazon S3 não oferece suporte à alteração das configurações de bloqueio de acesso público de um ponto de acesso multirregional após a criação do ponto de acesso multirregional.

6. Selecione Create Multi-Region Access Point (Criar ponto de acesso multirregional).

Uso do AWS CLI

Você pode usar a AWS CLI para criar um ponto de acesso multirregional. Lembre-se de que, ao criar o ponto de acesso multirregional, você precisa fornecer todos os buckets que ele suportará. Não há opção para adicionar buckets ao ponto de acesso multirregional depois de ele ter sido criado.

O exemplo a seguir cria um ponto de acesso multirregional com dois buckets usando a AWS CLI.

```
aws s3control create-multi-region-access-point --account-id 111122223333 --details '{  
    "Name": "simple-multiregionaccesspoint-with-two-regions",  
    "PublicAccessBlock": {  
        "BlockPublicAcls": true,  
        "IgnorePublicAcls": true,  
        "BlockPublicPolicy": true,  
        "RestrictPublicBuckets": true  
    },  
    "Regions": [  
        { "Bucket": "DOC-EXAMPLE-BUCKET1" },  
        { "Bucket": "DOC-EXAMPLE-BUCKET2" }  
    ]  
' --region us-west-2
```

Tópicos

- [Configurar um ponto de acesso multirregional para uso com AWS PrivateLink \(p. 344\)](#)

Configurar um ponto de acesso multirregional para uso com AWS PrivateLink

AWS PrivateLink fornece conectividade privada com o Amazon S3 usando endereços IP privados na nuvem privada virtual (VPC). Você pode provisionar um ou mais endpoints de interface dentro da VPC para se conectar aos pontos de acesso multirregionais do Amazon S3.

Você pode criar endpoints com.amazonaws.s3-global.accesspoint para pontos de acesso multirregionais por meio do AWS Management Console, da AWS CLI ou de AWS SDKs. Para saber mais sobre como configurar um endpoint de interface para o ponto de acesso multirregional, consulte [Endpoints de VPC da interface](#) no Guia do usuário da VPC.

Para fazer solicitações a um ponto de acesso multirregional por meio de endpoints de interface, siga estas etapas para configurar a VPC e o ponto de acesso multirregional.

Para configurar um ponto de acesso multirregional para usar com AWS PrivateLink

1. Crie ou tenha um endpoint de VPC apropriado que possa se conectar a pontos de acesso multirregionais. Para obter mais informações sobre a criação de endpoints de VPC, consulte [Endpoints da VPC da interface](#) no Guia do usuário da VPC.

Important

Certifique-se de criar um endpoint com.amazonaws.s3-global.accesspoint. Outros tipos de endpoint não podem acessar pontos de acesso multirregionais.

Depois que esse endpoint da VPC é criado, todas as solicitações de pontos de acesso multirregionais na VPC são roteadas por esse endpoint, se você tiver o DNS privado habilitado para o endpoint. Esta opção está ativada por padrão.

2. Se a política de ponto de acesso multirregional não oferecer suporte a conexões de endpoints da VPC, você precisará atualizá-la.
3. Verifique se as políticas de bucket individuais permitirão acesso aos usuários do ponto de acesso multirregional.

Lembre-se de que os pontos de acesso multirregionais funcionam roteando solicitações para buckets, não atendendo às próprias solicitações. É importante se lembrar disso porque o originador da solicitação deve ter permissões para o ponto de acesso multirregional e ter permissão para acessar os buckets individuais no ponto de acesso multirregional. Caso contrário, a solicitação pode ser encaminhada para um bucket onde o originador não tem permissões para atender à solicitação. Um ponto de acesso multirregional e os buckets devem pertencer à mesma conta da AWS. No entanto, as VPCs de contas diferentes poderão usar um ponto de acesso multirregional se as permissões estiverem configuradas corretamente.

Por isso, a política de endpoint da VPC deve permitir o acesso ao ponto de acesso multirregional e a cada bucket subjacente que você deseja que possa atender às solicitações. Por exemplo, digamos que você tenha um ponto de acesso multirregional com o alias `mfzwi23gnjvgw.mrap`. É suportado por buckets `doc-examplebucket1` e `doc-examplebucket2`, todos pertencentes à conta `123456789012` da AWS. Nesse caso, a política de VPCE a seguir permitiria solicitações `GetObject` da VPC feitas para `mfzwi23gnjvgw.mrap` a ser atendido por qualquer um dos buckets de suporte.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Read-buckets-and-MRAP-VPCE-policy",  
            "Principal": "*",  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:s3:::doc-examplebucket1/*",  
                "arn:aws:s3:::doc-examplebucket2/*",  
                "arn:aws:s3:::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/*"  
            ]  
        }  
    ]  
}
```

Conforme mencionado anteriormente, você também deve se certificar de que a política de pontos de acesso multirregionais esteja configurada para oferecer suporte ao acesso, por meio de um endpoint da VPC. Você não precisa especificar o endpoint da VPC que está solicitando acesso. A seguinte política de exemplo concederia acesso a qualquer solicitante que tentasse usar o ponto de acesso multirregional para as solicitações de `GetObject`.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Open-read-MRAP-policy",  
            "Effect": "Allow",  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws:s3:::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/*"  
        }  
    ]  
}
```

```
"Principal": "*",
"Action": [
    "s3:GetObject"
],
"Resource": "arn:aws:s3:::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/*"
}]}
```

E, claro, cada um dos buckets individuais precisaria de uma política para dar suporte ao acesso de solicitações enviadas por meio do endpoint da VPC. A política de exemplo a seguir concede acesso de leitura a usuários anônimos, o que incluiria solicitações feitas por meio do endpoint da VPC.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Public-read",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": [
                "arn:aws:s3:::doc-examplebucket1",
                "arn:aws:s3:::doc-examplebucket2/*"
            ]
        }
    ]
}
```

Para obter informações sobre como editar uma política de VPCE, consulte [Controlar o acesso aos serviços com endpoints da VPC](#) no Guia do usuário da VPC.

Remoção do acesso a um ponto de acesso multirregional de um endpoint da VPC

Se você possuir um ponto de acesso multirregional e quiser remover o acesso dele de um endpoint de interface, você deve fornecer uma nova política de acesso ao ponto de acesso multirregional que impede o acesso a solicitações que chegam por meio de endpoints da VPC. Lembre-se de que, se os buckets em seu ponto de acesso multirregional suportarem solicitações por meio de endpoints de VPC, eles continuarão a oferecer suporte a essas solicitações. Se quiser impedir esse suporte, você deverá atualizar também as políticas para os buckets. O fornecimento de uma nova política de acesso ao ponto de acesso multirregional só impede o acesso ao ponto de acesso multirregional.

Note

Não é possível excluir uma política de acesso para um ponto de acesso multirregional. Para remover o acesso a um ponto de acesso multirregional, você deve fornecer uma nova política de acesso com o acesso modificado desejado.

Como alternativa, você pode atualizar as políticas de bucket para evitar solicitações por meio de endpoints da VPC. Nesse caso, o usuário ainda poderia acessar o ponto de acesso multirregional por meio do endpoint da VPC. Porém, se a solicitação for encaminhada para um bucket onde a política de bucket impede o acesso, ela gerará uma mensagem de erro.

Como fazer solicitações usando um ponto de acesso multirregional

Os pontos de acesso multirregionais no Amazon S3 têm Amazon Resource Names (ARNs), que você pode usar para direcionar solicitações a elas usando os SDKs da AWS e identificar um ponto de acesso

multirregional nas políticas de controle de acesso. Um ARN de ponto de acesso multirregional não inclui nem divulga seu nome. Para obter mais informações sobre os ARNs, consulte [Amazon Resource Names \(ARNs\)](#) (Nomes de recurso da Amazon (ARNs) em AWS General Reference (Referência geral).

Os ARNs de ponto de acesso multirregional usam o formato `arn:aws:s3:<account-id>:accesspoint/<MRAP_alias>`. Veja a seguir alguns exemplos.

- `arn:aws:s3:::123456789012:accesspoint/mfzwi23gnjvgw.mrap` representa o ponto de acesso multirregional com o alias `mfzwi23gnjvgw.mrap`, de propriedade da conta `123456789012` da AWS.
- `arn:aws:s3:::123456789012:accesspoint/*` representa todos os pontos de acesso multirregionais na conta `123456789012`. Este ARN corresponde a todos os pontos de acesso multirregionais para a conta `123456789012`, mas não corresponde a nenhum ponto de acesso regional porque o ARN não inclui uma região da AWS. Por outro lado, o ARN `arn:aws:s3:::us-west-2:123456789012:accesspoint/*` corresponde a todos os pontos de acesso regionais na região `us-west-2` para a conta `123456789012`, mas não corresponde a nenhum ponto de acesso multirregional.

Os ARNs para objetos acessados por meio de um ponto de acesso multirregional usam o formato `arn:aws:s3:<account_id>:accesspoint/<MRAP_alias>/object/<key>`. Assim como com ARNs de pontos de acesso multirregionais, os ARNs para objetos acessados por meio de pontos de acesso multirregionais não incluem uma região da AWS. Aqui estão alguns exemplos.

- `arn:aws:s3:::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/unit-01` representa o objeto `unit-01`, acessado por meio do ponto de acesso multirregional com o alias `mfzwi23gnjvgw.mrap`, de propriedade da conta `123456789012`.
- `arn:aws:s3:::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/*` representa todos os objetos que podem ser acessados por meio do ponto de acesso multirregional com alias `mfzwi23gnjvgw.mrap`, na conta `123456789012`.
- `arn:aws:s3:::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/unit-01/finance/*` representa todos os objetos que podem ser acessados sob o prefixo `unit-01/finance/` para o ponto de acesso multirregional com o alias `mfzwi23gnjvgw.mrap`, na conta `123456789012`.

Nomes de host do ponto de acesso multirregional

Você pode acessar dados no Amazon S3 por meio de um ponto de acesso multirregional usando o nome do host do ponto de acesso multirregional. As solicitações podem ser direcionadas para esse nome de host da Internet pública ou de uma nuvem privada virtual (VPC), se você tiver configurado um ou mais gateways de Internet para o ponto de acesso multirregional. Para obter mais informações sobre a criação de endpoints de interface da VPC para usar com pontos de acesso multirregionais, consulte [Configurar um ponto de acesso multirregional para uso com AWS PrivateLink \(p. 344\)](#).

Você também pode fazer solicitações por meio de um ponto de acesso multirregional de uma VPC usando AWS PrivateLink, se você configurou um endpoint da VPC. Esteja ciente de que, com solicitações para um ponto de acesso multirregional usando AWS PrivateLink, você não pode usar diretamente um DNS regional específico do endpoint terminando com `<Region>.vpce.amazonaws.com`. Este nome de host não terá um certificado associado a ele, portanto, ele não pode ser usado diretamente. Você ainda pode usar o nome DNS público do endpoint da VPC como um destino de CNAME ou de ALIAS. Como alternativa, você pode habilitar o DNS privado no endpoint e usar os nomes de DNS `<MRAP_alias>.accesspoint.s3-global.amazonaws.com` do ponto de acesso multirregional padrão, conforme descrito abaixo.

Quando você usa as APIs REST para operações de dados do Amazon S3 (por exemplo, `GetObject`) por meio de um ponto de acesso multirregional, o nome do host para a solicitação é `<MRAP_alias>.accesspoint.s3-global.amazonaws.com`. Por exemplo, para fazer uma solicitação de `GetObject` por meio do ponto de acesso multirregional com alias `mfzwi23gnjvgw.mrap`, faça uma

solicitação para o nome do host `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`. Observe a parte `s3-global` do nome do host que indica que esse nome de host não é para uma região específica.

Fazer solicitações por meio de um ponto de acesso multirregional é semelhante a fazer solicitações por meio de um ponto de acesso de região única. É importante estar ciente das seguintes diferenças:

- ARNs de pontos de acesso multirregionais não incluem uma região da AWS. Eles seguem o formato `arn:aws:s3:::<account-id>:accesspoint/<MRAP_alias>`.
- Para solicitações feitas por meio das APIs REST (isso não requer o uso de um ARN), os pontos de acesso multirregionais usam um esquema de endpoint diferente. O esquema é `<MRAP_alias>.accesspoint.s3-global.amazonaws.com`, por exemplo, `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`. Observe as diferenças em comparação com um ponto de acesso de região única:
 - Os nomes de host de pontos de acesso multirregionais usam seu alias, não o nome do ponto de acesso multirregional.
 - Os nomes de host do ponto de acesso multirregional não incluem o ID da conta da AWS dos proprietários.
 - Os nomes de host do ponto de acesso multirregional não incluem uma região da AWS.
 - Os nomes de host do ponto de acesso multirregional incluem `s3-global.amazonaws.com`, em vez de `s3.amazonaws.com`.
- As solicitações devem ser assinadas usando o Signature versão 4A (SigV4a). Quando você usa o SDK da AWS, o SDK converte automaticamente uma assinatura Sigv4 em SigV4a. Para obter mais informações sobre SigV4A, consulte [Assinatura de solicitações de API da AWS](#) na Referência geral da AWS.

Pontos de acesso multirregionais e Amazon S3 Transfer Acceleration

O Amazon S3 Transfer Acceleration é um recurso que permite transferências de dados rápidas para os buckets. Ele é configurado no nível de bucket individual e você pode usá-lo para transferir objetos mais rapidamente para buckets. Para ler mais sobre o Transfer Acceleration, consulte [Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration \(p. 147\)](#).

Ao lidar com pontos de acesso multirregionais, é importante saber que eles usam um mecanismo de transferência acelerada semelhante ao do Transfer Acceleration para enviar objetos grandes pela rede da AWS. Por isso, você não precisa executar nenhuma configuração ou tratamento especial para obter os benefícios das taxas de transferência mais rápidas ao enviar solicitações por meio de um ponto de acesso multirregional. Essa performance aprimorada é incorporada automaticamente ao ponto de acesso multirregional.

Tópicos

- [Permissões do ponto de acesso multirregional \(p. 348\)](#)
- [Roteamento de solicitação de pontos de acesso multirregionais \(p. 350\)](#)
- [Como configurar a replicação de bucket para uso com pontos de acesso multirregionais \(p. 351\)](#)
- [Operações suportadas por pontos de acesso multirregionais \(p. 352\)](#)

Permissões do ponto de acesso multirregional

Quando você faz uma solicitação por meio de um ponto de acesso multirregional, o Amazon S3 autoriza a solicitação no ponto de acesso multirregional e no bucket subjacente para o qual a solicitação é roteada.

Assim, para que uma solicitação seja bem-sucedida, tanto o ponto de acesso multirregional e pelo menos um bucket subjacente devem permitir a operação.

Por exemplo, suponha que você faça uma solicitação de `GetObject` por meio de um ponto de acesso multirregional usando um usuário chamado `AppDataReader` em sua conta da AWS. Para ajudar a garantir que a solicitação não será negada, o usuário `AppDataReader` deve receber a permissão `s3:GetObject` pelo ponto de acesso e várias regiões e por cada bucket subjacente ao ponto de acesso multirregional. O `AppDataReader` não será capaz de recuperar dados de qualquer bucket que não conceder essa permissão.

Em geral, os buckets subjacentes ainda têm configurações individuais de bloqueio de acesso público do S3, políticas e listas de controle de acesso (ACLs, incluindo ACLs de objeto) que permanecem em vigor em todos os casos.

Como gerenciar o acesso público a um ponto de acesso multirregional

Os pontos de acesso multirregionais oferecem suporte a configurações de bloqueio de acesso público para cada ponto de acesso multirregional. Ao criar um ponto de acesso multirregional, você pode especificar configurações do bloqueio de acesso público que se aplicam a esse ponto de acesso multirregional.

Para qualquer solicitação feita por meio de um ponto de acesso multirregional, o Amazon S3 avalia as configurações de bloqueio de acesso público para esse ponto de acesso multirregional, os buckets subjacentes e a conta que possui o ponto de acesso multirregional e os buckets subjacentes. Se qualquer uma dessas configurações indicar que a solicitação deve ser bloqueada, o Amazon S3 rejeitará a solicitação. Para obter mais informações sobre o recurso de bloqueio de acesso público do Amazon S3, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

Important

Todas as configurações do bloqueio de acesso público são habilitadas por padrão para pontos de acesso multirregionais. Você deve desativar explicitamente todas as configurações que não deseja aplicar a um ponto de acesso multirregional. Atualmente, o Amazon S3 não oferece suporte à alteração das configurações do bloqueio de acesso público para um ponto de acesso multirregional, após sua criação.

Como delegar o controle de acesso a políticas de ponto de acesso multirregional

Você pode delegar o controle de acesso de um bucket à política de acesso do ponto de acesso multirregional. A política de bucket de exemplo a seguir permite acesso total a todos os pontos de acesso pertencentes à conta do proprietário do bucket. Isso significa que todo acesso a esse bucket é controlado pelas políticas anexadas aos seus pontos de acesso. Recomendamos configurar seus buckets dessa maneira para todos os casos de uso que não exigem acesso direto ao bucket.

```
{  
    "Version": "2012-10-17",  
    "Statement" : [  
        {  
            "Effect": "Allow",  
            "Principal" : { "AWS": "*" },  
            "Action" : "*",  
            "Resource" : [ "Bucket ARN", "Bucket ARN/*" ],  
            "Condition": {  
                "StringEquals" : { "s3:DataAccessPointAccount" : "Bucket owner's account ID" }  
            }  
        }  
    ]  
}
```

}

A política de bucket do exemplo a seguir delega o controle de acesso a qualquer um dos pontos de acesso multirregionais do bucket. Se você quiser delegar acesso a pontos de acesso multirregionais específicos, você pode usar o `s3:DataAccessPointArn`, em vez disso.

```
{  
    "Version": "2012-10-17",  
    "Statement" : [  
        {  
            "Effect": "Allow",  
            "Principal" : { "AWS": "*" },  
            "Action" : "*",  
            "Resource" : [ "Bucket ARN", "Bucket ARN/*"],  
            "Condition": {  
                "StringEquals" : { "s3:DataAccessPointArn" : "MRAP ARN" }  
            }  
        }  
    ]  
}
```

Roteamento de solicitação de pontos de acesso multirregionais

Quando você faz uma solicitação por meio de um ponto de acesso multirregional, o Amazon S3 determina quais dos buckets associados ao ponto de acesso multirregional podem responder à solicitação com a latência mais baixa. Em seguida, o Amazon S3 direciona a solicitação para esse bucket, independentemente da região da AWS no qual está localizado.

Depois que o ponto de acesso multirregional rotear a solicitação para o bucket de latência mais baixa, o Amazon S3 processará a solicitação como se você tivesse feito diretamente para esse bucket. Pontos de acesso multirregionais não estão cientes do conteúdo de dados de um bucket do Amazon S3. Se fizer solicitações GET para um ponto de acesso multirregional, você poderá configurar a replicação entre regiões do S3 para criar conjuntos de dados consistentes nos buckets do Amazon S3 atrás de um ponto de acesso multirregional. Em seguida, qualquer bucket pode cumprir a solicitação GET com êxito.

O Amazon S3 direciona solicitações de pontos de acesso multirregionais de acordo com as seguintes regras:

- O Amazon S3 otimiza solicitações para serem atendidas com a menor latência possível. Ele examina os buckets suportados pelo ponto de acesso multirregional e retransmite a solicitação para o bucket que tem a menor latência.
- Se a solicitação especificar um recurso existente (por exemplo, `GetObject`), o Amazon S3 não considera o nome do objeto ao atender a solicitação. Isso significa que um objeto pode existir em um bucket no ponto de acesso multirregional, mas sua solicitação será roteada para um bucket que não contém o objeto. Isso resultará em uma mensagem de erro 404 retornada ao cliente. Para garantir que suas solicitações sejam atendidas usando os objetos específicos que você deseja, recomendamos que você ative o controle de versão do bucket e inclua IDs de versão em suas solicitações. Isso ajuda a garantir que você tenha a versão correta do objeto que você está procurando.

Também recomendamos que configure a replicação para seus buckets. Isso ajuda a resolver o problema potencial quando o objeto desejado estiver em um bucket no ponto de acesso multirregional, mas ele não está localizado no bucket específico para o qual sua solicitação foi roteada. Para obter mais informações sobre como configurar a replicação, consulte [Como configurar a replicação de bucket para uso com pontos de acesso multirregionais \(p. 351\)](#).

- Se a solicitação for criar um recurso (por exemplo, `PutObject` ou `CreateMultipartUpload`), o Amazon S3 preenche a solicitação usando o bucket de latência mais baixa. Por exemplo, pense em

uma empresa de vídeo que deseja oferecer suporte a carregamentos de vídeo de qualquer lugar do mundo para o bucket com a menor latência. Quando um usuário faz uma solicitação `PUT` para o ponto de acesso multirregional, o objeto é colocado no bucket com a latência mais baixa. Isso demonstra uma das razões pelas quais a replicação bidirecional pode ser importante. Para obter mais informações sobre a replicação com pontos de acesso multirregionais, consulte [Como configurar a replicação de bucket para uso com pontos de acesso multirregionais \(p. 351\)](#)

Como configurar a replicação de bucket para uso com pontos de acesso multirregionais

Quando você faz uma solicitação para um endpoint do ponto de acesso multirregional, o Amazon S3 roteia automaticamente a solicitação para o bucket que responde à solicitação com a latência mais baixa. Não considera o conteúdo da solicitação ao tomar essa decisão. Se você fizer uma solicitação para `GET` um objeto, sua solicitação poderá ser roteada para um bucket que não tenha uma cópia desse objeto. Se isso acontecer, você receberá um erro 404. Se você quiser que o ponto de acesso multirregional consiga recuperar o objeto independentemente de qual bucket receba a solicitação, configure a replicação entre regiões do Amazon S3.

Considere um ponto de acesso multirregional com três buckets:

- Um bucket nomeado `my-bucket-usw2` na região `us-west-2` que contém o objeto `my-image.jpg`.
- Um bucket nomeado `my-bucket-aps1` na região `ap-south-1` que contém o objeto `my-image.jpg`.
- Um bucket nomeado `my-bucket-euc1` na região `eu-central-1` que não contém um objeto `my-image.jpg`.

Nessa situação, se você fizer uma solicitação `GetObject` para o objeto `my-image.jpg`, o sucesso dessa solicitação dependerá de qual bucket receberá sua solicitação. Como o Amazon S3 não considera o conteúdo da solicitação, ele pode rotear sua solicitação `GetObject` para o bucket `my-bucket-euc1`, se esse bucket responder com a latência mais baixa. Mesmo que seu objeto esteja em um bucket no ponto de acesso multirregional, você receberá um erro 404 porque o bucket individual que recebeu sua solicitação não tinha o objeto.

A habilitação da replicação ajuda a atenuar esse resultado. Com regras de replicação apropriadas, a imagem `my-image.jpg` é copiada para o `my-bucket-euc1`, o que significa que você recuperará o objeto se o Amazon S3 rotear sua solicitação para esse bucket.

A replicação funciona normalmente com buckets atribuídos a um ponto de acesso multirregional. O Amazon S3 não executa nenhum tratamento especial com buckets que estão em pontos de acesso multirregionais. O Amazon S3 fornece opções de replicação 1:N e N:N para sincronização flexível entre buckets. Para obter mais informações sobre como configurar a replicação em seus buckets, consulte [Configuração da replicação \(p. 807\)](#).

Temos algumas recomendações, se você quiser ter a melhor performance de replicação ao trabalhar com pontos de acesso multirregionais. Primeiro, recomendamos configurar o Controle de tempo de replicação do S3 (S3 RTC), mas saiba que isso tem um custo adicional. Para obter mais informações sobre o Controle do tempo de replicação do S3, consulte [Atendimento aos requisitos de conformidade usando o Controle do tempo de replicação do S3 \(S3 RTC\) \(p. 860\)](#). Também recomendamos habilitar a replicação bidirecional para suportar a manutenção de buckets sincronizados quando um bucket é atualizado por meio do ponto de acesso multirregional. Por fim, você deve habilitar as métricas do Amazon CloudWatch para monitorar os eventos de replicação.

Warning

Se você usar o AWS Management Console para criar regras de replicação no console do ponto de acesso multirregional, quaisquer configurações de replicação preexistentes nos buckets

especificados serão substituídas. Se você quiser adicionar ou modificar configurações de replicação existentes em vez de substituí-las, você pode modificar as regras usando a página de configuração de replicação de cada bucket no console ou usando a AWS CLI, os SDKs ou a API REST. Para obter mais informações sobre como modificar configurações de replicação, consulte [Configuração de replicação \(p. 808\)](#).

Operações suportadas por pontos de acesso multirregionais

Você pode usar pontos de acesso multirregionais para acessar um bucket usando o seguinte subconjunto de APIs do Amazon S3:

- [AbortMultipartUpload](#)
- [CompleteMultipartUpload](#)
- [CreateMultipartUpload](#)
- [DeleteObject](#)
- [DeleteObjectTagging](#)
- [GetObject](#)
- [GetObjectAcl](#)
- [GetObjectLegalHold](#)
- [GetObjectRetention](#)
- [GetObjectTagging](#)
- [HeadObject](#)
- [ListMultipartUploads](#)
- [ListObjectsV2](#)
- [ListParts](#)
- [PutObject](#)
- [PutObjectAcl](#)
- [PutObjectLegalHold](#)
- [PutObjectRetention](#)
- [PutObjectTagging](#)
- [RestoreObject](#)
- [UploadPart](#)

Note

Os pontos de acesso multirregionais não suportam a operação [CopyObject](#) da API. Em vez disso, você precisará executar as ações `CopyObject` diretamente entre buckets.

Gerenciamento de pontos de acesso de várias regiões

O Amazon S3 fornece um conjunto de operações para o gerenciamento de pontos de acesso de várias regiões. O Amazon S3 processa algumas dessas operações de forma síncrona e algumas de forma assíncrona. Quando você invoca uma operação assíncrona, o Amazon S3 primeiro autoriza de forma

síncrona a operação solicitada. Se a autorização for bem-sucedida, o Amazon S3 retornará um token que você pode usar para rastrear o progresso e os resultados da operação solicitada.

Note

Solicitações feitas por meio do AWS Management Console são sempre síncronas. O console aguarda até que a solicitação seja concluída antes de permitir que você envie outra solicitação.

Você pode exibir o status atual e os resultados das operações assíncronas usando o console ou usar a `DescribeMultiRegionAccessPointOperation` na AWS CLI, SDKs da AWS ou a API REST. O Amazon S3 fornece um token de rastreamento na resposta a uma operação assíncrona. Você inclui esse token de rastreamento como um argumento para a `DescribeMultiRegionAccessPointOperation`. Em seguida, o Amazon S3 retorna o status atual e os resultados da operação especificada, incluindo quaisquer erros ou informações relevantes de recursos. O Amazon S3 executa operações de `DescribeMultiRegionAccessPointOperation` de forma síncrona.

Todas as solicitações para criar ou manter pontos de acesso de várias regiões são encaminhadas para a região Oeste dos EUA (Oregon). Isso ocorre independentemente da região em que você esteja ao fazer a solicitação ou de quais regiões o ponto de acesso de várias regiões oferece suporte. Além disso, é preciso conceder a permissão de `s3>ListAllMyBuckets` para o usuário, função ou outra entidade do IAM que faz uma solicitação para gerenciar um ponto de acesso de várias regiões.

Monitoramento e registro de solicitações feitas por meio de um ponto de acesso multirregional para recursos subjacentes

O Amazon S3 registra solicitações feitas por meio de pontos de acesso multirregionais e solicitações feitas às APIs que os gerenciam, como `CreateMultiRegionAccessPoint` e `GetMultiRegionAccessPointPolicy`. As solicitações feitas ao Amazon S3 por meio de um ponto de acesso multirregional aparecem nos logs de acesso do servidor do Amazon S3 e em logs do AWS CloudTrail com o nome de host do ponto de acesso multirregional. O nome do host de um ponto de acesso assume a forma de <MRAP_alias>.accesspoint.s3-global.amazonaws.com. Por exemplo, suponha que você tenha a seguinte configuração de bucket e de ponto de acesso multirregional:

- Um bucket nomeado my-bucket-usw2 na região us-west-2 que contém o objeto my-image.jpg.
- Um bucket nomeado my-bucket-aps1 na região ap-south-1 que contém o objeto my-image.jpg.
- Um bucket nomeado my-bucket-euc1 na região eu-central-1 que não contém um objeto nomeado my-image.jpg.
- Um ponto de acesso multirregional chamado my-mrap com o alias mfzwi23gnjvgw.mrap que está configurado para atender a solicitações de todos os três buckets.
- O ID de sua conta da AWS é 123456789012.

Uma solicitação feita para recuperar my-image.jpg diretamente por meio de quaisquer buckets aparece nos logs com um nome de host de <bucket_name>.s3.<Region>.amazonaws.com.

Se você fizer a solicitação por meio do ponto de acesso multirregional, o Amazon S3 primeiro determinará quais dos buckets nas diferentes regiões atenderão à solicitação com a menor latência. Depois que o Amazon S3 determina qual bucket usar para atender à solicitação, ele envia a solicitação para esse bucket e registra a operação usando o nome de host do ponto de acesso multirregional. Neste exemplo, se o Amazon S3 retransmitiu a solicitação para my-bucket-aps1, seus logs refletirão uma solicitação GET bem-sucedida para my-image.jpg do my-bucket-aps1, usando um nome de host do mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com.

É importante estar ciente de que o Amazon S3 não realiza qualquer consideração sobre qual o bucket pode ser capaz de atender à solicitação. Se o Amazon S3 determinou que o my-bucket-euc1 teria a menor latência, seus logs refletirão uma falha de solicitação GET de my-image.jpg do my-bucket-euc1, usando um nome de host do mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com. Se a solicitação tiver sido roteada para my-bucket-usw2, em vez disso, seus logs indicarão uma solicitação GET bem-sucedida.

Para obter mais informações sobre logs de acesso do servidor do Amazon S3, consulte [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#). Para obter mais informações sobre o AWS CloudTrail, consulte [O que é o AWS CloudTrail?](#) no Guia do usuário do AWS CloudTrail.

Monitoramento e registro de solicitações feitas para APIs de gerenciamento de pontos de acesso multirregionais

O Amazon S3 fornece várias operações para gerenciar pontos de acesso multirregionais, como `CreateMultiRegionAccessPoint` e `GetMultiRegionAccessPointPolicy`. Quando você faz essas solicitações usando a AWS CLI, SDKs ou API REST, o Amazon S3 processa essas solicitações de forma assíncrona. Desde que você tenha as permissões apropriadas para a solicitação, o Amazon S3 retornará um token para essas solicitações. Você pode usar este token com `DescribeAsyncOperation` para ajudar você a visualizar o status das operações assíncronas em andamento. O S3 processa solicitações de `DescribeAsyncOperation` de forma síncrona. Você pode usar o AWS Management Console, a AWS CLI, SDKs ou API REST para visualizar o status das solicitações assíncronas.

Note

O console exibe somente o status de solicitações assíncronas feitas nos 14 dias anteriores. Para exibir o status das solicitações mais antigas, use a AWS CLI, SDKs ou a API REST.

As operações de gerenciamento assíncronas podem estar em um dos vários estados:

NEW

O Amazon S3 recebeu a solicitação e está se preparando para executar a operação.

IN_PROGRESS

O Amazon S3 está executando a operação no momento.

SUCCESS

A operação foi bem-sucedida. A resposta inclui informações relevantes, como o alias de ponto de acesso multirregional para uma solicitação `CreateMultiRegionAccessPoint`.

FAILED

Falha na operação. A resposta inclui uma mensagem de erro indicando o motivo da falha da solicitação.

Tópicos

- [AWS CloudTrail com pontos de acesso multirregionais \(p. 354\)](#)

AWS CloudTrail com pontos de acesso multirregionais

É possível usar o AWS CloudTrail para visualizar, pesquisar, fazer download, arquivar, analisar e responder às atividades da conta em toda a infraestrutura da AWS. Com pontos de acesso multirregionais e registro do CloudTrail, você pode identificar quem ou o que executou qual ação, quais recursos foram

acionados, quando o evento ocorreu e outros detalhes para ajudar a analisar e responder à atividade por meio de seu ponto de acesso multirregional.

Como configurar o AWS CloudTrail para pontos de acesso multirregionais

Para habilitar o registro do CloudTrail para quaisquer operações relacionadas à criação ou manutenção de pontos de acesso multirregionais, você deve configurar o registro do CloudTrail para registrar os eventos na região US West (Oregon). Isso ocorre independentemente da região em que você esteja ao fazer a solicitação ou de quais regiões o ponto de acesso multirregional oferece suporte. Todas as solicitações para criar ou manter pontos de acesso multirregionais são encaminhadas para a região US West (Oregon). Você deve adicionar essa região a uma trilha existente ou criar uma nova trilha contendo essa região e todas as regiões associadas ao ponto de acesso de várias regiões.

O Amazon S3 registra solicitações feitas por meio de pontos de acesso multirregionais e solicitações feitas às operações de APIs que os gerenciam, como `CreateMultiRegionAccessPoint` e `GetMultiRegionAccessPointPolicy`. Quando você registra essas solicitações por meio de um ponto de acesso multirregional, elas aparecem em seus logs do AWS CloudTrail com o nome do host do ponto de acesso multirregional. Por exemplo, se você fizer solicitações para um bucket por meio de um ponto de acesso multirregional com o alias `mfzwi23gnjvgw.mrap`, as entradas no log do CloudTrail teriam um nome de host de `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`.

Lembre-se de que os pontos de acesso multirregionais servem para rotear solicitações para o bucket que responde com a menor latência. Por isso, quando você estiver observando os logs do CloudTrail para um ponto de acesso multirregional, você verá solicitações sendo feitas dos buckets subjacentes. Algumas dessas solicitações podem ser solicitações diretas para o bucket e não roteadas pelo ponto de acesso multirregional. É importante ter isso em mente ao analisar o tráfego. Quando um bucket está em um ponto de acesso multirregional, as solicitações ainda podem ser feitas a esse bucket diretamente sem passar pelo ponto de acesso multirregional.

Há eventos assíncronos envolvidos na criação e gerenciamento de pontos de acesso multirregionais. As solicitações assíncronas não têm eventos de conclusão no log do CloudTrail. Para mais informações sobre solicitações assíncronas, consulte [Monitoramento e registro de solicitações feitas para APIs de gerenciamento de pontos de acesso multirregionais \(p. 354\)](#).

Para obter mais informações sobre o AWS CloudTrail, consulte [O que é o AWS CloudTrail?](#) no Guia do usuário do AWS CloudTrail.

Restrições e limitações de pontos de acesso de várias regiões

Os pontos de acesso de várias regiões do Amazon S3 têm as seguintes restrições e limitações:

- Nomes de pontos de acesso de várias regiões:
 - Devem ser exclusivos em uma única conta e região da AWS.
 - Devem começar com um número ou uma letra minúscula.
 - Devem conter entre 3 e 50 caracteres.
 - Não é possível começar ou terminar com um traço.
 - Não é possível conter sublinhados, letras maiúsculas ou pontos.
 - Não pode ser editado depois que eles são criados.
- Os aliases de pontos de acesso de várias regiões são gerados pelo Amazon S3 e não podem ser editados ou reutilizados.

- Não é possível acessar dados por meio de um ponto de acesso de várias regiões usando endpoints de gateway ou endpoints da interface. Para usar AWS PrivateLink, você deve criar endpoints de ponto de acesso de várias regiões. Para obter mais informações, consulte [Configurar um ponto de acesso multirregional para uso com AWS PrivateLink \(p. 344\)](#).
- Você não pode usar um ponto de acesso de várias regiões como a origem da distribuição do Amazon CloudFront.
- Requisitos mínimos para pontos de acesso de várias regiões:
 - Transport Layer Security (TLS) v1.2
 - Signature versão 4 (SigV4A)

Os pontos de acesso de várias regiões são compatíveis com o Signature versão 4A. Esta versão do SigV4 permite que as solicitações sejam assinadas para várias regiões da AWS. Isso é útil em operações de API que podem resultar em acesso a dados de uma das várias regiões. Ao usar o SDK da AWS, você fornece suas credenciais e as solicitações para pontos de acesso de várias regiões usarão o Signature versão 4A, sem configuração adicional. Para obter mais informações sobre SigV4A, consulte [Assinatura de solicitações de API da AWS](#) na Referência geral da AWS.

- Limitações do ponto de acesso de várias regiões:
 - O IPv6 não é compatível.
 - Não há suporte para buckets do Amazon S3 on Outposts.
 - No há suporte do `CopyObject`, seja como a origem ou o destino.
 - Não há suporte para operações em lote do S3.
- Limites de cota de serviço:
 - Há no máximo 100 pontos de acesso de várias regiões por conta.
 - Há um limite de 20 regiões para um único ponto de acesso de várias regiões.
- Somente as seguintes regiões da AWS são suportadas:
 - US East (N. Virginia)
 - US East (Ohio)
 - US West (N. California)
 - US West (Oregon)
 - Asia Pacific (Mumbai)
 - Asia Pacific (Osaka)
 - Asia Pacific (Seoul)
 - Asia Pacific (Singapore)
 - Asia Pacific (Sydney)
 - Asia Pacific (Tokyo)
 - Canada (Central)
 - Europe (Frankfurt)
 - Europe (Ireland)
 - Europe (London)
 - Europe (Paris)
 - Europe (Stockholm)
 - South America (São Paulo)

Segurança do Amazon S3

A segurança da nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você se beneficiará de um datacenter e de uma arquitetura de rede criados para atender aos requisitos das empresas com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isto como segurança da nuvem e segurança na nuvem:

Segurança da nuvem

A AWS é responsável pela proteção da infraestrutura que executa serviços da AWS na Nuvem AWS. A AWS também fornece a você serviços que podem ser usados com segurança. A eficácia da nossa segurança é regularmente testada e verificada por auditores de terceiros como parte dos [Programas de conformidade da AWS](#). Para saber mais sobre os programas de conformidade que se aplicam ao Amazon S3, consulte [Serviços da AWS no escopo pelo programa de conformidade](#).

Segurança na nuvem

Sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da sua organização e as leis e regulamentos aplicáveis. Para o Amazon S3, sua responsabilidade inclui as seguintes áreas:

- Gerenciar seus dados, incluindo [propriedade de objetos e criptografia](#).
- Classificar seus ativos.
- [Gerenciar o acesso a seus dados usando as funções do IAM e outras configurações de serviço](#) para aplicar as permissões apropriadas.
- Habilitar controles de detecção, como [AWS CloudTrail](#) ou [Amazon GuardDuty](#) para o Amazon S3.

Esta documentação ajudará você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Amazon S3. Os tópicos a seguir mostram como configurar o Amazon S3 para atender aos seus objetivos de segurança e de conformidade. Você também aprenderá como usar outros produtos da AWS que podem ajudar a monitorar e proteger seus recursos do Amazon S3.

Tópicos

- [Proteção de dados no Amazon S3 \(p. 358\)](#)
- [Proteção de dados usando criptografia \(p. 358\)](#)
- [Privacidade do tráfego entre redes \(p. 409\)](#)
- [AWS PrivateLink for Amazon S3 \(p. 410\)](#)
- [Identity and Access Management no Amazon S3 \(p. 419\)](#)
- [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#)
- [Registrar em log e monitorar no Amazon S3 \(p. 664\)](#)
- [Validação de conformidade para o Amazon S3 \(p. 666\)](#)
- [Resiliência no Amazon S3 \(p. 667\)](#)
- [Segurança da infraestrutura no Amazon S3 \(p. 670\)](#)
- [Análise de configuração e vulnerabilidade no Amazon S3 \(p. 671\)](#)
- [Melhores práticas de segurança para o Amazon S3 \(p. 672\)](#)

Proteção de dados no Amazon S3

O Amazon S3 fornece uma infraestrutura de armazenamento resiliente projetada para armazenamento de dados de missão crítica e primários. S3 Standard, S3 Intelligent-Tiering, S3 Standard – IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive armazenam objetos redundantemente em vários dispositivos em um mínimo de três zonas de disponibilidade de uma Região da AWS. Uma zona de disponibilidade é um ou mais datacenters discretos com energia, redes e conectividade redundantes em uma Região da AWS. As zonas de disponibilidade são fisicamente separadas por uma distância considerável, de muitos quilômetros, de qualquer outra zona de disponibilidade, embora todas estejam a até 100 quilômetros de distância uma da outra. A classe de armazenamento S3 One Zone – IA armazena dados de forma redundante em vários dispositivos dentro de uma única zona de disponibilidade. Esses serviços são projetados para lidar com falhas simultâneas de dispositivos, detectando e reparando rapidamente qualquer redundância perdida, além de verificar regularmente a integridade dos dados usando somas de verificação.

O armazenamento Amazon S3 Standard oferece os seguintes recursos:

- Respalhado pelo [Acordo de Nível de Serviço do Amazon S3](#).
- Projeto para fornecer 99,999999999% de durabilidade e 99,99% de disponibilidade dos objetos em determinado ano.
- S3 Standard, S3 Intelligent-Tiering, S3 Standard – IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive foram projetados para sustentar dados em caso de perda de uma zona de disponibilidade do Amazon S3 inteira.

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.

Para fins de proteção de dados, recomendamos que você proteja as credenciais da sua Conta da AWS e configure contas de usuário individuais com o AWS Identity and Access Management de modo que cada usuário receba somente as permissões necessárias para cumprir suas funções.

Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comando ou uma API, use um endpoint do FIPS. Para obter mais informações sobre endpoints do FIPS, consulte [Federal Information Processing Standard \(FIPS\) 140-2](#).

As melhores práticas de segurança a seguir também abordam a proteção de dados no Amazon S3:

- [Implement server-side encryption](#)
- [Enforce encryption of data in transit](#)
- [Consider using Macie with Amazon S3](#)
- [Identify and audit all your Amazon S3 buckets](#)
- [Monitor Amazon Web Services security advisories](#)

Proteção de dados usando criptografia

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). É possível proteger dados em trânsito usando SSL/TLS (Secure Socket Layer/Transport Layer Security) ou criptografia no lado do cliente. Você tem as seguintes opções de proteção de dados em repouso no Amazon S3:

- Criptografia no lado do servidor: peça que o Amazon S3 criptografe o objeto antes de salvá-lo em discos em seus datacenters e descriptografe-o ao fazer download dos objetos.

Para configurar a criptografia do lado do servidor, consulte [Especificando criptografia no lado do servidor com o AWS KMS \(SSE-KMS\) \(p. 364\)](#) ou [Especificação de criptografia do Amazon S3 \(p. 380\)](#).

- Criptografia no lado do cliente: criptografe dados no lado do cliente e faça upload dos dados criptografados no Amazon S3. Nesse caso, você gerencia o processo de criptografia, as chaves de criptografia e as ferramentas relacionadas.

Para configurar a criptografia do lado do cliente, consulte [Proteger dados usando a criptografia no lado do cliente \(p. 405\)](#).

Para obter mais informações sobre criptografia do lado do servidor e criptografia do lado do cliente, consulte os tópicos listados abaixo.

Tópicos

- [Proteção de dados usando criptografia no lado do servidor \(p. 359\)](#)
- [Proteger dados usando a criptografia no lado do cliente \(p. 405\)](#)

Proteção de dados usando criptografia no lado do servidor

A criptografia do lado do servidor é a criptografia de dados em seu destino pela aplicação ou serviço que os recebe. O Amazon S3 criptografa os dados no nível do objeto no momento em que os grava em discos nos datacenters e descriptografa-os 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. Além disso, quando você lista objetos no bucket, a API de lista retorna uma lista de todos os objetos, independentemente de estarem ou 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 escolhe gerenciar as chaves de criptografia.

Criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3)

Quando usar criptografia no lado do servidor com as chaves gerenciadas pelo Amazon S3 (SSE-S3), cada objeto será criptografado com uma chave exclusiva. Como uma proteção adicional, ela criptografa a própria chave utilizando uma chave-raiz que alterna regularmente. A criptografia no lado do servidor do Amazon S3 usa uma das criptografias de bloco mais fortes disponíveis, o GCM padrão de criptografia avançada de 256 bits (AES-256), para criptografar seus dados. Sobre objetos criptografados antes do AES-GCM, o AES-CBC ainda é compatível com a descriptografia desses objetos. Para obter mais informações, consulte [Proteção de dados usando criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 \(SSE-S3\) \(p. 378\)](#).

Criptografia no lado do servidor com chaves do KMS armazenadas no AWS Key Management Service (SSE-KMS)

A criptografia no lado do servidor com AWS KMS keys (SSE-KMS) é semelhante a SSE-S3, mas com alguns benefícios adicionais e cobranças para usar esse serviço. Há outras permissões para uso de uma chave do KMS que fornece maior proteção contra acesso não autorizado de seus objetos no Amazon

S3. O SSE-KMS também fornece uma trilha de auditoria que mostra quando a chave do KMS foi usada e por quem. Além disso, é possível criar e gerenciar chaves gerenciadas pelo cliente ou usar chaves gerenciadas pela AWS que são exclusivas para você, para o seu serviço e para a sua região. Para obter mais informações, consulte [Proteção de dados usando criptografia do servidor com AWS Key Management Service \(SSE-KMS\) \(p. 360\)](#).

Criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C)

Com a criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C), você gerencia as chaves de criptografia e o Amazon S3 gerencia a criptografia, conforme grava em discos; e a descriptografia, ao acessar os objetos. Para obter mais informações, consulte [Proteger dados usando a criptografia de servidor com chaves de criptografia fornecidas pelo cliente \(SSE-C\) \(p. 390\)](#).

Proteção de dados usando criptografia do servidor com AWS Key Management Service (SSE-KMS)

A criptografia no lado do servidor é a criptografia dos dados em seu destino pelo aplicativo ou serviço que os recebe. 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 Amazon S3 usa AWS KMS keys para criptografar seus objetos do Amazon S3. O AWS KMS criptografa apenas os dados do objeto. A soma de verificação e o algoritmo especificado são armazenados como parte dos metadados do objeto. Se a criptografia no lado do servidor for solicitada para o objeto, a soma de verificação será armazenada em formato criptografado.

Se você usar chaves do KMS, poderá usar o AWS KMS por meio do [AWS Management Console](#) ou de [APIs do AWS KMS](#) para fazer o seguinte:

- Criar chaves do KMS centralmente
- Definir políticas que controlam como as chaves do KMS podem ser usadas
- Auditá-lo para provar que elas estão sendo usadas corretamente

Os controles de segurança do AWS KMS podem ajudá-lo a satisfazer os requisitos de conformidade relacionados à criptografia. Você pode usar essas chaves do KMS para proteger seus dados em buckets do Amazon S3. Ao usar a criptografia SSE-KMS com um bucket do S3, as AWS KMS keys devem estar na mesma região que o bucket.

Não há custos adicionais por usar o AWS KMS keys. Para obter mais informações, consulte [Conceitos de AWS KMS key](#) no Guia do desenvolvedor do AWS Key Management Service e [Preços do AWS KMS](#).

Permissões

Para carregar um objeto criptografado com uma AWS KMS key para o Amazon S3, você precisa de permissões `kms:GenerateDataKey` na chave. Para baixar um objeto criptografado com uma AWS KMS key, você precisa de permissões `kms:Decrypt`. Para obter informações sobre permissões do AWS KMS necessárias para o carregamento fracionado, consulte [API de multipart upload e permissões \(p. 183\)](#).

Tópicos

- [AWS KMS keys \(p. 361\)](#)
- [Amazon S3 Bucket Keys \(p. 361\)](#)
- [Exigir a criptografia no lado do servidor \(p. 362\)](#)
- [Contexto de criptografia \(p. 362\)](#)
- [AWS Signature versão 4 \(p. 363\)](#)
- [Especificando criptografia no lado do servidor com o AWS KMS \(SSE-KMS\) \(p. 364\)](#)
- [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3 \(p. 369\)](#)

AWS KMS keys

Ao usar criptografia no servidor com o AWS KMS (SSE-KMS), você pode usar a [chave gerenciada pela AWS](#) padrão ou especificar uma [chave gerenciada pelo cliente](#) que você já tenha criado. O AWS KMS usa criptografia de envelope para proteger ainda mais os dados. Criptografia de envelope é a prática de criptografar dados de texto simples com uma chave de dados e criptografar essa chave de dados com uma chave-raiz.

Se você não especificar uma chave gerenciada pelo cliente, o Amazon S3 criará automaticamente uma AWS KMS key em sua Conta da AWS na primeira vez que você adicionar um objeto criptografado com SSE-KMS a um bucket. Por padrão, o Amazon S3 usa essa chave do KMS para SSE-KMS.

Se você quiser usar uma chave gerenciada pelo cliente para SSE-KMS, crie a chave gerenciada pelo cliente antes de configurar o SSE-KMS. Depois, ao configurar o SSE-KMS para seu bucket, especifique a chave gerenciada pelo cliente existente.

A criação de uma chave gerenciada pelo cliente oferece a você mais flexibilidade e controle. Por exemplo, você pode criar, alternar e desabilitar chaves gerenciadas pelo cliente. Você também pode definir controles de acesso e auditar a chave gerenciada pelo cliente que você usa para proteger seus dados. Para obter mais informações sobre chaves gerenciadas pelo cliente e gerenciadas pela AWS, consulte [Conceitos do AWS KMS](#) no Guia do desenvolvedor do AWS Key Management Service.

Se você optar por criptografar seus dados usando uma chave do KMS ou uma chave gerenciada pelo cliente, o AWS KMS e o Amazon S3 realizarão estas ações:

- O Amazon S3 solicitará uma [chave de dados](#) em texto simples e uma cópia da chave criptografada na chave do KMS especificada.
- O AWS KMS gera uma chave de dados, criptografa-a com a chave do KMS e envia a chave de dados em texto simples e a chave de dados criptografada ao Amazon S3.
- O Amazon S3 criptografa os dados usando a chave de dados e remove a chave de texto não criptografado da memória assim que possível, após o uso.
- O Amazon S3 armazena a chave de dados criptografada como metadados com os dados criptografados.

Quando você solicitar que os dados sejam descriptografados, o Amazon S3 e o AWS KMS farão o seguinte:

- O Amazon S3 envia a chave de dados criptografada ao AWS KMS.
- O AWS KMS descriptografa a chave usando a mesma chave do KMS e retorna a chave de dados de texto simples para o Amazon S3.
- O Amazon S3 descriptografa o texto cifrado e remove a chave de dados de texto não criptografado da memória assim que possível.

Important

Ao usar uma AWS KMS key para criptografia no lado do servidor no Amazon S3, você deve escolher uma chave de criptografia do KMS simétrica. O Amazon S3 só é compatível com chaves do KMS de criptografia simétrica, e não com chaves assimétricas. Para obter mais informações, consulte [Uso de chaves simétricas e assimétricas](#) no Guia do desenvolvedor do AWS Key Management Service.

Amazon S3 Bucket Keys

Ao configurar a criptografia do lado do servidor usando o AWS KMS (SSE-KMS), você pode configurar seu bucket para usar chaves de bucket do S3 para SSE-KMS. Usar uma chave em nível de bucket para SSE-KMS pode reduzir os custos de solicitação do AWS KMS em até 99% ao reduzir o tráfego de solicitação do Amazon S3 para o AWS KMS.

Quando você configura seu bucket para usar chaves de bucket do S3 para SSE-KMS em novos objetos, o AWS KMS gera uma chave no nível de bucket usada para criar [chaves de dados](#) exclusivas para objetos no bucket. Essa chave de bucket é usada por um período limitado no Amazon S3, reduzindo ainda mais a necessidade do Amazon S3 fazer solicitações ao AWS KMS para concluir operações de criptografia. Para obter mais informações sobre como usar Chaves de bucket do S3, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3 \(p. 369\)](#).

Exigir a criptografia no lado do servidor

Para exigir criptografia no lado do servidor de todos os objetos em um bucket específico do Amazon S3, é possível usar uma política. 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": "PutObjectPolicy",  
    "Statement": [  
        {  
            "Sid": "DenyUnEncryptedObjectUploads",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*",  
            "Condition": {  
                "StringNotEquals": {  
                    "s3:x-amz-server-side-encryption": "aws:kms"  
                }  
            }  
        }  
    ]  
}
```

Para exigir que uma AWS KMS key específica seja usada para criptografar os objetos em um bucket, use a chave de condição `s3:x-amz-server-side-encryption-aws-kms-key-id`. Para especificar a chave do KMS, é necessário usar um nome de recurso da Amazon (ARN) da chave no 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 assumirá que você quer usar a chave gerenciada pela AWS. No entanto, o ID de chave do AWS KMS que o Amazon S3 usa para a criptografia de objeto deve corresponder ao ID de chave do AWS KMS na política. Caso contrário, o Amazon S3 negará a solicitação.

Para obter uma lista completa de chaves de condição específicas do Amazon S3, consulte [Chaves de condição do Amazon S3](#).

Contexto de criptografia

Um contexto de criptografia é um conjunto de pares chave-valor que contém informações contextuais adicionais sobre os dados. O contexto de criptografia não é criptografado. Quando um contexto de criptografia é especificado para uma operação criptográfica, o Amazon S3 deve especificar o mesmo contexto de criptografia da operação de descriptografia. Caso contrário, a descriptografia falhará. O AWS KMS usa o contexto de criptografia como [dados autenticados adicionais](#) (AAD) para oferecer suporte à [criptografia autenticada](#). Para obter mais informações sobre o contexto de criptografia, consulte [Contexto de criptografia](#) no Guia do desenvolvedor do AWS Key Management Service.

O Amazon S3 usa automaticamente o nome do recurso da Amazon (ARN) do objeto ou bucket como o par de contexto de criptografia:

- Se você usar o SSE-KMS sem habilitar uma chave de bucket do S3, o ARN do objeto será usado como o contexto de criptografia.

```
arn:aws:s3:::object_ARN
```

- Se você usar o SSE-KMS e habilitar uma chave de bucket do S3, o ARN do bucket será usado como o contexto de criptografia. Para obter mais informações sobre chaves de buckets do S3, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3 \(p. 369\)](#).

```
arn:aws:s3:::bucket_ARN
```

Opcionalmente, você pode fornecer um par de contexto de criptografia adicional usando o cabeçalho `x-amz-server-side-encryption-context` em uma solicitação [s3:PutObject](#). No entanto, como o contexto de criptografia não é criptografado, ele não deve incluir informações sigilosas. O Amazon S3 armazena esse par de chaves adicional junto com o contexto de criptografia padrão. Ao processar sua solicitação PUT, o Amazon S3 acrescenta o contexto de criptografia padrão de `aws:s3:arn` àquele fornecido por você.

É possível usar o contexto de criptografia para identificar e categorizar suas operações de criptografia. Você também pode usar o valor do ARN do contexto de criptografia padrão para rastrear solicitações relevantes no AWS CloudTrail visualizando qual ARN do Amazon S3 foi usado com qual chave de criptografia.

No campo `requestParameters` de um arquivo de log do CloudTrail, o contexto de criptografia é parecido com este.

```
"encryptionContext": {  
    "aws:s3:arn": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/file_name"  
}
```

Quando você usa o SSE-KMS com o recurso opcional S3 Bucket Keys, o valor do contexto de criptografia é o ARN do bucket.

```
"encryptionContext": {  
    "aws:s3:arn": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"  
}
```

AWS Signature versão 4

O Signature versão 4 é o processo para adicionar informações de autenticação às solicitações da AWS enviadas por HTTP. Por segurança, a maioria das solicitações para AWS deve ser assinada com uma chave de acesso, que consiste em um ID de chave de acesso e na chave de acesso secreta. Essas duas chaves são comumente conhecidas como suas credenciais de segurança. Para obter mais informações, consulte [Autenticação de solicitações \(AWS Signature Version 4\)](#) e [Processo de assinatura do Signature Version 4](#).

Important

- Todas as solicitações GET e PUT para objetos criptografados do AWS KMS devem ser criados usando o Secure Sockets Layer (SSL) ou Transport Layer Security (TLS). As solicitações também devem ser assinadas usando credenciais válidas, como AWS Signature Version 4 (ou AWS Signature Version 2).
- Se o objeto usar SSE-KMS, os cabeçalhos de solicitação de criptografia não deverão ser enviados para solicitações GET e solicitações HEAD, ou um erro HTTP 400 BadRequest será exibido.

Tópicos

- [Especificando criptografia no lado do servidor com o AWS KMS \(SSE-KMS\) \(p. 364\)](#)
- [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3 \(p. 369\)](#)

Especificando criptografia no lado do servidor com o AWS KMS (SSE-KMS)

Ao criar um objeto, você pode especificar o uso da criptografia no lado do servidor com as chaves do AWS Key Management Service (AWS KMS) para criptografar seus dados. Essa criptografia é conhecida como SSE-KMS. É possível aplicar criptografia durante o upload de um novo objeto ou ao copiar um objeto existente.

Você pode especificar o SSE-KMS usando o console do Amazon S3, operações da API REST, AWSSDKs e a AWS Command Line Interface (AWS CLI). Para obter mais informações, consulte os tópicos a seguir.

Note

Você pode usar uma AWS KMS keys de várias regiões no Amazon S3. No entanto, o Amazon S3 trata no momento as chaves de várias regiões como se fossem chaves de região única e não usa os recursos de várias regiões da chave. Para obter mais informações, consulte [Usar chaves de várias regiões](#) no Guia do desenvolvedor do AWS Key Management Service.

Uso do console do S3

Este tópico descreve como definir ou alterar o tipo de criptografia de um objeto usando o console do Amazon S3.

Note

Se você alterar a criptografia de um objeto, um novo objeto será criado para substituir o antigo. Se o versionamento do S3 estiver habilitado, uma nova versão do objeto será criada e o objeto existente se tornará uma versão mais antiga. A função que altera a propriedade também se torna o proprietário do novo objeto ou (versão do objeto).

Como adicionar ou alterar a criptografia de um objeto

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket que contém o objeto.
3. Na lista Name (Nome), escolha o nome do objeto ao qual você deseja adicionar ou no qual deseja alterar a criptografia.

A Object overview (Visão geral do objeto) será exibida, mostrando as propriedades do objeto.

4. Em Server-side encryption settings (Configurações de criptografia do lado do servidor), escolha Edit (Editar).

A página da Edit server-side encryption (criptografia do lado do servidor da edição) é aberta

5. Para habilitar a criptografia do lado do servidor para seu objeto, em Server-side encryption (Criptografia do lado do servidor), escolha Enable (Ativar).
6. Em Encryption key type (Tipo de chave de criptografia), escolha AWS Key Management Service chave (SSE-KMS).

Important

Se você usar a opção do AWS 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](#).

7. Em AWS KMS key (Chave do AWS KMS), selecione uma das seguintes opções:
 - Chave gerenciada da AWS (aws/s3)
 - Choose from your AWS KMS keys (Escolher de suas chaves do KMS), e escolha sua chave do KMS.
 - Enter KMS master key ARN (Inserir ARN da chave mestra do KMS) e insira o nome do recurso da Amazon (ARN) da sua chave do AWS KMS.

Important

Você só pode usar AWS KMS keys habilitadas na mesma Região da AWS do bucket. Quando você seleciona Choose from your AWS KMS keys (Escolher entre suas AWS KMS keys), o console do S3 lista somente 100 chaves do KMS por região. Se você tiver mais de 100 chaves do KMS na mesma região, será possível ver somente as primeiras 100 chaves do KMS no console do S3. Para usar uma chave do KMS que não esteja listada no console, escolha Custom KMS ARN (Personalizar o ARN do KMS) e insira o ARN da chave do KMS. Quando você usa uma AWS KMS key para criptografia no lado do servidor no Amazon S3, você deve escolher uma chave do KMS habilitada na mesma região do bucket. Além disso, o Amazon S3 só oferece suporte a chaves do KMS simétricas, e não a chaves do KMS assimétricas. Para obter mais informações, consulte [Uso de chaves simétricas e assimétricas](#) no Guia do desenvolvedor do AWS Key Management Service.

Para obter mais informações sobre como criar uma AWS KMS key, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service. Para obter mais informações sobre como usar o AWS KMS com o Amazon S3, consulte [Proteção de dados usando criptografia do servidor com AWS Key Management Service \(SSE-KMS\) \(p. 360\)](#).

8. Selecione Save changes.

Note

Essa ação aplica criptografia a todos os objetos especificados. Ao criptografar pastas, aguarde a conclusão da operação de salvamento antes de adicionar novos objetos à pasta.

Uso dos REST API

Quando você cria um objeto, ou seja, quando você carrega um novo objeto ou copia um objeto existente, você pode especificar a utilização de criptografia do lado do servidor com AWS KMS keys para criptografar seus dados. Para fazer isso, adicione 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 que o objeto foi armazenado usando SSE-KMS retornando o cabeçalho de resposta `x-amz-server-side-encryption`.

Se você especificar o cabeçalho `x-amz-server-side-encryption` com um valor de `aws:kms`, também poderá usar os seguintes cabeçalhos de solicitação:

- `x-amz-server-side-encryption-aws-kms-key-id`
- `x-amz-server-side-encryption-context`
- `x-amz-server-side-encryption-bucket-key-enabled`

Tópicos

- [Operações da API REST do Amazon S3 compatíveis com SSE-KMS \(p. 366\)](#)
- [Contexto de criptografia \(`x-amz-server-side-encryption-context`\) \(p. 366\)](#)
- [ID da chave do AWS KMS \(`x-amz-server-side-encryption-aws-kms-key-id`\) \(p. 367\)](#)
- [Chaves de bucket do S3 \(`x-amz-server-side-encryption-aws-bucket-key-enabled`\) \(p. 367\)](#)

Operações da API REST do Amazon S3 compatíveis com SSE-KMS

As seguintes operações da API REST aceitam os cabeçalhos de solicitação `x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-id`, e `x-amz-server-side-encryption-context`.

- [PUT object](#): ao carregar dados usando a operação API PUT, você pode especificar esses cabeçalhos de solicitação.
- [COPY object](#): quando você copia um objeto, há um objeto de origem e um objeto de destino. Ao transmitir cabeçalhos de SSE-KMS com a operação de COPY (Copiar), eles são aplicados somente ao objeto de destino. 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.
- [Objeto POST](#): ao usar uma operação POST para fazer upload de um objeto, em vez dos cabeçalhos de solicitação, você fornece as mesmas informações nos campos de formulário.
- [Create Multipart Upload](#): ao carregar objetos grandes usando a API de carregamento fracionado, você pode especificar esses cabeçalhos. Esses cabeçalhos são especificados na solicitação de início.

Os cabeçalhos de resposta das APIs REST a seguir retornam o cabeçalho `x-amz-server-side-encryption` quando um objeto é armazenado usando criptografia de servidor.

- [PUT object](#)
- [COPY object](#)
- [Objeto POST](#)
- [Create Multipart Upload](#)
- [Carregar parte](#)
- [Upload Part Copy](#)
- [Concluir carregamento fracionado](#)
- [Get Object](#)
- [Head Object](#)

Important

- Todas as solicitações GET e PUT para um objeto protegido por AWS KMS falharão se elas não forem feitas Secure Sockets Layer (SSL), Transport Layer Security (TLS) ou Signature Version 4.
- Se o objeto usar SSE-KMS, os cabeçalhos de solicitação de criptografia não deverão ser enviados para solicitações GET e solicitações HEAD, ou um erro HTTP 400 BadRequest será exibido.

Contexto de criptografia (`x-amz-server-side-encryption-context`)

Se você especificar `x-amz-server-side-encryption:aws:kms`, a API do Amazon S3 oferece suporte a um contexto de criptografia com o cabeçalho `x-amz-server-side-encryption-context`. Um contexto de criptografia é um conjunto de pares chave-valor que contém informações contextuais adicionais sobre os dados.

O Amazon S3 usa automaticamente o nome do recurso da Amazon (ARN) do objeto ou bucket como o par de contexto de criptografia. Se você usar o SSE-KMS sem habilitar uma chave de bucket do S3, você usará o ARN de objeto como seu contexto de criptografia, por exemplo, `arn:aws:s3:::object_ARN`. No entanto, se você usar o SSE-KMS e habilitar uma chave de bucket do S3, use o ARN do bucket para o contexto de criptografia, por exemplo, `arn:aws:s3:::bucket_ARN`.

Você também pode fornecer um par adicional de contexto de criptografia usando o cabeçalho `x-amz-server-side-encryption-context`. No entanto, como o contexto de criptografia não é criptografado, ele não deve incluir informações sigilosas. O Amazon S3 armazena esse par de chaves adicional junto com o contexto de criptografia padrão.

Para obter informações sobre o contexto de criptografia no Amazon S3, consulte [Contexto de criptografia \(p. 362\)](#). Para obter informações gerais sobre o contexto de criptografia, consulte [Conceitos do AWS Key Management Service: contexto de criptografia](#) no Guia do desenvolvedor do AWS Key Management Service.

[ID da chave do AWS KMS \(x-amz-server-side-encryption-aws-kms-key-id\)](#)

Você pode usar o cabeçalho `x-amz-server-side-encryption-aws-kms-key-id` para especificar o ID da chave gerenciada pelo cliente usada para proteger os dados. Se você especificar o `x-amz-server-side-encryption:aws:kms`, mas não fornecer `x-amz-server-side-encryption-aws-kms-key-id`, o Amazon S3 usa a chave gerenciada pela AWS para proteger os dados. Se desejar usar uma chave gerenciada pelo cliente, você deve fornecer o cabeçalho `x-amz-server-side-encryption-aws-kms-key-id` da chave gerenciada pelo cliente.

Important

Ao usar uma AWS KMS key para criptografia no lado do servidor no Amazon S3, você deve escolher uma chave de criptografia do KMS simétrica. O Amazon S3 só é compatível com chaves do KMS de criptografia simétrica, e não com chaves assimétricas. Para obter mais informações, consulte [Uso de chaves simétricas e assimétricas](#) no Guia do desenvolvedor do AWS Key Management Service.

[Chaves de bucket do S3 \(x-amz-server-side-encryption-aws-bucket-key-enabled\)](#)

Você pode usar o cabeçalho da `x-amz-server-side-encryption-aws-bucket-key-enabled` solicitação para ativar ou desativar uma chave de bucket do S3 no nível do objeto. As chaves de bucket do S3 podem reduzir os custos de solicitação do AWS KMS diminuindo o tráfego de solicitação do Amazon S3 para o AWS KMS. Para obter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3 \(p. 369\)](#).

Se você especificar `x-amz-server-side-encryption:aws:kms`, mas não fornecer `x-amz-server-side-encryption-aws-bucket-key-enabled`, seu objeto usará as configurações da chave de bucket do S3 para o bucket de destino para criptografar seu objeto. Para obter mais informações, consulte [Configuração de uma chave de bucket do S3 no nível de objeto usando operações em lote, a API REST, os AWS SDKs ou a AWS CLI \(p. 375\)](#).

[Uso de AWS SDKs](#)

Ao usar AWS SDKs, você pode solicitar ao Amazon S3 que use as AWS KMS keys. 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](#).

Important

Ao usar uma AWS KMS key para criptografia no lado do servidor no Amazon S3, você deve escolher uma chave de criptografia do KMS simétrica. O Amazon S3 só é compatível com chaves do KMS de criptografia simétrica, e não com chaves assimétricas. Para obter mais informações, consulte [Uso de chaves simétricas e assimétricas](#) no Guia do desenvolvedor do AWS Key Management Service.

[Operação de cópia](#)

Ao copiar objetos, você adiciona as mesmas propriedades de solicitação (`ServerSideEncryptionMethod` e `ServerSideEncryptionKeyManagementServiceKeyId`) para solicitar ao Amazon S3 que utilize uma AWS KMS key. Para obter mais informações sobre cópia de objetos, consulte [Cópia de objetos \(p. 215\)](#).

Operação PUT

Java

Ao carregar um objeto usando o AWS SDK for Java, você pode solicitar que o Amazon S3 use uma AWS KMS key adicionando a propriedade `SSEAwsKeyManagementParams`, conforme exibido na solicitação a seguir.

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,  
    keyName, file).withSSEAwsKeyManagementParams(new SSEAwsKeyManagementParams());
```

Nesse caso, o Amazon S3 usa a chave gerenciada pela AWS (consulte [Proteção de dados usando criptografia do servidor com AWS Key Management Service \(SSE-KMS\) \(p. 360\)](#)). Como opção, é possível criar uma chave de criptografia simétrica do KMS e especificar isso na solicitação.

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,  
    keyName, file).withSSEAwsKeyManagementParams(new SSEAwsKeyManagementParams(keyID));
```

Para obter mais informações sobre como criar chaves gerenciadas pelo cliente, consulte [Programação da API do AWS KMS](#) no Guia do desenvolvedor do AWS Key Management Service.

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 [Fazer upload de objetos \(p. 171\)](#).
- Para um multipart upload, consulte os seguintes tópicos:
 - Para usar a API de carregamento fracionado de alto nível, consulte [Fazer upload de um objeto usando multipart upload \(p. 187\)](#).
 - Para usar a API de carregamento fracionado de baixo nível, consulte [Usar os AWS SDKs \(API de baixo nível\) \(p. 194\)](#).

.NET

Ao carregar um objeto usando o AWS SDK for .NET, você pode solicitar que o Amazon S3 use uma AWS KMS key adicionando a propriedade `ServerSideEncryptionMethod`, conforme exibido na solicitação a seguir.

```
PutObjectRequest putRequest = new PutObjectRequest  
{  
    BucketName = bucketName,  
    Key = keyName,  
    // other properties.  
    ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS  
};
```

Nesse caso, o Amazon S3 usa a chave gerenciada pela AWS. Para obter mais informações, consulte [Proteção de dados usando criptografia do servidor com AWS Key Management Service \(SSE-KMS\) \(p. 360\)](#). É possível criar sua própria chave gerenciada pelo cliente de criptografia simétrica e especificar isso na solicitação.

```
PutObjectRequest putRequest1 = new PutObjectRequest  
{  
    BucketName = bucketName,  
    Key = keyName,  
    // other properties.  
    ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS,
```

```
    ServerSideEncryptionKeyManagementServiceKeyId = keyId
};
```

Para obter mais informações sobre como criar chaves gerenciadas pelo cliente, consulte [Programação da API do AWS KMS](#) no Guia do desenvolvedor do AWS Key Management Service.

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 [Fazer upload de objetos \(p. 171\)](#).
- Para carregamento fracionado, consulte os seguintes tópicos:
 - Para usar a API de carregamento fracionado de alto nível, consulte [Fazer upload de um objeto usando multipart upload \(p. 187\)](#).
 - Para usar a API de carregamento fracionado de baixo nível, consulte [Fazer upload de um objeto usando multipart upload \(p. 187\)](#).

Pre-signed URLs

Java

Ao criar um URL pré-assinado para um objeto criptografado com uma AWS KMS key, você deverá especificar explicitamente o Signature Version 4.

```
ClientConfiguration clientConfiguration = new ClientConfiguration();
clientConfiguration.setSignerOverride("AWSS3V4SignerType");
AmazonS3Client s3client = new AmazonS3Client(
    new ProfileCredentialsProvider(), clientConfiguration);
...
```

Para ver um exemplo de código, consulte [Compartilhar objetos usando URLs pré-assinados \(p. 275\)](#).

.NET

Ao criar um URL pré-assinado para um objeto criptografado com uma AWS KMS key, você deverá especificar explicitamente o Signature Version 4.

```
AWSConfigs.S3Config.UseSignatureVersion4 = true;
```

Para ver um exemplo de código, consulte [Compartilhar objetos usando URLs pré-assinados \(p. 275\)](#).

Redução do custo do SSE-KMS com chaves de bucket do Amazon S3

As chaves de bucket do Amazon S3 reduzem o custo da criptografia no lado do servidor do Amazon S3 usando o AWS Key Management Service (SSE-KMS). Essa nova chave de nível de bucket para SSE pode reduzir os custos de solicitação do AWS KMS em até 99%, diminuindo o tráfego de solicitações do Amazon S3 para o AWS KMS. Com alguns cliques no AWS Management Console e sem alterações em aplicações do cliente, você pode configurar seus buckets para usar uma Chave de bucket do S3 para criptografia baseada em AWS KMS em novos objetos.

Chaves de bucket S3 para SSE-KMS

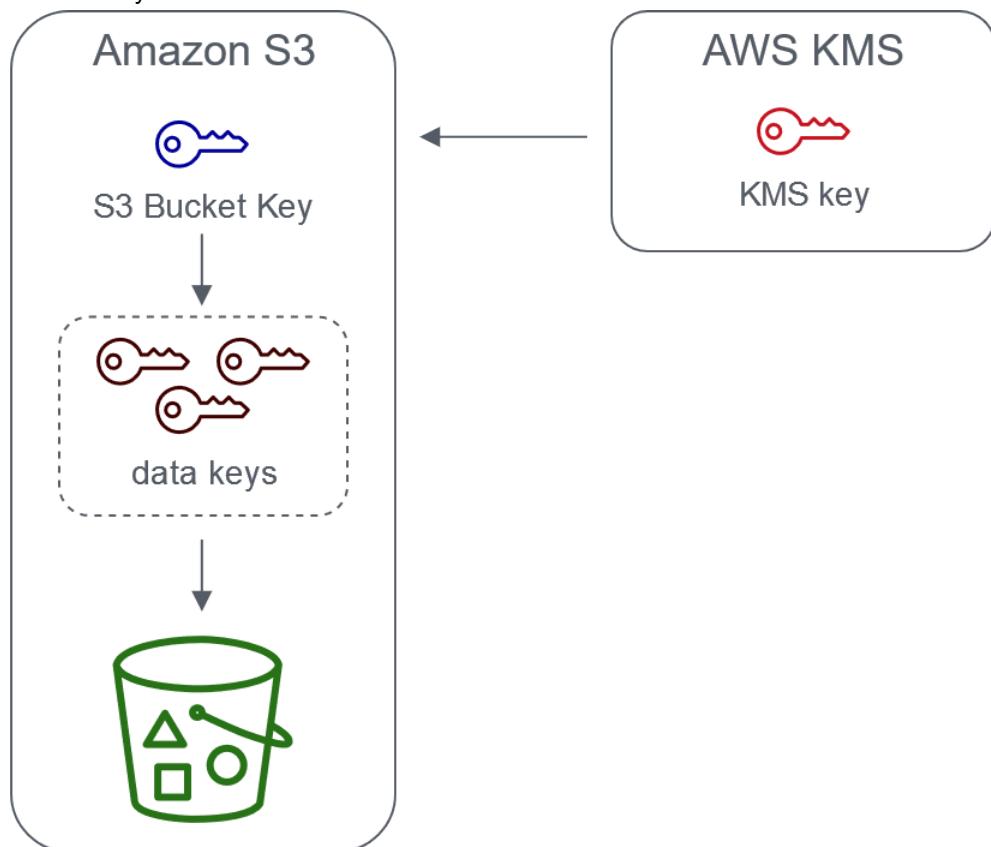
Cargas de trabalho que acessam milhões ou bilhões de objetos criptografados com o SSE-KMS podem gerar grandes volumes de solicitações para o AWS KMS. Quando você usa o SSE-KMS para proteger seus dados sem uma chave de bucket do S3, o Amazon S3 usa uma [chave de dados](#) individual do AWS KMS para cada objeto. Ele faz uma chamada para o AWS KMS sempre que uma solicitação é

feita contra um objeto criptografado no KMS. Para obter informações sobre como o SSE-KMS funciona, consulte [Proteção de dados usando criptografia do servidor com AWS Key Management Service \(SSE-KMS\) \(p. 360\)](#).

Quando você configura seu bucket para usar uma chave de bucket do S3 para SSE-KMS, o AWS KMS gera uma chave em nível de bucket que é usada para criar chaves de dados exclusivas para novos objetos que você adiciona ao bucket. Essa chave de bucket do S3 é usada por um período limitado no Amazon S3, reduzindo a necessidade do Amazon S3 fazer solicitações ao AWS KMS para concluir operações de criptografia. Isso reduz o tráfego do S3 para o AWS KMS, permitindo que você acesse objetos criptografados no AWS KMS no S3 por uma fração do custo anterior.

Quando você configura uma chave de bucket do S3, os objetos que já estão no bucket não usam a chave do bucket do S3. Para configurar uma chave de bucket do S3 para objetos existentes, você pode usar uma operação COPY. Para obter mais informações, consulte [Configuração de uma chave de bucket do S3 no nível de objeto usando operações em lote, a API REST, os AWS SDKs ou a AWS CLI \(p. 375\)](#).

O Amazon S3 compartilhará apenas uma chave de bucket do S3 para objetos criptografados pela mesma AWS KMS key.



Server-side encryption with AWS Key Management service using an S3 Bucket Key

Configurando chaves de bucket do S3

Você pode configurar seu bucket para usar uma chave de bucket do S3 para SSE-KMS em novos objetos por meio do console do Amazon S3, dos AWS SDKs, da AWS CLI ou da API REST. Você também pode substituir a configuração da chave de bucket do S3 para objetos específicos em um bucket por uma chave KMS individual por objeto usando a API REST, o AWS SDK ou a AWS CLI. Você também pode visualizar as configurações da Chave do bucket do S3.

Antes de configurar seu bucket para usar uma chave de bucket do S3, revise [Alterações na observação antes de habilitar uma chave de bucket do S3 \(p. 371\)](#).

Configuração de uma chave de bucket do S3 usando o console do Amazon S3

Ao criar um novo bucket, você pode configurar seu bucket para usar uma Chave de bucket do S3 para SSE-KMS em novos objetos. Você também pode configurar um bucket existente para usar uma Chave de bucket do S3 para SSE-KMS em novos objetos atualizando suas propriedades do bucket.

Para obter mais informações, consulte [Configurando seu bucket para usar uma chave de bucket do S3 com SSE-KMS para novos objetos \(p. 372\)](#).

Compatibilidade da API REST, da AWS CLI e do AWS SDK com chaves de bucket do S3

Você pode usar a API REST, a AWS CLI ou o AWS SDK para configurar seu bucket para usar uma chave de bucket do S3 para SSE-KMS em novos objetos. Você também pode habilitar uma chave de bucket do S3 no nível do objeto.

Para obter mais informações, consulte:

- [Configuração de uma chave de bucket do S3 no nível de objeto usando operações em lote, a API REST, os AWS SDKs ou a AWS CLI \(p. 375\)](#)
- [Configurando seu bucket para usar uma chave de bucket do S3 com SSE-KMS para novos objetos \(p. 372\)](#)

As seguintes APIs suportam chaves de bucket do S3 para SSE-KMS:

- [PutBucketEncryption](#)
 - `ServerSideEncryptionRule` aceita o parâmetro `BucketKeyEnabled` para habilitar e desabilitar uma chave de bucket do S3.
- [GetBucketEncryption](#)
 - `ServerSideEncryptionRule` retorna as configurações para `BucketKeyEnabled`.
- [PutObject](#), [CopyObject](#), [CreateMultipartUpload](#) e [PostObject](#)
 - O cabeçalho de solicitação `x-amz-server-side-encryption-bucket-key-enabled` ativa ou desativa uma Chave de bucket do S3 no nível do objeto.
- [HeadObject](#), [GetObject](#), [UploadPartCopy](#), [UploadPart](#) e [CompleteMultipartUpload](#)
 - O cabeçalho de resposta `x-amz-server-side-encryption-bucket-key-enabled` indica se uma Chave de bucket S3 está ativada ou desativada para um objeto.

Trabalho com AWS CloudFormation

No AWS CloudFormation, o recurso `AWS::S3::Bucket` inclui uma propriedade de criptografia chamada `BucketKeyEnabled` que você pode usar para ativar ou desativar uma chave de bucket do S3.

Para obter mais informações, consulte [Usar o AWS CloudFormation \(p. 375\)](#).

Alterações na observação antes de habilitar uma chave de bucket do S3

Antes de ativar uma chave de bucket do S3, observe as seguintes alterações relacionadas:

Políticas de chave do IAM ou KMS

Se suas políticas de IAM ou políticas de chave do AWS KMS usarem seu objeto do nome do recurso da Amazon (ARN) como contexto de criptografia para refinar ou limitar o acesso à chave do KMS, essas políticas não funcionarão com uma chave de bucket do S3. As Chaves de bucket do S3 usam o ARN do bucket como contexto de criptografia. Antes de ativar uma chave de bucket do S3, atualize suas políticas do IAM ou políticas de chave do AWS KMS para usar o ARN do bucket como contexto de criptografia.

Para obter mais informações sobre contexto de criptografia e chaves de bucket do S3, consulte [Contexto de criptografia \(p. 362\)](#).

Eventos do AWS KMS CloudTrail

Depois de habilitar uma chave de bucket do S3, seus eventos do AWS KMS CloudTrail registram em log o ARN do bucket em vez do ARN do objeto. Além disso, você vê menos eventos do KMS CloudTrail para objetos SSE-KMS em seus logs. Como o material de chave tem uma limitação de tempo no Amazon S3, menos solicitações são feitas ao AWS KMS.

Uso de uma chave de bucket do S3 com replicação

Você pode usar chaves de bucket do S3 com SRR (Same-Region Replication, replicação entre regiões) e CRR (Cross-Region Replication, replicação entre regiões).

Quando o Amazon S3 replica um objeto criptografado, ele geralmente preserva as configurações de criptografia do objeto de réplica no bucket de destino. No entanto, se o objeto de origem não for criptografado e seu bucket de destino usar criptografia padrão ou uma chave de bucket do S3, o Amazon S3 criptografa o objeto com a configuração do bucket de destino.

Os exemplos a seguir ilustram como uma chave de bucket do S3 funciona com a replicação. Para obter mais informações, consulte [Replicação de objetos criados com criptografia no lado do servidor \(SSE\) usando chaves do KMS \(p. 867\)](#).

Example Exemplo 1 — O objeto de origem usa chaves de bucket do S3, o bucket de destino usa criptografia padrão

Se o objeto de origem usa uma Chave de bucket do S3, mas seu bucket de destino usa criptografia padrão com o SSE-KMS, o objeto de réplica manterá suas configurações de criptografia S3 Bucket Key no bucket de destino. O bucket de destino ainda usa criptografia padrão com SSE-KMS.

Example Exemplo 2 — O objeto de origem não é criptografado, o bucket de destino usa uma chave de bucket do S3 com SSE-KMS

Se o objeto de origem não estiver criptografado e o bucket de destino usar uma chave de bucket S3 com SSE-KMS, o objeto de origem será criptografado com uma chave de bucket S3 usando SSE-KMS no 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.

Como trabalhar com chaves de bucket do S3

Para obter mais informações sobre como ativar e trabalhar com chaves de bucket do S3, consulte as seguintes seções:

- [Configurando seu bucket para usar uma chave de bucket do S3 com SSE-KMS para novos objetos \(p. 372\)](#)
- [Configuração de uma chave de bucket do S3 no nível de objeto usando operações em lote, a API REST, os AWS SDKs ou a AWS CLI \(p. 375\)](#)
- [Exibição de configurações para uma chave de bucket do S3 \(p. 377\)](#)

Configurando seu bucket para usar uma chave de bucket do S3 com SSE-KMS para novos objetos

Ao configurar a criptografia do lado do servidor usando o SSE-KMS, você pode configurar seu bucket para usar uma chave de bucket do S3 para SSE-KMS em novos objetos. As chaves de bucket do S3 diminuem o tráfego de solicitações do Amazon S3 para o AWS Key Management Service (AWS KMS) e reduzem o custo do SSE-KMS. Para obter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3 \(p. 369\)](#).

Você pode configurar seu bucket para usar uma chave de bucket do S3 para SSE-KMS em novos objetos usando o console do Amazon S3, a API REST, o AWS SDK, a AWS CLI ou o AWS CloudFormation. Se você quiser habilitar ou desabilitar uma chave de bucket do S3 para objetos existentes, poderá usar uma operação COPY. Para obter mais informações, consulte [Configuração de uma chave de bucket do S3 no nível de objeto usando operações em lote, a API REST, os AWS SDKs ou a AWS CLI \(p. 375\)](#) e [Uso do S3 Batch Operations para criptografar objetos com chaves de bucket do S3 \(p. 959\)](#).

Quando uma chave de bucket do S3 estiver habilitada para o bucket de origem ou de destino, o contexto de criptografia será o Amazon Resource Name (ARN) do bucket e não o ARN do objeto, por exemplo, `arn:aws:s3:::bucket_ARN`. Você precisa atualizar suas políticas do IAM para usar o ARN de bucket para o contexto de criptografia. Para obter mais informações, consulte [Conceder permissões adicionais para a função do IAM \(p. 869\)](#).

Os exemplos a seguir ilustram como uma chave de bucket do S3 funciona com a replicação. Para obter mais informações, consulte [Replicação de objetos criados com criptografia no lado do servidor \(SSE\) usando chaves do KMS \(p. 867\)](#).

Pré-requisito:

Antes de configurar seu bucket para usar uma chave de bucket do S3, revise [Alterações na observação antes de habilitar uma chave de bucket do S3 \(p. 371\)](#).

Uso do console do S3

No console do S3, você pode ativar ou desativar uma chave de bucket do S3 para um bucket novo ou existente. Os objetos no console do S3 herdam sua configuração de chave de bucket do S3 da configuração do bucket. Quando você habilita uma chave de bucket do S3 para seu bucket, novos objetos que você envia para o bucket usam uma chave de bucket do S3 para criptografia no lado do servidor usando o AWS KMS.

Carregando, copiando ou modificando objetos em buckets que tenham uma chave de bucket S3 ativada

Se você carregar, modificar ou copiar um objeto em um bucket que tenha uma chave de bucket S3 ativada, as configurações da chave de bucket do S3 para esse objeto poderão ser atualizadas para alinhar com a configuração do bucket.

Se um objeto já tiver uma chave de bucket S3 ativada, as configurações da chave de bucket do S3 para esse objeto não serão alteradas quando você copia ou modifica o objeto. No entanto, se você modificar ou copiar um objeto que não tenha uma chave de bucket S3 ativada e o bucket de destino tiver uma configuração de chave de bucket S3, o objeto herdará as configurações da chave de bucket S3 do bucket de destino. Por exemplo, se o objeto de origem não tiver uma chave de bucket S3 ativada, mas o bucket de destino tiver a chave de bucket S3 ativada, uma chave de bucket S3 será habilitada para o objeto.

Para habilitar uma chave de bucket do S3 ao criar um novo bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Selecione Create bucket (Criar bucket).
3. Insira o nome do bucket e escolha sua Região da AWS.
4. Em Default encryption (Criptografia padrão), escolha Edit (Editar).
5. Em Encryption type (Tipo de criptografia), selecione AWS Key Management Service key (SSE-KMS) [Chave do AWS Key Management Service (SSE-KMS)].
6. Escolha uma chave do AWS KMS:
 - Escolha AWS managed key (aws/s3) (Chave gerenciada pela AWS (aws/s3)).
 - Escolha Customer managed key (Chave gerenciada pelo cliente) e uma chave gerenciada pelo cliente de criptografia simétrica na mesma região de seu bucket.

7. Em Bucket key (Chave do bucket), escolha Enable (Ativar).
8. Selecione Create bucket (Criar bucket).

O Amazon S3 cria seu bucket com uma chave de bucket do S3 ativada. Se você fizer upload de novos objetos no bucket, eles usarão uma chave de bucket do S3. Para desabilitar uma chave de bucket do S3, siga as etapas anteriores e escolha disable (desabilitar).

Para habilitar uma chave de bucket do S3 para um bucket existente

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets (Buckets), escolha o bucket para o qual você deseja habilitar uma chave de bucket do S3.
3. Escolha Properties (Propriedades).
4. Em Default encryption (Criptografia padrão), escolha Edit (Editar).
5. Em Default encryption (Criptografia padrão), escolha Edit (Editar).
6. Em Encryption type (Tipo de criptografia), selecione AWS Key Management Service key (SSE-KMS) [Chave do AWS Key Management Service (SSE-KMS)].
7. Escolha uma chave do AWS KMS:
 - Escolha AWS managed key (aws/s3) (Chave gerenciada pela AWS (aws/s3)).
 - Escolha Customer managed key (Chave gerenciada pelo cliente) e uma chave gerenciada pelo cliente de criptografia simétrica na mesma região de seu bucket.
8. Em Bucket key (Chave do bucket), escolha Enable (Ativar).
9. Selecione Save changes.

O Amazon S3 habilita uma chave de bucket do S3 para novos objetos adicionados ao seu bucket. Os objetos existentes não usam a chave de bucket do S3. Para desativar uma chave de bucket do S3, siga as etapas anteriores e escolha Disable (Desativar).

Uso dos REST API

Você pode usar [PutBucketEncryption](#) para habilitar ou desabilitar uma chave de bucket do S3 para seu bucket. Para configurar uma chave de bucket do S3 com [PutBucketEncryption](#), especifique o [ServerSideEncryptionRule](#), que inclui criptografia padrão com criptografia no lado do servidor usando AWS KMS key. Você também pode usar opcionalmente uma chave gerenciada pelo cliente especificando o ID da chave do KMS para a chave gerenciada pelo cliente.

Para obter mais informações e sintaxe de exemplo, consulte [PutBucketEncryption](#).

Usar o AWS SDK para Java

O exemplo a seguir habilita a criptografia de bucket padrão com SSE-KMS e uma chave de bucket do S3 usando o AWS SDK for Java.

Java

```
AmazonS3 s3client = AmazonS3ClientBuilder.standard()
    .withRegion(Regions.DEFAULT_REGION)
    .build();

ServerSideEncryptionByDefault serverSideEncryptionByDefault = new
    ServerSideEncryptionByDefault()
        .withSSEAlgorithm(SSEAlgorithm.KMS);
ServerSideEncryptionRule rule = new ServerSideEncryptionRule()
    .withApplyServerSideEncryptionByDefault(serverSideEncryptionByDefault)
```

```
.withBucketKeyEnabled(true);
ServerSideEncryptionConfiguration serverSideEncryptionConfiguration =
    new ServerSideEncryptionConfiguration().withRules(Collections.singleton(rule));

SetBucketEncryptionRequest setBucketEncryptionRequest = new
    SetBucketEncryptionRequest()
        .withServerSideEncryptionConfiguration(serverSideEncryptionConfiguration)
        .withBucketName(bucketName);

s3client.setBucketEncryption(setBucketEncryptionRequest);
```

Uso do AWS CLI

O exemplo a seguir habilita a criptografia de bucket padrão com SSE-KMS e uma chave de bucket do S3 usando o AWS CLI.

```
aws s3api put-bucket-encryption --bucket <bucket-name> --server-side-encryption-
configuration '{
    "Rules": [
        {
            "ApplyServerSideEncryptionByDefault": {
                "SSEAlgorithm": "aws:kms",
                "KMSMasterKeyID": "<KMS-Key-ARN>"
            },
            "BucketKeyEnabled": true
        }
    ]
}'
```

Usar o AWS CloudFormation

Para obter mais informações sobre como configurar uma chave de bucket do S3 usando AWS CloudFormation, consulte [AWS::S3::Bucket ServerSideEncryptionRule](#) no Guia do usuário do AWS CloudFormation.

Configuração de uma chave de bucket do S3 no nível de objeto usando operações em lote, a API REST, os AWS SDKs ou a AWS CLI

Ao executar uma operação PUT ou COPY usando a API REST, os AWS SDKs ou a AWS CLI, você pode habilitar ou desabilitar uma chave de bucket do S3 no nível de objeto. As chaves de bucket do S3 reduzem o custo da criptografia do lado do servidor usando o AWS Key Management Service (AWS KMS) (SSE-KMS), diminuindo assim o tráfego de solicitações do Amazon S3 para o AWS KMS. Para obter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3 \(p. 369\)](#).

Quando você configura uma chave de bucket do S3 para um objeto usando uma operação PUT ou COPY, o Amazon S3 atualiza somente as configurações desse objeto. As configurações da chave de bucket S3 para o bucket de destino não são alteradas. Se você não especificar uma chave de bucket do S3 para seu objeto, o Amazon S3 aplicará as configurações da chave de bucket do S3 para o bucket de destino ao objeto.

Pré-requisito:

Antes de configurar seu objeto para usar uma chave de bucket do S3, revise [Alterações na observação antes de habilitar uma chave de bucket do S3 \(p. 371\)](#).

Tópicos

- [Amazon S3 Batch Operations \(p. 376\)](#)
- [Uso dos REST API \(p. 376\)](#)

- [Usar o AWSSDK Java \(PutObject\) \(p. 376\)](#)
- [Como usar a AWS CLI \(PutObject\) \(p. 377\)](#)

Amazon S3 Batch Operations

Para criptografar objetos existentes do Amazon S3 com uma única solicitação, você pode usar o Amazon S3 Batch Operations. Você fornece uma lista de objetos às operações em lote do S3 que, por sua vez, chamam a respectiva API para realizar a operação especificada. É possível usar a [operação Copy do S3 Batch Operations](#) para copiar objetos não criptografados existentes e gravá-los de volta no mesmo bucket que os objetos criptografados. Um único trabalho do Batch Operations pode realizar a operação especificada em bilhões de objetos. Para obter mais informações, consulte [Executar operações em lote de grande escala em objetos do Amazon S3 \(p. 938\)](#) e [Criptografia de objetos existentes com o Amazon S3 Batch Operations](#).

Uso dos REST API

Ao usar o SSE-KMS, você pode habilitar uma chave de bucket do S3 para um objeto usando as seguintes APIs:

- [PutObject](#) — Ao fazer upload de um objeto, você pode especificar o cabeçalho da `x-amz-server-side-encryption-bucket-key-enabled` solicitação para ativar ou desativar uma chave de bucket do S3 no nível do objeto.
- [CopyObject](#) (Copiar objeto) — Quando você copia um objeto e configura o SSE-KMS, você pode especificar o cabeçalho da solicitação `x-amz-server-side-encryption-bucket-key-enabled` para ativar ou desativar uma chave de bucket do S3 para seu objeto.
- [PostObject](#) — Quando você usa uma operação POST para fazer upload de um objeto e configurar o SSE-KMS, você pode usar o campo do formulário `x-amz-server-side-encryption-bucket-key-enabled` para ativar ou desativar uma chave de bucket S3 para seu objeto.
- [CreateMultipartUpload](#): ao carregar objetos grandes usando a API de carregamento fracionado e configurar o SSE-KMS, você pode usar o cabeçalho de solicitação `x-amz-server-side-encryption-bucket-key-enabled` para habilitar ou desabilitar uma chave de bucket do S3 para seu objeto.

Para habilitar uma chave de bucket do S3 no nível do objeto, inclua o cabeçalho da solicitação `x-amz-server-side-encryption-bucket-key-enabled`. Para obter mais informações sobre o SSE-KMS e a API REST, consulte [Uso dos REST API \(p. 365\)](#).

Usar o AWSSDK Java (PutObject)

Você pode usar o exemplo a seguir para configurar uma chave de bucket do S3 no nível de objeto usando o AWS SDK for Java.

Java

```
AmazonS3 s3client = AmazonS3ClientBuilder.standard()
    .withRegion(Regions.DEFAULT_REGION)
    .build();

String bucketName = "bucket name";
String keyName = "key name for object";
String contents = "file contents";

PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, keyName, contents)
    .withBucketKeyEnabled(true);

s3client.putObject(putObjectRequest);
```

Como usar a AWS CLI (PutObject)

Você pode usar o seguinte exemplo da AWS CLI para configurar uma chave de bucket do S3 no nível de objeto como parte de uma solicitação PutObject.

```
aws s3api put-object --bucket <bucket name> --key <object key name> --server-side-  
encryption aws:kms --bucket-key-enabled --body <filepath>
```

Exibição de configurações para uma chave de bucket do S3

Você pode visualizar as configurações de uma chave de bucket do S3 no nível de bucket ou objeto usando o console do Amazon S3, a API REST, a AWS CLI ou os AWS SDKs.

As chaves de bucket do S3 diminuem o tráfego de solicitações do Amazon S3 para o AWS KMS e reduzem o custo da criptografia do lado do servidor usando o AWS Key Management Service (SSE-KMS). Para obter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3 \(p. 369\)](#).

Para exibir as configurações da chave de bucket do S3 para um bucket ou um objeto que herdou as configurações da chave de bucket S3 da configuração do bucket, você precisa de permissão para executar a ação `s3:GetEncryptionConfiguration`. Para obter mais informações, consulte [GetBucketEncryption](#) na Referência da API do Amazon Simple Storage Service.

Uso do console do S3

No console do S3, você pode visualizar as configurações da chave de bucket do S3 para seu bucket ou objeto. As configurações da chave de bucket do S3 são herdadas da configuração do bucket, a menos que os objetos de origem já tenham uma chave de bucket S3 configurada.

Objetos e pastas no mesmo bucket podem ter configurações diferentes da chave de bucket do S3. Por exemplo, se você fizer upload de um objeto usando a API REST e habilitar uma chave de bucket S3 para o objeto, o objeto manterá sua configuração da chave de bucket de S3 no bucket de destino, mesmo que a chave de bucket S3 esteja desativada no bucket de destino. Como outro exemplo, se você habilitar uma chave de bucket S3 para um bucket existente, os objetos que já estão no bucket não usarão uma chave de bucket do S3. No entanto, novos objetos têm uma chave de bucket S3 ativada.

Para exibir o nível de bucket uma configuração de chave de bucket do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets (Buckets), escolha o bucket para o qual você deseja habilitar uma chave de bucket do S3.
3. Escolha Properties (Propriedades).
4. Na seção Default encryption (Criptografia padrão), em Bucket Key (Chave do bucket), você verá a configuração da chave de bucket do S3 para seu bucket.

Se você não conseguir ver a configuração da chave de bucket do S3, talvez não tenha permissão para executar a ação `s3:GetEncryptionConfiguration`. Para obter mais informações, consulte [GetBucketEncryption](#) na Referência da API do Amazon Simple Storage Service.

Para exibir a configuração da chave de bucket do S3 para seu objeto

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets (Buckets), escolha o bucket para o qual você deseja habilitar uma chave de bucket do S3.
3. Na lista Objects (Objetos), escolha o nome do objeto.

4. Na guia Details (Detalhes), em Server-side encryption settings (Configurações de criptografia do lado do servidor), escolha Edit (Editar).

Em Bucket Key, (Chave de bucket) você vê a configuração da chave de bucket do S3 para seu objeto, mas não pode editá-la.

Uso dos REST API

Para retornar as configurações da chave de bucket do S3 no nível de bucket

Para retornar informações de criptografia de um bucket, incluindo configurações para uma chave de bucket do S3, use a operação [GetBucketEncryption](#). As configurações da chave de bucket do S3 são retornadas no corpo da resposta no `ServerSideEncryptionConfiguration` com a configuração `BucketKeyEnabled`. Para obter mais informações, consulte [GetBucketEncryption](#) na Referência da API do Amazon S3.

Para retornar configurações de nível de objeto para uma chave de bucket do S3

Para retornar o status da chave de bucket S3 para um objeto, use a operação [HeadObject](#). O `HeadObject` retorna o cabeçalho de resposta de `x-amz-server-side-encryption-bucket-key-enabled` para mostrar se uma chave de bucket de S3 está ativada ou desativada para o objeto. Para obter mais informações, consulte [HeadObject](#) (Objeto do cabeçalho) na Referência da API do Amazon S3.

As seguintes operações de API também retornam o cabeçalho de resposta `x-amz-server-side-encryption-bucket-key-enabled` se uma chave de bucket de S3 estiver configurada para um objeto:

- [PutObject](#)
- [PostObject](#)
- [CopyObject](#)
- [CreateMultipartUpload](#)
- [UploadPartCopy](#)
- [UploadPart](#)
- [CompleteMultipartUpload](#)
- [GetObject](#)

Proteção de dados usando criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 (SSE-S3)

A criptografia no lado do servidor protege dados em repouso. O Amazon S3 criptografa cada objeto com uma chave exclusiva. Como uma proteção adicional, ela criptografa a si mesma utilizando uma chave que alterna regularmente. A criptografia no lado do servidor do Amazon S3 usa uma das cifras de bloco mais fortes disponíveis para criptografar seus dados, o padrão de criptografia avançada de 256 bits (AES-256).

Não existem encargos adicionais pelo uso da criptografia no lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3). Mas, as solicitações de configuração do recurso de criptografia padrão geram os encargos de solicitação padrão do Amazon S3. Para obter mais informações sobre definição de preços, consulte [Definição de preços do Amazon S3](#).

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": "PutObjectPolicy",  
    "Statement": [  
        {
```

```
        "Sid": "DenyIncorrectEncryptionHeader",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:PutObject",
        "Resource": "arn:aws:s3:::awsexamplebucket1/*",
        "Condition": {
            "StringNotEquals": {
                "s3:x-amz-server-side-encryption": "AES256"
            }
        }
    },
{
    "Sid": "DenyUnencryptedObjectUploads",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::awsexamplebucket1/*",
    "Condition": {
        "Null": {
            "s3:x-amz-server-side-encryption": "true"
        }
    }
}
]
```

Note

- 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 [Uso dos REST API \(p. 380\)](#).

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 — Quando você copia um objeto, 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 AWS SDKs também fornecem APIs de wrapper que você pode usar para solicitar criptografia no lado do servidor. Você também pode usar o AWS Management Console para fazer upload de objetos e solicitar a criptografia no lado do servidor.

Tópicos

- [Especificação de criptografia do Amazon S3 \(p. 380\)](#)

Especificação de criptografia do Amazon S3

Ao criar um objeto, você pode especificar o uso da criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 para criptografar seus dados. Isso é verdade quando você está fazendo upload de um novo objeto ou copiando um objeto existente. Essa criptografia é conhecida como SSE-S3.

Você pode especificar o SSE-S3 usando o console do S3, APIs REST, AWS SDKs e a AWS CLI. Para obter mais informações, consulte os tópicos abaixo.

Para obter um exemplo de como copiar um objeto sem criptografia, consulte [Cópia de objetos \(p. 215\)](#).

Uso do console do S3

Este tópico descreve como definir ou alterar o tipo de criptografia de um objeto usando o console do AWS Management Console. Quando você copia um objeto usando o console, ele copia o objeto no estado em que ele se encontra. Ou seja, se a origem é criptografada, o objeto de destino também é criptografado. O console também permite que você adicione ou altere a criptografia de um objeto.

Note

Se você alterar a criptografia de um objeto, um novo objeto será criado para substituir o antigo. Se o versionamento do S3 estiver habilitado, uma nova versão do objeto será criada e o objeto existente se tornará uma versão mais antiga. A função que altera a propriedade também se torna o proprietário do novo objeto ou (versão do objeto).

Como adicionar ou alterar a criptografia de um objeto

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket que contém o objeto.
3. Na lista Name (Nome), escolha o nome do objeto ao qual você deseja adicionar ou no qual deseja alterar a criptografia.

A Object overview (Visão geral do objeto) será exibida, mostrando as propriedades do objeto.

4. Em Server-side encryption settings (Configurações de criptografia do lado do servidor), escolha Edit (Editar).

A página Edit server-side encryption (Editar criptografia do lado do servidor) é aberta.

5. Para habilitar a criptografia do lado do servidor para seu objeto, em Server-side encryption (Criptografia do lado do servidor), escolha Enable (Ativar).
6. Para habilitar a criptografia no lado do servidor usando uma chave gerenciada pelo Amazon S3, em Encryption key type (Tipo de chave de criptografia), escolha Amazon S3 key (SSE-S3) (Chave do Amazon S3 (SSE-S3)).

Para obter mais informações sobre como usar a criptografia no lado do servidor do Amazon S3 para criptografar seus dados, consulte [Proteção de dados usando criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 \(SSE-S3\) \(p. 378\)](#).

7. Selecione Save changes.

Note

Essa ação aplica criptografia a todos os objetos especificados. Ao criptografar pastas, aguarde a conclusão da operação de salvamento antes de adicionar novos objetos à pasta.

Uso dos REST API

No momento da criação do objeto (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 compatível com o Amazon S3. O Amazon S3 confirma que o objeto foi armazenado usando a criptografia no lado do servidor retornando o cabeçalho de 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](#)
- [Carregar parte](#)
- [Carregamento de parte: copiar](#)
- [Concluir carregamento fracionado](#)
- [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.

Uso de AWS SDKs

Ao usar AWS SDKs, você pode solicitar ao Amazon S3 que use chaves de criptografia gerenciadas pelo Amazon S3. Esta seção fornece exemplos de uso dos AWS SDKs em várias linguagens. Para obter informações sobre outros SDKs, consulte [Código de exemplo e bibliotecas](#).

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 [Fazer upload de um objeto usando multipart upload \(p. 187\)](#).
 - 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 [Usar os AWS SDKs \(API de baixo nível\) \(p. 194\)](#).

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, faça uma cópia do objeto, especificando o estado desejado de criptografia para a cópia e, em seguida, exclua o objeto original. O Amazon S3 criptografa o objeto copiado somente se você solicitar explicitamente criptografia no lado do servidor. 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 [Uso dos REST API](#) (p. 380). Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3](#) (p. 1267).

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

import java.io.ByteArrayInputStream;

public class SpecifyServerSideEncryption {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyNameToEncrypt = "*** Key name for an object to upload and encrypt ***";
        String keyNameToCopyAndEncrypt = "*** Key name for an unencrypted object to be encrypted by copying ***";
        String copiedObjectKeyName = "*** Key name for the encrypted copy of the unencrypted object ***";

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

            // Create a new object and specify the server-side encryption
            // algorithm to use for the object
            PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, "mykey", "myfile");
            putObjectRequest.setSSEAlgorithm("AES256");

            // Upload the object
            s3Client.putObject(putObjectRequest);
        } catch (AmazonServiceException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

```
.build();

// Upload an object and encrypt it with SSE.
uploadObjectWithSSEEncryption(s3Client, bucketName, keyNameToEncrypt);

// Upload a new unencrypted object, then change its encryption state
// to encrypted by making a copy.
changeSSEEncryptionStatusByCopying(s3Client,
        bucketName,
        keyNameToCopyAndEncrypt,
        copiedObjectKeyName);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}

private static void uploadObjectWithSSEEncryption(AmazonS3 s3Client, String
bucketName, String keyName) {
    String objectContent = "Test object encrypted with SSE";
    byte[] objectBytes = objectContent.getBytes();

    // Specify server-side encryption.
    ObjectMetadata objectMetadata = new ObjectMetadata();
    objectMetadata.setContentLength(objectBytes.length);
    objectMetadata.setSSEAlgorithm(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);
    PutObjectRequest putRequest = new PutObjectRequest(bucketName,
            keyName,
            new ByteArrayInputStream(objectBytes),
            objectMetadata);

    // Upload the object and check its encryption status.
    PutObjectResult putResult = s3Client.putObject(putRequest);
    System.out.println("Object \"" + keyName + "\" uploaded with SSE.");
    printEncryptionStatus(putResult);
}

private static void changeSSEEncryptionStatusByCopying(AmazonS3 s3Client,
                                                    String bucketName,
                                                    String sourceKey,
                                                    String destKey) {
    // Upload a new, unencrypted object.
    PutObjectResult putResult = s3Client.putObject(bucketName, sourceKey, "Object
example to encrypt by copying");
    System.out.println("Unencrypted object \"" + sourceKey + "\" uploaded.");
    printEncryptionStatus(putResult);

    // Make a copy of the object and use server-side encryption when storing the
    // copy.
    CopyObjectRequest request = new CopyObjectRequest(bucketName,
            sourceKey,
            bucketName,
            destKey);
    ObjectMetadata objectMetadata = new ObjectMetadata();
    objectMetadata.setSSEAlgorithm(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);
    request.setNewObjectMetadata(objectMetadata);

    // Perform the copy operation and display the copy's encryption status.
    CopyObjectResult response = s3Client.copyObject(request);
    System.out.println("Object \"" + destKey + "\" uploaded with SSE.");
    printEncryptionStatus(response);
```

```
// Delete the original, unencrypted object, leaving only the encrypted copy in
Amazon S3.
    s3Client.deleteObject(bucketName, sourceKey);
    System.out.println("Unencrypted object \"\" + sourceKey + "\" deleted.");
}

private static void printEncryptionStatus(SSEResultBase response) {
    String encryptionStatus = response.getSSEAlgorithm();
    if (encryptionStatus == null) {
        encryptionStatus = "Not encrypted with SSE";
    }
    System.out.println("Object encryption status is: " + encryptionStatus);
}
}
```

.NET

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 [Uso de AWS SDKs \(p. 218\)](#).

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 do Amazon S3 .NET \(p. 1268\)](#).

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

namespace Amazon.DocSamples.S3
{
    class SpecifyServerSideEncryptionTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** key name for object created ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            WritingAnObjectAsync().Wait();
        }

        static async Task WritingAnObjectAsync()
        {
            try
            {
                var putRequest = new PutObjectRequest
                {
                    BucketName = bucketName,
```

```
        Key = keyName,
        ContentBody = "sample text",
        ServerSideEncryptionMethod = ServerSideEncryptionMethod.AES256
    };

    var putResponse = await client.PutObjectAsync(putRequest);

    // Determine the encryption state of an object.
    GetObjectMetadataRequest metadataRequest = new GetObjectMetadataRequest
    {
        BucketName = bucketName,
        Key = keyName
    };
    GetObjectMetadataResponse response = await
    client.GetObjectMetadataAsync(metadataRequest);
    ServerSideEncryptionMethod objectEncryption =
    response.ServerSideEncryptionMethod;

    Console.WriteLine("Encryption method used: {0}",
    objectEncryption.ToString());
}
catch (AmazonS3Exception e)
{
    Console.WriteLine("Error encountered ***. Message:{0}' when writing an
object", e.Message);
}
catch (Exception e)
{
    Console.WriteLine("Unknown encountered on server. Message:{0}' when
writing an object", e.Message);
}
}
}
}
```

PHP

Este tópico mostra como usar classes da versão 3 do AWS SDK for PHP para adicionar criptografia no lado do servidor a objetos que você carrega para o Amazon Simple Storage Service (Amazon S3). Pressupõe-se que você já esteja seguindo as instruções para [Usar o AWS SDK for PHP e executar exemplos do PHP \(p. 1268\)](#) e tenha o AWS SDK for 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 [Uso dos REST API \(p. 380\)](#).

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

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

// $filepath should be an absolute path to a file on disk.
$filepath = '*** Your File Path ***';

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

// Upload a file with server-side encryption.
```

```
$result = $s3->putObject([
    'Bucket'           => $bucket,
    'Key'              => $keyname,
    'SourceFile'       => $filepath,
    'ServerSideEncryption' => 'AES256',
]);
```

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 os AWS SDKs \(API de baixo nível\) \(p. 194\)](#).
- 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 [Fazer upload de um objeto usando multipart upload \(p. 187\)](#).

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.

```
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'];
```

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. Por padrão, o `copyObject()` não criptografa 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 no lado do servidor ao objeto copiado.

```
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',
]);

```

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

Ruby

Ao usar o AWS SDK for 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 for Ruby – Versão 3 a seguir demonstra como especificar que um arquivo carregado no Amazon S3 seja criptografado em repouso.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectPutSseWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object_encrypted(object_content, encryption)
    @object.put(body: object_content, server_side_encryption: encryption)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put your content to #{object.key}. Here's why: #{e.message}"
    false
  end
end

def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-encrypted-content"
  object_content = "This is my super-secret content."
  encryption = "AES256"

  wrapper = ObjectPutSseWrapper.new(Aws::S3::Object.new(bucket_name, object_content))
  return unless wrapper.put_object_encrypted(object_content, encryption)

  puts "Put your content into #{bucket_name}:#{object_key} and encrypted it with
  #{encryption}."
end
```

```
run_demo if $PROGRAM_NAME == __FILE__
```

Para obter um exemplo que mostra como fazer upload de um objeto sem SSE, consulte [Fazer upload de objetos \(p. 171\)](#).

O exemplo de código a seguir demonstra como determinar o estado de criptografia de um objeto existente.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectGetEncryptionWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Gets the object into memory.
  #
  # @return [Aws::S3::Types::GetObjectOutput, nil] The retrieved object data if
  # successful; otherwise nil.
  def get_object
    @object.get
    rescue Aws::Errors::ServiceError => e
      puts "Couldn't get object #{@object.key}. Here's why: #{e.message}"
    end
  end

  # Replace bucket name and object key with an existing bucket and object that you own.
  def run_demo
    bucket_name = "doc-example-bucket"
    object_key = "my-object.txt"

    wrapper = ObjectGetEncryptionWrapper.new(Aws::S3::Object.new(bucket_name,
      object_key))
    obj_data = wrapper.get_object
    return unless obj_data

    encryption = obj_data.server_side_encryption.nil? ? "no" :
    obj_data.server_side_encryption
    puts "Object #{object_key} uses #{encryption} encryption."
  end

run_demo if $PROGRAM_NAME == __FILE__
```

Se a criptografia no lado do servidor não for usada para o objeto que é armazenado no Amazon S3, o método retornará nulo.

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 mostrado no seguinte exemplo de código Ruby. O exemplo de código demonstra como copiar um objeto e criptografar a cópia.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyEncryptWrapper
  attr_reader :source_object
```

```
# @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is used
as the source object for
#                                         copy actions.
def initialize(source_object)
  @source_object = source_object
end

# Copy the source object to the specified target bucket, rename it with the target
key, and encrypt it.
#
# @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
object is copied.
# @param target_object_key [String] The key to give the copy of the object.
# @return [Aws::S3::Object, nil] The copied object when successful; otherwise, nil.
def copy_object(target_bucket, target_object_key, encryption)
  @source_object.copy_to(bucket: target_bucket.name, key: target_object_key,
server_side_encryption: encryption)
  target_bucket.object(target_object_key)
rescue Aws::Errors::ServiceError => e
  puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
#{e.message}"
end
end

# Replace the source and target bucket names with existing buckets you own and replace
the source object key
# with an existing object in the source bucket.
def run_demo
  source_bucket_name = "doc-example-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "doc-example-bucket2"
  target_key = "my-target-file.txt"
  target_encryption = "AES256"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyEncryptWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key, target_encryption)
  return unless target_object

  puts "Copied #{source_key} from #{source_bucket_name} to
#{target_object.bucket_name}:#{target_object.key} and \"\
      \"encrypted the target with #{target_object.server_side_encryption} encryption."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Uso do AWS CLI

Para especificar o SSE-S3 ao carregar um objeto usando a AWS CLI, use o exemplo a seguir.

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET1 --key object-key-name --server-side-
encryption AES256 --body file path
```

Para obter mais informações, consulte [put-object](#) na referência da AWS CLI. Para especificar o SSE-S3 ao copiar um objeto usando a AWS CLI, consulte [copy-object](#).

Usar o AWS CloudFormation

Para obter exemplos de configuração de criptografia usando o AWS CloudFormation, consulte [Criar um bucket com criptografia padrão](#) e [Criar um bucket usando criptografia do lado do servidor do AWS KMS com uma Chave de bucket do S3](#) no Guia do usuário do AWS CloudFormation.

Proteger dados usando a criptografia de servidor com chaves de criptografia fornecidas pelo cliente (SSE-C)

A criptografia de servidor envolve a proteção de dados em repouso. A criptografia no lado do servidor criptografa somente os dados de objeto, não os metadados de objeto. 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 a 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. Quando você recupera um objeto, deve fornecer a mesma chave de criptografia como parte de sua solicitação. O Amazon S3 primeiro verifica se a chave de criptografia fornecida é correspondente e, depois, decifra o objeto antes de retornar os dados de objeto.

Não há novas cobranças pelo uso da criptografia do lado do servidor com chaves de criptografia fornecidas pelo cliente (SSE-C). No entanto, as solicitações para configurar e usar o SSE-C incorrem em cobranças padrão de solicitação do Amazon S3. Para obter mais informações sobre definição de preços, consulte [Definição de preços do Amazon S3](#).

Important

O Amazon S3 não armazena a chave de criptografia que você fornece. Em vez disso, ele armazena 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.

Para obter mais informações sobre SSE-C, consulte os tópicos a seguir.

Tópicos

- [Visão geral do SSE-C \(p. 390\)](#)
- [Exigência e restrição de SSE-C \(p. 391\)](#)
- [Pre-signed URLs e SSE-C \(p. 392\)](#)
- [Especificação de criptografia no lado do servidor com chaves fornecidas pelo cliente \(SSE-C\). \(p. 392\)](#)

Visão geral do SSE-C

Esta seção fornece uma visão geral do SSE-C.

- Você deve usar HTTPS.

Important

O Amazon S3 rejeitará todas as solicitações feitas por HTTP ao usar SSE-C. Por questões de segurança, recomendamos considerar que todas as chaves enviadas erroneamente por HTTP estão comprometidas. Você deve descartar a chave e alterná-la 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.

Exigência e restrição de SSE-C

Para exigir criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C) para todos os objetos em um bucket específico do Amazon S3, é possível usar uma política de bucket.

Por exemplo, a política de bucket a seguir nega permissões para upload de objeto (s3:PutObject) a todas as solicitações que não incluem o cabeçalho x-amz-server-side-encryption-customer-algorithm que solicita SSE-C.

```
{  
    "Version": "2012-10-17",  
    "Id": "PutObjectPolicy",  
    "Statement": [  
        {  
            "Sid": "RequireSSECObjectUploads",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",  
            "Condition": {  
                "Null": {  
                    "s3:x-amz-server-side-encryption-customer-algorithm": "true"  
                }  
            }  
        }  
    ]  
}
```

Para restringir a criptografia do lado do servidor de todos os objetos em um bucket específico do Amazon S3, é possível usar uma política.

Por exemplo, a política de bucket a seguir negará permissão de upload de objeto (s3:PutObject) a todos se a solicitação incluir o cabeçalho x-amz-server-side-encryption-customer-algorithm que solicita criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C).

```
{  
    "Version": "2012-10-17",  
    "Id": "PutObjectPolicy",  
    "Statement": [  
        {  
            "Sid": "RestrictSSECObjectUploads",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",  
            "Condition": {  
                "Null": {  
                    "s3:x-amz-server-side-encryption-customer-algorithm": "false"  
                }  
            }  
        }  
    ]  
}
```

```
        }
    ]
}
```

Pre-signed URLs 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.

Especificação de criptografia no lado do servidor com chaves fornecidas pelo cliente (SSE-C).

No momento da criação do objeto com a API REST, você pode especificar a criptografia do lado do servidor com chaves de criptografia fornecidas pelo cliente (SSE-C). Ao usar o SSE-C, você deve fornecer informações da chave de encriptação utilizando os cabeçalhos de pedido a seguir.

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 com codificação base64 de 128 bits da chave de criptografia de acordo com RFC 1321 . O Amazon S3 usa esse cabeçalho para fazer uma verificação de integridade de mensagens e conferir se a chave de criptografia foi transmitida sem erros.

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 na aplicação.

Note

Não é possível usar o console do Amazon S3 para fazer upload de um objeto e solicitar 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.

Uso dos REST API

APIs REST do Amazon S3 que oferecem suporte ao SSE-C

As APIs do Amazon S3 a seguir oferecem suporte à criptografia pelo servidor com chaves de criptografia fornecidas pelo cliente (SSE-C).

- Operação GET: ao recuperar objetos usando a API GET (consulte [GET Object](#)), você pode especificar os cabeçalhos da solicitação.
- Operação HEAD: para recuperar metadados de objeto usando a API HEAD (consulte [HEAD Object](#)), você pode especificar esses cabeçalhos de solicitação.
- Operação PUT: ao fazer upload de dados usando a API PUT (consulte [PUT Object](#)), 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 (consulte [Fazer upload de parte](#)) ou

[Upload de parte - Copiar](#)

-). 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: quando você copia um objeto (consulte [Objeto PUT - Copiar](#)), tem um objeto de origem e um objeto de destino.
 - 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 pode 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 RFC 1321 .

Uso dos AWS SDKs para especificar SSE-C para operações PUT, GET, Head e Copy

O exemplo a seguir mostra como solicitar a criptografia de servidor com chaves fornecidas pelo cliente (SSE-C) para objetos. Os exemplos executam 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: faça download do objeto carregado na etapa anterior. Na solicitação, você fornece as mesmas informações de criptografia fornecidas quando o objeto foi carregado. O Amazon S3 precisa dessas informações para descriptografar o objeto e para que ele possa ser devolvido a você.
- Obter metadados do objeto: recupera os metadados do objeto. Você fornece as mesmas informações de criptografia usadas quando o objeto foi criado.
- Copy object (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 direciona o Amazon S3 a armazenar uma cópia criptografada do objeto.

Java

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 carregamentos fracionados usando o AWS SDK for Java, consulte [Fazer upload de um objeto usando multipart upload \(p. 187\)](#).

Para adicionar informações necessárias de criptografia, inclua uma `SSECustomerKey` na solicitação. Para obter mais informações sobre a classe `SSECustomerKey`, consulte a seção REST API.

Para obter informações sobre SSE-C, consulte [Proteger dados usando a criptografia de servidor com chaves de criptografia fornecidas pelo cliente \(SSE-C\) \(p. 390\)](#). Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

Example

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

import javax.crypto.KeyGenerator;
import java.io.BufferedReader;
import java.io.File;
import java.io.IOException;
import java.io.InputStreamReader;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;

public class ServerSideEncryptionUsingClientSideEncryptionKey {
    private static SSECustomerKey SSE_KEY;
    private static AmazonS3 S3_CLIENT;
    private static KeyGenerator KEY_GENERATOR;
```

```
    public static void main(String[] args) throws IOException, NoSuchAlgorithmException
{
    Regions clientRegion = Regions.DEFAULT_REGION;
    String bucketName = "*** Bucket name ***";
    String keyName = "*** Key name ***";
    String uploadFileName = "*** File path ***";
    String targetKeyName = "*** Target key name ***";

    // Create an encryption key.
    KEY_GENERATOR = KeyGenerator.getInstance("AES");
    KEY_GENERATOR.init(256, new SecureRandom());
    SSE_KEY = new SSECustomerKey(KEY_GENERATOR.generateKey());

    try {
        S3_CLIENT = AmazonS3ClientBuilder.standard()
            .withCredentials(new ProfileCredentialsProvider())
            .withRegion(clientRegion)
            .build();

        // Upload an object.
        uploadObject(bucketName, keyName, new File(uploadFileName));

        // Download the object.
        downloadObject(bucketName, keyName);

        // Verify that the object is properly encrypted by attempting to retrieve
it
        // using the encryption key.
        retrieveObjectMetadata(bucketName, keyName);

        // Copy the object into a new object that also uses SSE-C.
        copyObject(bucketName, keyName, targetKeyName);
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}

private static void uploadObject(String bucketName, String keyName, File file) {
    PutObjectRequest putRequest = new PutObjectRequest(bucketName, keyName,
file).withSSECustomerKey(SSE_KEY);
    S3_CLIENT.putObject(putRequest);
    System.out.println("Object uploaded");
}

private static void downloadObject(String bucketName, String keyName) throws
IOException {
    GetObjectRequest getObjectRequest = new GetObjectRequest(bucketName,
keyName).withSSECustomerKey(SSE_KEY);
    S3Object object = S3_CLIENT.getObject(getObjectRequest);

    System.out.println("Object content: ");
    displayTextInputStream(object.getObjectContent());
}

private static void retrieveObjectMetadata(String bucketName, String keyName) {
    GetObjectMetadataRequest getMetadataRequest = new
GetObjectMetadataRequest(bucketName, keyName)
        .withSSECustomerKey(SSE_KEY);
    ObjectMetadata objectMetadata =
S3_CLIENT.getObjectMetadata(getMetadataRequest);
```

```
        System.out.println("Metadata retrieved. Object size: " +
objectMetadata.getContentLength());
    }

    private static void copyObject(String bucketName, String keyName, String
targetKeyName)
        throws NoSuchAlgorithmException {
    // Create a new encryption key for target so that the target is saved using
SSE-C.
    SSECustomerKey newSSEKey = new SSECustomerKey(KEY_GENERATOR.generateKey());

    CopyObjectRequest copyRequest = new CopyObjectRequest(bucketName, keyName,
bucketName, targetKeyName)
        .withSourceSSECustomerKey(SSE_KEY)
        .withDestinationSSECustomerKey(newSSEKey);

    S3_CLIENT.copyObject(copyRequest);
    System.out.println("Object copied");
}

private static void displayTextInputStream(S3ObjectInputStream input) throws
IOException {
    // Read one line at a time from the input stream and display each line.
    BufferedReader reader = new BufferedReader(new InputStreamReader(input));
    String line;
    while ((line = reader.readLine()) != null) {
        System.out.println(line);
    }
    System.out.println();
}
}
```

.NET

Note

Para exemplos de upload de objetos grandes usando a API multipart upload, consulte [Fazer upload de um objeto usando multipart upload \(p. 187\)](#) e [Usar os AWS SDKs \(API de baixo nível\) \(p. 194\)](#).

Para obter informações sobre SSE-C, consulte [Proteger dados usando a criptografia de servidor com chaves de criptografia fornecidas pelo cliente \(SSE-C\) \(p. 390\)](#)). Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SSEClientEncryptionKeyObjectOperationsTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** key name for new object created ***";
        private const string copyTargetKeyName = "*** key name for object copy ***";
```

```
// Specify your bucket region (an example region is shown).
private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
private static IAmazonS3 client;

public static void Main()
{
    client = new AmazonS3Client(bucketRegion);
    ObjectOpsUsingClientEncryptionKeyAsync().Wait();
}
private static async Task ObjectOpsUsingClientEncryptionKeyAsync()
{
    try
    {
        // Create an encryption key.
        Aes aesEncryption = Aes.Create();
        aesEncryption.KeySize = 256;
        aesEncryption.GenerateKey();
        string base64Key = Convert.ToBase64String(aesEncryption.Key);

        // 1. Upload the object.
        PutObjectRequest putObjectRequest = await UploadObjectAsync(base64Key);
        // 2. Download the object and verify that its contents matches what you
uploaded.
        await DownloadObjectAsync(base64Key, putObjectRequest);
        // 3. Get object metadata and verify that the object uses AES-256
encryption.
        await GetObjectMetadataAsync(base64Key);
        // 4. Copy both the source and target objects using server-side
encryption with
        //      a customer-provided encryption key.
        await CopyObjectAsync(aesEncryption, base64Key);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered ***. Message:'{0}' when writing an
object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
}

private static async Task<PutObjectRequest> UploadObjectAsync(string base64Key)
{
    PutObjectRequest putObjectRequest = new PutObjectRequest
    {
        BucketName = bucketName,
        Key = keyName,
        ContentBody = "sample text",
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = base64Key
    };
    PutObjectResponse putObjectResponse = await
client.PutObjectAsync(putObjectRequest);
    return putObjectRequest;
}
private static async Task DownloadObjectAsync(string base64Key,
PutObjectRequest putObjectRequest)
{
    GetObjectRequest getObjectRequest = new GetObjectRequest
    {
        BucketName = bucketName,
        Key = keyName,
```

```
// Provide encryption information for the object stored in Amazon S3.  
ServerSideEncryptionCustomerMethod =  
ServerSideEncryptionCustomerMethod.AES256,  
    ServerSideEncryptionCustomerProvidedKey = base64Key  
};  
  
using (GetObjectResponse getResponse = await  
client.GetObjectAsync(getObjectRequest))  
    using (StreamReader reader = new StreamReader(getResponse.ResponseStream))  
{  
    string content = reader.ReadToEnd();  
    if (String.Compare(putObjectRequest.ContentBody, content) == 0)  
        Console.WriteLine("Object content is same as we uploaded");  
    else  
        Console.WriteLine("Error...Object content is not same.");  
  
    if (getResponse.ServerSideEncryptionCustomerMethod ==  
ServerSideEncryptionCustomerMethod.AES256)  
        Console.WriteLine("Object encryption method is AES256, same as we  
set");  
    else  
        Console.WriteLine("Error...Object encryption method is not the same  
as AES256 we set");  
  
    // Assert.AreEqual(putObjectRequest.ContentBody, content);  
    // Assert.AreEqual(ServerSideEncryptionCustomerMethod.AES256,  
getResponse.ServerSideEncryptionCustomerMethod);  
}  
}  
private static async Task GetObjectMetadataAsync(string base64Key)  
{  
    GetObjectMetadataRequest getObjectMetadataRequest = new  
GetObjectMetadataRequest  
{  
    BucketName = bucketName,  
    Key = keyName,  
  
    // The object stored in Amazon S3 is encrypted, so provide the  
necessary encryption information.  
    ServerSideEncryptionCustomerMethod =  
ServerSideEncryptionCustomerMethod.AES256,  
    ServerSideEncryptionCustomerProvidedKey = base64Key  
};  
  
GetObjectMetadataResponse getObjectMetadataResponse = await  
client.GetObjectMetadataAsync(getObjectMetadataRequest);  
Console.WriteLine("The object metadata show encryption method used is:  
{0}", getObjectMetadataResponse.ServerSideEncryptionCustomerMethod);  
// Assert.AreEqual(ServerSideEncryptionCustomerMethod.AES256,  
getObjectMetadataResponse.ServerSideEncryptionCustomerMethod);  
}  
private static async Task CopyObjectAsync(Aes aesEncryption, string base64Key)  
{  
    aesEncryption.GenerateKey();  
    string copyBase64Key = Convert.ToBase64String(aesEncryption.Key);  
  
CopyObjectRequest copyRequest = new CopyObjectRequest  
{  
    SourceBucket = bucketName,  
    SourceKey = keyName,  
    DestinationBucket = bucketName,  
    DestinationKey = copyTargetKeyName,  
    // Information about the source object's encryption.  
    CopySourceServerSideEncryptionCustomerMethod =  
ServerSideEncryptionCustomerMethod.AES256,  
    CopySourceServerSideEncryptionCustomerProvidedKey = base64Key,
```

```
// Information about the target object's encryption.  
ServerSideEncryptionCustomerMethod =  
    ServerSideEncryptionCustomerMethod.AES256,  
    ServerSideEncryptionCustomerProvidedKey = copyBase64Key  
};  
await client.CopyObjectAsync(copyRequest);  
}  
}  
}
```

Usar AWS SDKs para especificar SSE-C para uploads fracionados

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.

Java

Para carregar objetos grandes, você pode usar a API de carregamento fracionado (consulte [Carregar e copiar objetos usando multipart upload \(p. 180\)](#)). 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 [Fazer upload de um objeto usando multipart upload \(p. 187\)](#)).
- 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 os AWS SDKs \(API de baixo nível\) \(p. 194\)](#).

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. O Amazon S3 criptografa o objeto 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

```
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import com.amazonaws.auth.profile.ProfileCredentialsProvider;  
import com.amazonaws.regions.Regions;  
import com.amazonaws.services.s3.AmazonS3;  
import com.amazonaws.services.s3.AmazonS3ClientBuilder;  
import com.amazonaws.services.s3.model.CopyObjectRequest;  
import com.amazonaws.services.s3.model.PutObjectRequest;  
import com.amazonaws.services.s3.model.SSECustomerKey;  
import com.amazonaws.services.s3.transfer.Copy;
```

```
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.TransferManagerBuilder;
import com.amazonaws.services.s3.transfer.Upload;

import javax.crypto.KeyGenerator;
import java.io.File;
import java.security.SecureRandom;

public class ServerSideEncryptionCopyObjectUsingHLwithSSEC {

    public static void main(String[] args) throws Exception {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String fileToUpload = "*** File path ***";
        String keyName = "*** New object key name ***";
        String targetKeyName = "*** Key name for object copy ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withRegion(clientRegion)
                .withCredentials(new ProfileCredentialsProvider())
                .build();
            TransferManager tm = TransferManagerBuilder.standard()
                .withS3Client(s3Client)
                .build();

            // Create an object from a file.
            PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName,
keyName, new File(fileToUpload));

            // Create an encryption key.
            KeyGenerator keyGenerator = KeyGenerator.getInstance("AES");
            keyGenerator.init(256, new SecureRandom());
            SSECustomerKey sseCustomerEncryptionKey = new
SSECustomerKey(keyGenerator.generateKey());

            // Upload the object. TransferManager uploads asynchronously, so this call
returns immediately.
            putObjectRequest.setSSECustomerKey(sseCustomerEncryptionKey);
            Upload upload = tm.upload(putObjectRequest);

            // Optionally, wait for the upload to finish before continuing.
            upload.waitForCompletion();
            System.out.println("Object created.");

            // Copy the object and store the copy using SSE-C with a new key.
            CopyObjectRequest copyObjectRequest = new CopyObjectRequest(bucketName,
keyName, bucketName, targetKeyName);
            SSECustomerKey sseTargetObjectEncryptionKey = new
SSECustomerKey(keyGenerator.generateKey());
            copyObjectRequest.setSourceSSECustomerKey(sseCustomerEncryptionKey);

            copyObjectRequest.setDestinationSSECustomerKey(sseTargetObjectEncryptionKey);

            // Copy the object. TransferManager copies asynchronously, so this call
returns immediately.
            Copy copy = tm.copy(copyObjectRequest);

            // Optionally, wait for the upload to finish before continuing.
            copy.waitForCompletion();
            System.out.println("Copy complete.");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
```

```
// Amazon S3 couldn't be contacted for a response, or the client
// couldn't parse the response from Amazon S3.
e.printStackTrace();
}
}
```

.NET

Para carregar objetos grandes, você pode usar a API de carregamentos fracionados (consulte [Carregar e copiar objetos usando multipart upload \(p. 180\)](#)). AWS O SDK for .NET fornece 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 Transfer-Utility de alto nível, você fornece os cabeçalhos específicos de criptografia na TransferUtilityUploadRequest, conforme mostrado. Para obter exemplos de código, consulte [Fazer upload de um objeto usando multipart upload \(p. 187\)](#).

```
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 os AWS SDKs \(API de baixo nível\) \(p. 194\)](#).

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

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SSECLowLevelMPUcopyObjectTest
```

```
{  
    private const string existingBucketName = "*** bucket name ***";  
    private const string sourceKeyName      = "*** source object key name ***";  
    private const string targetKeyName     = "*** key name for the target object  
***";  
    private const string filePath          = @"*** file path ***";  
    // Specify your bucket region (an example region is shown).  
    private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;  
    private static IAmazonS3 s3Client;  
    static void Main()  
    {  
        s3Client = new AmazonS3Client(bucketRegion);  
        CopyObjClientEncryptionKeyAsync().Wait();  
    }  
  
    private static async Task CopyObjClientEncryptionKeyAsync()  
    {  
        Aes aesEncryption = Aes.Create();  
        aesEncryption.KeySize = 256;  
        aesEncryption.GenerateKey();  
        string base64Key = Convert.ToBase64String(aesEncryption.Key);  
  
        await CreateSampleObjUsingClientEncryptionKeyAsync(base64Key, s3Client);  
  
        await CopyObjectAsync(s3Client, base64Key);  
    }  
    private static async Task CopyObjectAsync(IAmazonS3 s3Client, string  
base64Key)  
    {  
        List<CopyPartResponse> uploadResponses = new List<CopyPartResponse>();  
  
        // 1. Initialize.  
        InitiateMultipartUploadRequest initiateRequest = new  
InitiateMultipartUploadRequest  
        {  
            BucketName = existingBucketName,  
            Key = targetKeyName,  
            ServerSideEncryptionCustomerMethod =  
ServerSideEncryptionCustomerMethod.AES256,  
            ServerSideEncryptionCustomerProvidedKey = base64Key,  
        };  
  
        InitiateMultipartUploadResponse initResponse =  
            await s3Client.InitiateMultipartUploadAsync(initiateRequest);  
  
        // 2. Upload Parts.  
        long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB  
        long firstByte = 0;  
        long lastByte = partSize;  
  
        try  
        {  
            // First find source object size. Because object is stored encrypted  
            with  
            // customer provided key you need to provide encryption information  
            in your request.  
            GetObjectMetadataRequest getObjectMetadataRequest = new  
GetObjectMetadataRequest()  
            {  
                BucketName = existingBucketName,  
                Key = sourceKeyName,  
                ServerSideEncryptionCustomerMethod =  
ServerSideEncryptionCustomerMethod.AES256,  
                ServerSideEncryptionCustomerProvidedKey = base64Key // *  
**source object encryption key ***"  
            };  
        }
```

```
        GetObjectMetadataResponse getObjectMetadataResponse = await
s3Client.GetObjectMetadataAsync(getObjectMetadataRequest);

        long filePosition = 0;
        for (int i = 1; filePosition <
getObjectMetadataResponse.ContentLength; i++)
{
    CopyPartRequest copyPartRequest = new CopyPartRequest
    {
        UploadId = initResponse.UploadId,
        // Source.
        SourceBucket = existingBucketName,
        SourceKey = sourceKeyName,
        // Source object is stored using SSE-C. Provide encryption
information.
        CopySourceServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        CopySourceServerSideEncryptionCustomerProvidedKey =
base64Key, //****source object encryption key ***",
        FirstByte = firstByte,
        // If the last part is smaller then our normal part size then
use the remaining size.
        LastByte = lastByte >
getObjectMetadataResponse.ContentLength ?
            getObjectMetadataResponse.ContentLength - 1 : lastByte,

        // Target.
        DestinationBucket = existingBucketName,
        DestinationKey = targetKeyName,
        PartNumber = i,
        // Encryption information for the target object.
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = base64Key
    };
    uploadResponses.Add(await
s3Client.CopyPartAsync(copyPartRequest));
    filePosition += partSize;
    firstByte += partSize;
    lastByte += partSize;
}

// Step 3: complete.
CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = targetKeyName,
    UploadId = initResponse.UploadId,
};
completeRequest.AddPartETags(uploadResponses);

CompleteMultipartUploadResponse completeUploadResponse =
    await s3Client.CompleteMultipartUploadAsync(completeRequest);
}
catch (Exception exception)
{
    Console.WriteLine("Exception occurred: {0}", exception.Message);
    AbortMultipartUploadRequest abortMPURequest = new
AbortMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = targetKeyName,
    UploadId = initResponse.UploadId
};
```

```
        s3Client.AbortMultipartUpload(abortMPUREquest);
    }
}
private static async Task CreateSampleObjUsingClientEncryptionKeyAsync(string
base64Key, IAmazonS3 s3Client)
{
    // List to store upload part responses.
    List<UploadPartResponse> uploadResponses = new
List<UploadPartResponse>();

    // 1. Initialize.
    InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
    {
        BucketName = existingBucketName,
        Key = sourceKeyName,
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = base64Key
    };

    InitiateMultipartUploadResponse initResponse =
        await s3Client.InitiateMultipartUploadAsync(initiateRequest);

    // 2. Upload Parts.
    long contentLength = new FileInfo(filePath).Length;
    long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB

    try
    {
        long filePosition = 0;
        for (int i = 1; filePosition < contentLength; i++)
        {
            UploadPartRequest uploadRequest = new UploadPartRequest
            {
                BucketName = existingBucketName,
                Key = sourceKeyName,
                UploadId = initResponse.UploadId,
                PartNumber = i,
                PartSize = partSize,
                FilePosition = filePosition,
                FilePath = filePath,
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key
            };

            // Upload part and add response to our list.
            uploadResponses.Add(await
s3Client.UploadPartAsync(uploadRequest));

            filePosition += partSize;
        }

        // Step 3: complete.
        CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
        {
            BucketName = existingBucketName,
            Key = sourceKeyName,
            UploadId = initResponse.UploadId,
            //PartETags = new List<PartETag>(uploadResponses)

        };
        completeRequest.AddPartETags(uploadResponses);
    }
}
```

```
        CompleteMultipartUploadResponse completeUploadResponse =  
            await s3Client.CompleteMultipartUploadAsync(completeRequest);  
  
    }  
    catch (Exception exception)  
    {  
        Console.WriteLine("Exception occurred: {0}", exception.Message);  
        AbortMultipartUploadRequest abortMPUREquest = new  
        AbortMultipartUploadRequest  
        {  
            BucketName = existingBucketName,  
            Key = sourceKeyName,  
            UploadId = initResponse.UploadId  
        };  
        await s3Client.AbortMultipartUploadAsync(abortMPUREquest);  
    }  
}
```

Proteger dados usando a criptografia no lado do cliente

A criptografia do lado do cliente é o ato de criptografar dados localmente para garantir sua segurança à medida que eles são transmitidos ao serviço do Amazon S3. O serviço Amazon S3 recebe os dados criptografados. Ele não atua na criptografia ou descriptografia desses dados.

Para habilitar a criptografia no lado do cliente, você tem as seguintes opções:

- Usar uma chave armazenada em AWS Key Management Service (AWS KMS).
- Use uma chave que você armazena na sua aplicação.

Note

O Amazon S3 só é compatível com chaves do KMS de criptografia simétrica, e não com chaves assimétricas do KMS. Para obter mais informações, consulte [Como usar chaves simétricas e assimétricas](#) no AWS Key Management Service Guia do desenvolvedor.

AWS Encryption SDK

O [AWS Encryption SDK](#) é uma biblioteca de criptografia do lado do cliente que é separada dos SDKs específicos da linguagem. Você pode usar essa biblioteca de criptografia para implementar com mais facilidade as práticas recomendadas no Amazon S3. Ao contrário dos clientes de criptografia do Amazon S3 nos AWS SDKs específicos da linguagem, o AWS Encryption SDK não é vinculado ao Amazon S3 e pode ser usado para criptografar ou descriptografar os dados a serem armazenados em qualquer lugar.

Os clientes de criptografia do AWS Encryption SDK e do Amazon S3 não são compatíveis, pois eles produzem textos cifrados com diferentes formatos de dados. Para obter mais informações sobre o AWS Encryption SDK, consulte o [Guia do desenvolvedor do AWS Encryption SDK](#).

AWSSuporte do SDK para criptografia do Amazon S3 do lado do cliente

Os AWS SDKs a seguir oferecem suporte à criptografia no lado do cliente:

- [AWS SDK for .NET](#)
- [AWS SDK for Go](#)

- AWS SDK for Java
- AWS SDK for PHP
- AWS SDK for Ruby
- AWS SDK for C++

Para obter informações e exemplos, consulte [Suporte do AWS SDK para criptografia do lado do cliente](#) na Referência geral da AWS.

Opção 1: usar uma chave do KMS armazenada no AWS KMS

Com essa opção, você usa uma AWS KMS key para criptografia do lado do cliente ao carregar ou baixar dados no Amazon S3.

- Ao carregar um objeto: usando o ID da chave do KMS, o cliente primeiro envia uma solicitação ao AWS KMS de uma chave de criptografia simétrica que possa ser usada para criptografar os dados do objeto. O AWS KMS retorna duas versões de uma chave de dados gerada aleatoriamente:
 - Uma versão em texto simples da chave de dados que o cliente usa para criptografar os dados de objeto.
 - Um blob de criptografia da mesma chave de dados que o cliente faz upload para o Amazon S3 como metadados de objeto.

Note

O cliente obtém uma chave de dados exclusiva para cada objeto cujo upload é feito.

- Ao 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 dados armazenada como metadados de objeto. Depois, o cliente envia o blob de criptografia para o AWS KMS para obter a versão em texto simples da chave de dados 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 Guia do desenvolvedor do AWS Key Management Service.

Example

O exemplo de código a seguir demonstra como carregar um objeto para o Amazon S3 usando o AWS KMS com o AWS SDK for Java. O exemplo usa uma chave gerenciada pela AWS para criptografar dados no lado do cliente antes de carregar no Amazon S3. Se já tiver uma chave do KMS, você poderá usá-la especificando o valor da variável `keyId` no código de exemplo. Se não tiver uma chave do KMS, ou precisar de outra, você poderá gerar uma por meio da API Java. O código de exemplo gera automaticamente uma chave do KMS a ser usada.

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
AWSKMS kmsClient = AWSKMSClientBuilder.standard()
    .withRegion(Regions.DEFAULT_REGION)
    .build();

    // create KMS key for testing this example
CreateKeyRequest createKeyRequest = new CreateKeyRequest();
CreateKeyResult createKeyResult = kmsClient.createKey(createKeyRequest);

// --
    // specify an AWS KMS key ID
String keyId = createKeyResult.getKeyMetadata().getKeyId();
```

```
String s3ObjectKey = "EncryptedContent1.txt";
String s3ObjectContent = "This is the 1st content to encrypt";
// --

AmazonS3EncryptionV2 s3Encryption = AmazonS3EncryptionClientV2Builder.standard()
    .withRegion(Regions.US_WEST_2)
    .withCryptoConfiguration(new
CryptoConfigurationV2().withCryptoMode(CryptoMode.StrictAuthenticatedEncryption))
    .withEncryptionMaterialsProvider(new KMSEncryptionMaterialsProvider(keyId))
    .build();

s3Encryption.putObject(bucket_name, s3ObjectKey, s3ObjectContent);
System.out.println(s3Encryption.getObjectAsString(bucket_name, s3ObjectKey));

// schedule deletion of KMS key generated for testing
ScheduleKeyDeletionRequest scheduleKeyDeletionRequest =
    new
ScheduleKeyDeletionRequest().withKeyId(keyId).withPendingWindowInDays(7);
kmsClient.scheduleKeyDeletion(scheduleKeyDeletionRequest);

s3Encryption.shutdown();
kmsClient.shutdown();
```

Opção 2: Use uma chave armazenada na sua aplicação.

Com essa opção, você usa uma chave-raiz armazenada na aplicação para a criptografia de dados do lado do cliente.

Important

Suas chaves 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 perdê-las, você não poderá descriptografar os seus dados.

Como funciona:

- Ao carregar um objeto: você fornece uma chave-raiz no lado do cliente para o cliente de criptografia do Amazon S3. O cliente usa a chave-raiz apenas para criptografar a chave de criptografia de dados que gera aleatoriamente.

As etapas a seguir descrevem o processo:

1. O cliente de criptografia do Amazon S3 gera localmente uma chave de criptografia 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-raiz 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-raiz 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-raiz para descriptografar a chave de dados e, em seguida, usa a chave de dados para descriptografar o objeto.

A chave raiz no lado do cliente que você fornece pode ser uma chave de criptografia simétrica ou um par de chaves pública/privada. Os exemplos de código a seguir mostram como usar cada tipo de chave.

Para obter mais informações, consulte [Criptografia de dados no lado do cliente com o AWS SDK for Java e o Amazon S3](#) e [Suporte do AWS SDK para criptografia do lado do cliente](#).

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 \(Java Cryptography Extension\)](#).

Example

O exemplo de código a seguir mostra como fazer essas tarefas:

- Gere uma chave AES de 256 bits.
- 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.
- Imprima uma representação de string do objeto descriptografado.

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
KeyGenerator keyGenerator = KeyGenerator.getInstance("AES");
keyGenerator.init(256);

// --
// generate a symmetric encryption key for testing
SecretKey secretKey = keyGenerator.generateKey();

String s3ObjectKey = "EncryptedContent2.txt";
String s3ObjectContent = "This is the 2nd content to encrypt";
// --

AmazonS3EncryptionV2 s3Encryption = AmazonS3EncryptionClientV2Builder.standard()
    .withRegion(Regions.DEFAULT_REGION)
    .withClientConfiguration(new ClientConfiguration())
    .withCryptoConfiguration(new
CryptoConfigurationV2().withCryptoMode(CryptoMode.AuthenticatedEncryption))
    .withEncryptionMaterialsProvider(new StaticEncryptionMaterialsProvider(new
EncryptionMaterials(secretKey)))
    .build();

s3Encryption.putObject(bucket_name, s3ObjectKey, s3ObjectContent);
System.out.println(s3Encryption.getObjectAsString(bucket_name, s3ObjectKey));
s3Encryption.shutdown();
```

Example

O exemplo de código a seguir mostra como fazer essas tarefas:

- Gere um par de chaves RSA de 2048 bits para fins de teste.
- 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.
- Imprima uma representação de string do objeto descriptografado.

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance("RSA");
keyPairGenerator.initialize(2048);

// --
// generate an asymmetric key pair for testing
KeyPair keyPair = keyPairGenerator.generateKeyPair();

String s3ObjectKey = "EncryptedContent3.txt";
String s3ObjectContent = "This is the 3rd content to encrypt";
// --

AmazonS3EncryptionV2 s3Encryption = AmazonS3EncryptionClientV2Builder.standard()
    .withRegion(Regions.US_WEST_2)
    .withCryptoConfiguration(new
CryptoConfigurationV2().withCryptoMode(CryptoMode.StrictAuthenticatedEncryption))
    .withEncryptionMaterialsProvider(new StaticEncryptionMaterialsProvider(new
EncryptionMaterials(keyPair)))
    .build();

s3Encryption.putObject(bucket_name, s3ObjectKey, s3ObjectContent);
System.out.println(s3Encryption.getObjectAsString(bucket_name, s3ObjectKey));
s3Encryption.shutdown();
```

Privacidade do tráfego entre redes

Este tópico descreve como o Amazon S3 protege conexões do serviço com outros locais.

Tráfego entre clientes de serviço e on-premises e os aplicativos

As seguintes conexões podem ser combinadas com o AWS PrivateLink para fornecer conectividade entre sua rede privada e a AWS:

- Uma conexão AWS Site-to-Site VPN. Para obter mais informações, consulte [O que é o AWS Site-to-Site VPN?](#)
- Uma conexão AWS Direct Connect. Para obter mais informações, consulte [O que é o AWS Direct Connect?](#)

O acesso ao Amazon S3 pela rede acontece por meio de APIs publicadas pela AWS. Os clientes devem ter suporte ao Transport Layer Security (TLS) 1.0. Recomendamos TLS 1.2 ou posterior. Os clientes também devem ter suporte a pacotes de criptografia com sigilo de encaminhamento perfeito (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos como Java 7 e versões posteriores oferece suporte a esses modos. Além disso, você deve assinar solicitações usando um ID da chave de acesso e uma chave de acesso secreta associados a uma entidade principal do IAM. Ou você pode usar o [AWS Security Token Service \(STS\)](#) para gerar credenciais de segurança temporárias para assinar solicitações.

Tráfego entre recursos da AWS na mesma região

Um endpoint da Virtual Private Cloud (VPC) para Amazon S3 é uma entidade lógica dentro de uma VPC que permite conectividade apenas com o Amazon S3. A VPC roteia as solicitações para o Amazon S3 e

as respostas de volta para a VPC. Para obter mais informações, consulte [Endpoints da VPC](#) no Guia do usuário da VPC. Para obter políticas de bucket de exemplo que podem ser usadas para controlar o acesso ao bucket do S3 de VPC endpoints, consulte [Controlar o acesso a partir de VPC endpoints com políticas de bucket \(p. 517\)](#).

AWS PrivateLink for Amazon S3

Com o AWS PrivateLink para o Amazon S3, você pode provisionar endpoints da VPC de interface (endpoints de interface) em sua Virtual Private Cloud (VPC). Esses endpoints são diretamente acessíveis a partir de aplicações que estão no local por meio de VPN e AWS Direct Connect, ou em uma Região da AWS diferente por emparelhamento de VPC.

Os endpoints de interface são representados por uma ou mais interfaces de rede elástica (ENIs) que recebem endereços IP privados de sub-redes em sua VPC. As solicitações feitas para endpoints de interface para o Amazon S3 são roteadas automaticamente para o Amazon S3 na rede da Amazon. Você também pode acessar endpoints de interface em sua VPC via aplicações on-premises por meio do AWS Direct Connect ou da AWS Virtual Private Network (AWS VPN). Para obter mais informações sobre como conectar sua VPC à rede on-premises, consulte o [Guia do usuário do AWS Direct Connect](#) e o [Guia do usuário do AWS Site-to-Site VPN](#).

Para obter informações gerais sobre endpoints de interface, consulte [Endpoints da VPC de interface \(AWS PrivateLink\)](#) no Manual do AWS PrivateLink.

Tópicos

- [Tipos de VPC endpoints para o Amazon S3 \(p. 410\)](#)
- [Restrições e limitações do AWS PrivateLink do Amazon S3 \(p. 411\)](#)
- [Criar um endpoint da VPC \(p. 411\)](#)
- [Acessar endpoints da interface do Amazon S3 \(p. 411\)](#)
- [Acessar buckets e pontos de acesso do S3 a partir de endpoints de interface S3 \(p. 412\)](#)
- [Atualizar uma configuração de DNS on-premises \(p. 415\)](#)
- [Criar uma política de VPC endpoint para o Amazon S3 \(p. 416\)](#)

Tipos de VPC endpoints para o Amazon S3

Você pode usar dois tipos de endpoint da VPC para acessar o Amazon S3: endpoints de gateway e endpoints de interface (com o uso do AWS PrivateLink). Um endpoint de gateway é um gateway que você especifica em sua tabela de rotas para acessar o Amazon S3 da sua VPC pela rede da AWS. Os endpoints de interface estendem a funcionalidade dos endpoints de gateway usando endereços IP privados para rotear solicitações para o Amazon S3 de dentro da sua VPC on-premises ou de uma PVC em outra Região da AWS usando emparelhamento da VPC ou o AWS Transit Gateway. Para obter mais informações, consulte [O que é emparelhamento da VPC?](#) em [Transit Gateway vs. emparelhamento da VPC](#).

Os endpoints de interface são compatíveis com os endpoints de gateway. Se você tiver um endpoint de gateway existente na VPC, poderá usar ambos os tipos de endpoints na mesma VPC.

Endpoints de gateway para o Amazon S3	Endpoints de interface para o Amazon S3
Em ambos os casos, o tráfego de rede permanece na rede AWS.	
Usar endereços IP públicos do Amazon S3	Usar endereços IP privados da VPC para acessar o Amazon S3

Endpoints de gateway para o Amazon S3	Endpoints de interface para o Amazon S3
Usar os mesmos nomes DNS do Amazon S3	Exigir nomes DNS do Amazon S3 específicos do endpoint
Não permite o acesso a partir do local	Permitir acesso desde on-premises
Não permitir o acesso usando outra Região da AWS	Permitir acesso de uma VPC em outra Região da AWS usando emparelhamento da VPC ou AWS Transit Gateway
Não faturado	Faturado

Para obter mais informações sobre endpoints de gateway, consulte [Endpoints da VPC de gateway](#) no Guia do usuário do AWS PrivateLink.

Restrições e limitações do AWS PrivateLink do Amazon S3

As limitações da VPC aplicam-se ao AWS PrivateLink do Amazon S3. Para obter mais informações, consulte [Propriedades e limitações de endpoints de interface](#) e [Cotas do AWS PrivateLink](#) no Guia do usuário do AWS PrivateLink. Além disso, aplicam-se as restrições a seguir.

O AWS PrivateLink do Amazon S3 não é compatível com:

- [Endpoints do Federal Information Processing Standard \(FIPS – Padrões Federais de Processamento de Informações\)](#)
- [Endpoints de site \(p. 1188\)](#)
- [Endpoints globais herdados \(p. 1254\)](#)
- Uso da [API CopyObject](#) ou [API UploadPartCopy](#) entre buckets em diferentes Regiões da AWS
- Transport Layer Security (TLS) 1.0

Criar um endpoint da VPC

Para criar um endpoint da VPC de interface, consulte [Create a VPC endpoint](#) (Criar um endpoint da VPC) no Guia do AWS PrivateLink.

Acessar endpoints da interface do Amazon S3

Important

Para acessar o Amazon S3 usando o AWS PrivateLink, você deve atualizar suas aplicações para usar nomes de DNS específicos de endpoint.

Quando você cria um endpoint de interface, o Amazon S3 gera dois tipos de nome de DNS do S3 específicos do endpoint: regional e zonal.

- Os nomes de DNS regionais incluem um ID de VPC endpoint exclusivo, um identificador de serviço, a Região da AWS e `vpce.amazonaws.com` em seu nome. Por exemplo, para o ID de VPC endpoint `vpce-1a2b3c4d`, o nome de DNS gerado pode ser semelhante a `vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com`.
- Os nomes de DNS zonais incluem a zona de disponibilidade. Por exemplo, `vpce-1a2b3c4d-5e6f-us-east-1a.s3.us-east-1.vpce.amazonaws.com`. Você pode usar essa

opção se sua arquitetura isola zonas de disponibilidade. Por exemplo, você pode usar para contenção de falhas ou para reduzir os custos regionais de transferência de dados.

Os nomes de DNS do S3 específicos do endpoint podem ser resolvidos a partir do domínio DNS público do S3.

Note

Os endpoints de interface do Amazon S3 não são compatíveis com o recurso DNS privado dos endpoints de interface. Para obter mais informações sobre [DNS privado para endpoints de interface](#), consulte o Guia do usuário do AWS PrivateLink.

Acessar buckets e pontos de acesso do S3 a partir de endpoints de interface S3

Você pode usar a AWS CLI ou o AWS SDK para acessar buckets, pontos de acesso do S3 e APIs de controle do S3 por meio de endpoints de interface do S3.

A imagem a seguir mostra a guia Detalhes do console da VPC, onde você pode encontrar o nome de DNS de um VPC endpoint. Neste exemplo, o ID do VPC endpoint (vpce-id) é `vpce-0e25b8cdd720f900e` e o nome de DNS é `*.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com`. Lembre-se de substituir `*` ao usar o Nome DNS. Por exemplo, para acessar um bucket, o DNS name (Nome DNS) seria `bucket.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com`.

Details	Subnets	Security Groups	Policy	Notifications	Tags
<p>Endpoint ID <code>vpce-0e25b8cdd720f900e</code> Status <code>available</code> Creation time January 8, 2021 at 1:30:11 AM UTC-8 Endpoint type Interface</p>				<p>VPC ID <code>vpc-0ccb9d87b1734bd</code> VPCStack VPC</p> <p>Status message Service name <code>com.amazonaws.us-east-1.s3</code> DNS names <code>*.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com</code> (ZTHUB22UJLQXV)</p>	

Para obter mais informações sobre como visualizar os nomes de DNS específicos do endpoint, consulte [Visualizar a configuração de nome de DNS privado do serviço de endpoint](#) no Guia do usuário da VPC.

Exemplos do AWS CLI

Use os parâmetros `--region` e `--endpoint-url` para acessar buckets do S3, pontos de acesso do S3 ou APIs de controle do S3 por meio de endpoints de interface do S3.

Exemplo: usar o URL de endpoint para listar objetos no bucket

No exemplo a seguir, substitua a região `us-east-1`, ID do endpoint de VPC `vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com` e o nome do bucket `my-bucket` pelas informações apropriadas.

```
aws s3 --region us-east-1 --endpoint-url https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com ls s3://my-bucket/
```

Exemplo: usar o URL do endpoint para listar objetos de um ponto de acesso

No exemplo a seguir, substitua o ARN `us-east-1:123456789012:accesspoint/test`, a região `us-east-1` e o ID do endpoint de VPC `vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com` pelas informações apropriadas.

```
aws s3api list-objects-v2 --bucket arn:aws:s3:us-east-1:123456789012:accesspoint/test --region us-east-1 --endpoint-url https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com
```

Exemplo: usar o URL do endpoint para listar trabalhos com controle do S3

No exemplo a seguir, substitua a região `us-east-1`, o ID do endpoint de VPC `vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com` e o ID da conta `12345678` pelas informações apropriadas.

```
aws s3control --region us-east-1 --endpoint-url https://control.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com list-jobs --account-id 12345678
```

Exemplos de AWS SDK

Atualize seus SDKs para a versão mais recente e configure seus clientes para usar um URL de endpoint para acessar um bucket, ponto de acesso ou API de controle do S3 por meio de endpoints de interface do S3.

SDK for Python (Boto3)

Exemplo: use um URL de endpoint para acessar um bucket do S3

No exemplo a seguir, substitua a região `us-east-1` e o ID do endpoint de VPC `vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com` pelas informações apropriadas.

```
s3_client = session.client(  
    service_name='s3',  
    region_name='us-east-1',  
    endpoint_url='https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com'  
)
```

Exemplo: use um URL de endpoint para acessar um ponto de acesso do S3

No exemplo a seguir, substitua a região `us-east-1` e o ID do endpoint de VPC `vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com` pelas informações apropriadas.

```
ap_client = session.client(  
    service_name='s3',  
    region_name='us-east-1',  
    endpoint_url='https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com'  
)
```

Exemplo: use um URL de endpoint para acessar a API de controle do S3

No exemplo a seguir, substitua a região `us-east-1` e o ID do endpoint de VPC `vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com` pelas informações apropriadas.

```
control_client = session.client(  
    service_name='s3control',  
    region_name='us-east-1',  
    endpoint_url='https://control.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com'  
)
```

SDK for Java 1.x

Exemplo: use um URL de endpoint para acessar um bucket do S3

No exemplo a seguir, substitua o ID do endpoint da VPC `vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com` por informações apropriadas.

```
// bucket client  
final AmazonS3 s3 = AmazonS3ClientBuilder.standard().withEndpointConfiguration(
```

```
new AwsClientBuilder.EndpointConfiguration(  
    "https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com",  
    Regions.DEFAULT_REGION.getName()  
)  
.build();  
List<Bucket> buckets = s3.listBuckets();
```

Exemplo: use um URL de endpoint para acessar um ponto de acesso do S3

No exemplo a seguir, substitua o ID do endpoint da VPC **vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com** e o ARN **us-east-1:123456789012:accesspoint/prod** por informações apropriadas.

```
// accesspoint client  
final AmazonS3 s3accesspoint =  
    AmazonS3ClientBuilder.standard().withEndpointConfiguration(  
        new AwsClientBuilder.EndpointConfiguration(  
            "https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com",  
            Regions.DEFAULT_REGION.getName()  
)  
.build();  
ObjectListing objects = s3accesspoint.listObjects("arn:aws:s3:us-east-1:123456789012:accesspoint/prod");
```

Exemplo: use um URL de endpoint para acessar a API de controle do S3

No exemplo a seguir, substitua o ID do endpoint da VPC **vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com** por informações apropriadas.

```
// control client  
final AWSS3Control s3control = AWSS3ControlClient.builder().withEndpointConfiguration(  
    new AwsClientBuilder.EndpointConfiguration(  
        "https://control.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com",  
        Regions.DEFAULT_REGION.getName()  
)  
.build();  
final ListJobsResult jobs = s3control.listJobs(new ListJobsRequest());
```

SDK for Java 2.x

Exemplo: use um URL de endpoint para acessar um bucket do S3

No exemplo a seguir, substitua o ID do endpoint da VPC **vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com** e a região **Region.US_EAST_1** por informações apropriadas.

```
// bucket client  
Region region = Region.US_EAST_1;  
s3Client = S3Client.builder().region(region)  
  
.endpointOverride(URI.create("https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com"))  
.build()
```

Exemplo: use um URL de endpoint para acessar um ponto de acesso do S3

No exemplo a seguir, substitua o ID do endpoint da VPC **vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com** e a região **Region.US_EAST_1** por informações apropriadas.

```
// accesspoint client  
Region region = Region.US_EAST_1;
```

```
s3Client = S3Client.builder().region(region)
    .endpointOverride(URI.create("https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com"))
    .build()
```

Exemplo: use um URL de endpoint para acessar a API de controle do S3

No exemplo a seguir, substitua o ID do endpoint da VPC `vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com` e a região `Region.US_EAST_1` por informações apropriadas.

```
// control client
Region region = Region.US_EAST_1;
s3ControlClient = S3ControlClient.builder().region(region)

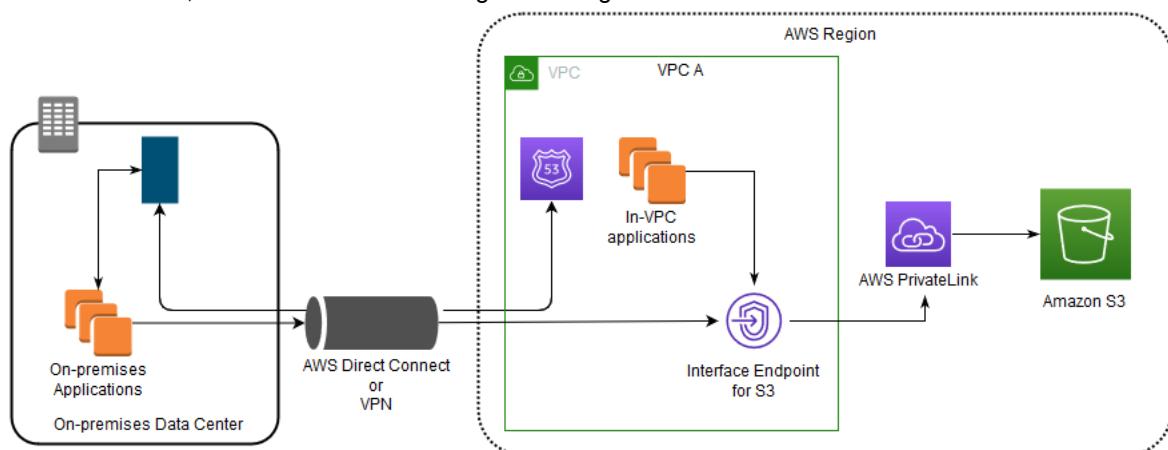
.endpointOverride(URI.create("https://control.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com"))
.build()
```

Atualizar uma configuração de DNS on-premises

Ao usar nomes de DNS específicos do endpoint para acessar os endpoints de interface do Amazon S3, você não precisa atualizar seu resolvedor de DNS on-premises. Você pode resolver o nome de DNS específico do endpoint com o endereço IP privado do endpoint de interface do domínio DNS público do Amazon S3.

Usar endpoints de interface para acessar o Amazon S3 sem um endpoint de gateway ou um gateway da Internet na VPC

Os endpoints de interface na VPC podem rotear aplicativos na VPC e on-premises para o Amazon S3 pela rede da Amazon, conforme ilustrado no diagrama a seguir.



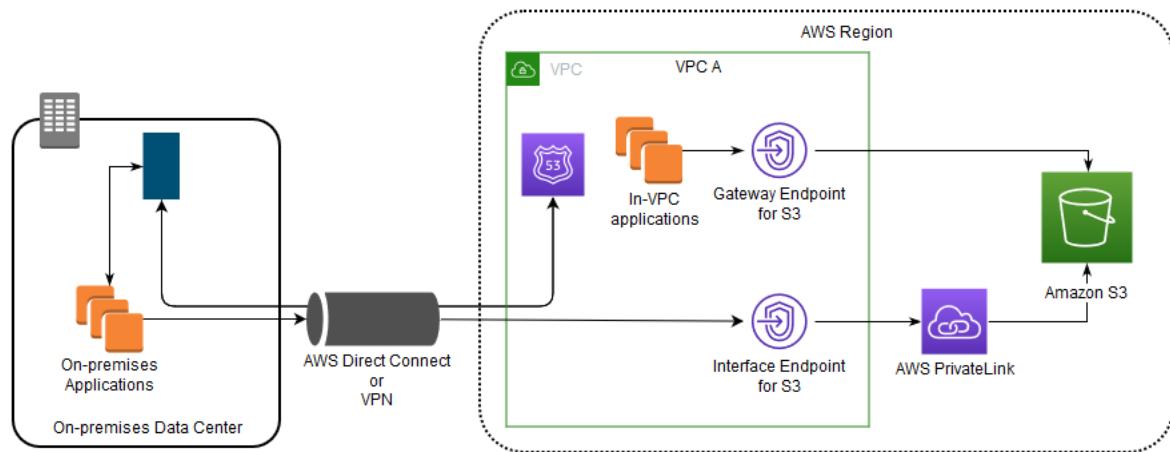
O diagrama ilustra o seguinte:

- Sua rede on-premises usa AWS Direct Connect ou AWS VPN para se conectar à VPC A.
- Seus aplicativos on-premises e na VPC A usam nomes de DNS específicos do endpoint para acessar o Amazon S3 por meio do endpoint de interface do S3.
- As aplicações on-premises enviam dados ao endpoint de interface na VPC via AWS Direct Connect (ou AWS VPN). O AWS PrivateLink move os dados do endpoint de interface para o Amazon S3 por meio da rede da AWS.

- As aplicações na VPC também enviam o tráfego ao endpoint de interface. O AWS PrivateLink move os dados do endpoint de interface para o Amazon S3 por meio da rede da AWS.

Usar endpoints de gateway e de interface juntos na mesma VPC para acessar o Amazon S3

Você pode criar endpoints de interface e reter o endpoint de gateway existente na mesma VPC, como mostra o diagrama a seguir. Ao fazer isso, você permite que as aplicações na VPC continuem acessando o Amazon S3 por meio do endpoint de gateway, que não é cobrado. Em seguida, apenas as aplicações on-premises usariam endpoints de interface para acessar o Amazon S3. Para acessar o S3 dessa maneira, você deve atualizar as aplicações on-premises para usar nomes de DNS específicos do endpoint para Amazon S3.



O diagrama ilustra o seguinte:

- As aplicações on-premises usam nomes de DNS específicos do endpoint para enviar dados ao endpoint de interface dentro da VPC via AWS Direct Connect (ou AWS VPN). O AWS PrivateLink move os dados do endpoint de interface para o Amazon S3 por meio da rede da AWS.
- Usando nomes regionais padrão do Amazon S3, as aplicações na VPC enviam dados ao endpoint de gateway que se conecta ao Amazon S3 pela rede da AWS.

Para obter mais informações sobre endpoints de gateway, consulte [Endpoints da VPC de gateway](#) no Guia do usuário da VPC.

Criar uma política de VPC endpoint para o Amazon S3

É possível anexar uma política de endpoint ao VPC endpoint que controla o acesso ao Amazon S3. Essa política especifica as seguintes informações:

- A entidade principal do AWS Identity and Access Management (IAM) que pode executar ações
- As ações que podem ser executadas
- Os recursos nos quais as ações podem ser executadas

Você também pode usar políticas de bucket do Amazon S3 para restringir o acesso a buckets específicos de um VPC endpoint específico usando a condição `aws:sourceVpc` na política de bucket. Os exemplos a seguir mostram políticas que restringem o acesso a um bucket ou a um endpoint.

Tópicos

- Exemplo: restringir o acesso a um bucket específico a partir de um endpoint da VPC (p. 417)
- Exemplo: restringir o acesso a buckets em uma conta específica a partir de um endpoint da VPC (p. 417)
- Exemplo: restringir o acesso a um endpoint da VPC específico na política de bucket do S3 (p. 418)

Important

- Ao aplicar políticas de bucket do Amazon S3 para os VPC endpoints descritos nesta seção, talvez você bloqueie o acesso ao bucket inadvertidamente. As permissões de bucket destinadas especificamente a 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 [Minha política de bucket tem o ID da VPC ou do endpoint da VPC incorreto. Como corrigir a política para que eu possa acessar o bucket?](#) na Central de conhecimento do AWS Support.
- Antes de usar a política de exemplo a seguir, substitua o ID do VPC endpoint por um valor apropriado para o caso de uso. Caso contrário, não será possível acessar o bucket.
- Essa política desabilita o acesso do console ao bucket especificado, pois as solicitações do console não se originam do VPC endpoint especificado.

Exemplo: restringir o acesso a um bucket específico a partir de um endpoint da VPC

Você pode criar uma política de endpoint que restringe o acesso apenas a buckets específicos do Amazon S3. Isso será útil se houver outros produtos da AWS em sua VPC que usam buckets do S3. A política de bucket a seguir restringe o acesso somente a [DOC-EXAMPLE-BUCKET1](#). Substitua [DOC-EXAMPLE-BUCKET1](#) pelo nome do seu bucket.

```
{  
    "Version": "2012-10-17",  
    "Id": "Policy1415115909151",  
    "Statement": [  
        { "Sid": "Access-to-specific-bucket-only",  
          "Principal": "*",  
          "Action": [  
              "s3:GetObject",  
              "s3:PutObject"  
            ],  
          "Effect": "Allow",  
          "Resource": [ "arn:aws:s3:::DOC-EXAMPLE-BUCKET1",  
                      "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*" ]  
        }  
    ]  
}
```

Exemplo: restringir o acesso a buckets em uma conta específica a partir de um endpoint da VPC

Você pode criar uma política que restrinja o acesso apenas aos buckets do S3 em uma Conta da AWS específica. Use isso para impedir que os clientes dentro de sua VPC acessem buckets que você não possui. O exemplo a seguir cria uma política que restringe o acesso a recursos pertencentes a um único ID de Conta da AWS, [111122223333](#).

```
{
```

```
"Statement": [
    {
        "Sid": "Access-to-bucket-in-specific-account-only",
        "Principal": "*",
        "Action": [
            "s3:GetObject",
            "s3:PutObject"
        ],
        "Effect": "Deny",
        "Resource": "arn:aws:s3:::*",
        "Condition": {
            "StringNotEquals": {
                "aws:ResourceAccount": "111122223333"
            }
        }
    }
]
```

Note

Você pode usar a chave `aws:ResourceAccount` ou `s3:ResourceAccount` em sua política do IAM para especificar o ID da Conta da AWS do recurso que está sendo acessado. No entanto, saiba que os serviços da AWS dependem do acesso a buckets gerenciados pela AWS. Portanto, usar a chave `aws:ResourceAccount` ou `s3:ResourceAccount` em sua política do IAM também pode afetar o acesso a esses recursos.

Exemplo: restringir o acesso a um endpoint da VPC específico na política de bucket do S3

Exemplo: restringir o acesso a um endpoint da VPC específico na política de bucket do S3

A política de bucket do Amazon S3 a seguir permite o acesso a um bucket específico, `DOC-EXAMPLE-BUCKET2`, apenas a partir do endpoint `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 e não requer um nome de recurso da Amazon (ARN) para o recurso de VPC endpoint, apenas o ID do endpoint. Substitua `DOC-EXAMPLE-BUCKET2` e `vpce-1a2b3c4d` por um nome de bucket real e endpoint.

```
{
    "Version": "2012-10-17",
    "Id": "Policy1415115909152",
    "Statement": [
        {
            "Sid": "Access-to-specific-VPCE-only",
            "Principal": "*",
            "Action": "s3:*",
            "Effect": "Deny",
            "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET2",
                        "arn:aws:s3:::DOC-EXAMPLE-BUCKET2/*"],
            "Condition": {"StringNotEquals": {"aws:sourceVpce": "vpce-1a2b3c4d"}}
        }
    ]
}
```

Para obter mais exemplos de políticas, consulte [Endpoints para o Amazon S3](#) no Guia do usuário da VPC.

Para obter mais informações sobre conectividade de VPC, consulte [Network-to-VPC connectivity options](#) (Opções de conectividade entre rede e VPC) no whitepaper da AWS [Amazon Virtual Private Cloud Connectivity Options](#) (Opções de conectividade do Amazon Virtual Private Cloud).

Identity and Access Management no Amazon S3

Por padrão, todos os recursos do Amazon S3, como buckets, objetos e sub-recursos relacionados (por exemplo, configuração de `lifecycle` e de `website`), são privados. Somente o proprietário do recurso, a Conta da AWS que o criou, pode acessar o recurso. 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 políticas de ponto de acesso são políticas baseadas 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. Também é possível usar as listas de controle de acesso (ACLs) para conceder permissões básicas de leitura/gravação a outras contas da Contas da AWS.

Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. É possível usar Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Como resultado, o controle de acesso para seus dados é baseado em políticas, como políticas do IAM, políticas de bucket do S3, políticas de endpoint da Virtual Private Cloud (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations.

A maioria dos casos de uso modernos no Amazon S3 não requer mais o uso de ACLs, e recomendamos desabilitar as ACLs, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com Object Ownership, é possível desabilitar ACLs e confiar em políticas para controle de acesso. Ao desabilitar ACLs, você pode facilmente manter um bucket com objetos carregados por diferentes Contas da AWS. Como proprietário do bucket, você possui todos os objetos do bucket e pode gerenciar o acesso a eles usando políticas. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Para obter mais informações sobre como gerenciar o acesso aos seus objetos e buckets do Amazon S3, consulte os tópicos abaixo.

Tópicos

- [Visão geral do gerenciamento de acesso \(p. 419\)](#)
- [Diretrizes para políticas de acesso \(p. 425\)](#)
- [Como o Amazon S3 autoriza uma solicitação \(p. 429\)](#)
- [Políticas de bucket e políticas de usuário \(p. 437\)](#)
- [Políticas gerenciadas da AWS para o Amazon S3 \(p. 585\)](#)
- [Gerenciar o acesso com ACLs \(p. 587\)](#)
- [Usar o compartilhamento de recursos de origem cruzada \(CORS\) \(p. 607\)](#)
- [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#)
- [Revisar o acesso de bucket usando o Access Analyzer for S3 \(p. 629\)](#)
- [Verificar a propriedade do bucket com a condição de proprietário do bucket \(p. 634\)](#)

Visão geral do gerenciamento de acesso

Ao conceder permissões no Amazon S3, você decide quem as recebe, a quais recursos do Amazon S3 as permissões se referem e as ações específicas que deseja permitir nesses recursos. As seções a seguir fornecem uma visão geral dos recursos do Amazon S3 e como determinar o melhor método para controlar o acesso a eles.

Tópicos

- [Recursos do Amazon S3: buckets e objetos \(p. 420\)](#)
- [Propriedade de bucket e objeto do Amazon S3 \(p. 420\)](#)
- [Operações de recurso \(p. 421\)](#)
- [Gerenciar o acesso aos recursos \(p. 422\)](#)
- [Qual método de controle de acesso devo usar? \(p. 425\)](#)

Recursos do Amazon S3: buckets e objetos

Na AWS, um recurso é uma entidade com a qual você pode trabalhar. No Amazon S3, os buckets e os objetos são os recursos e ambos têm sub-recursos associados.

Os sub-recursos de bucket incluem o seguinte:

- `lifecycle`: armazena informações sobre a configuração do ciclo de vida. Para obter mais informações, consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).
- `website`: armazena informações de configuração do site se você configurar o bucket para hospedagem de sites. Para obter mais informações, consulte [Hospedagem de um site estático usando o Amazon S3 \(p. 1188\)](#).
- `versioning`: armazena a configuração de versionamento. Para obter mais informações, consulte [Versionamento do bucket PUT](#) na Referência da API do Amazon Simple Storage Service.
- `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. Para obter mais informações, consulte [Usar o compartilhamento de recursos de origem cruzada \(CORS\) \(p. 607\)](#).
- `object ownership`: permite que o proprietário do bucket assuma a propriedade de novos objetos no bucket, independentemente de quem faz upload deles. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).
- `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. Para obter mais informações, consulte [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#).
- `restore`: oferece suporte para restaurar, temporariamente, um objeto arquivado. Para obter mais informações, consulte [Restauração do objeto POST](#) na Referência de APIs do Amazon Simple Storage Service.

Um objeto na classe de armazenamento do S3 Glacier Flexible Retrieval 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 [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

Propriedade de bucket e objeto do Amazon S3

Buckets e objetos são recursos do Amazon S3. Por padrão, apenas o proprietário dos recursos pode acessá-los. 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 carregar objetos é a proprietária desses recursos.

- Se você carregar um objeto usando credenciais de função ou usuário do AWS Identity and Access Management (IAM), a Conta da AWS à qual o usuário ou a função pertence será proprietária do objeto.
- Um proprietário do bucket pode conceder permissões entre contas à outra Conta da AWS (ou aos usuários em outra conta) para carregar objetos. Nesse caso, a Conta da AWS que carregar os objetos será proprietária deles. 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 [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

Autenticação de solicitação e propriedade

Todas as solicitações para um bucket são autenticadas ou não autenticadas. As solicitações autenticadas devem incluir um valor de assinatura que autentica o remetente da solicitação. Para solicitações não autenticadas, isso não é necessário. Para obter mais informações sobre autenticação da solicitação, consulte [Fazer solicitações \(p. 1211\)](#).

Um proprietário do bucket pode permitir solicitações não autenticadas. Por exemplo, solicitações [PUT Object](#) não autenticadas são permitidas quando um bucket tem uma política pública de bucket, ou quando uma ACL do bucket concede acesso `WRITE` ou `FULL_CONTROL` ao grupo All Users (Todos os usuários) ou ao usuário anônimo especificamente. Para obter mais informações sobre políticas de bucket público e listas de controle de acesso (ACLs) públicas, consulte [O significado de "público" \(p. 622\)](#).

Todas as solicitações não autenticadas são feitas pelo usuário anônimo. Esse usuário é representado em ACLs pelo ID de usuário canônico específico `65a011a29cdf8ec533ec3d1ccaae921c`. Se for feito o upload de um objeto em um bucket por meio de uma solicitação não autenticada, o usuário anônimo será proprietário do objeto. A ACL padrão do objeto concede `FULL_CONTROL` ao usuário anônimo como o proprietário do objeto. Portanto, o Amazon S3 permite que solicitações não autenticadas recuperem o objeto ou modifiquem a ACL dele.

Para evitar que os objetos sejam modificados pelo usuário anônimo, recomendamos que você não implemente políticas de bucket que permitam gravações públicas anônimas em seu bucket ou use ACLs que concedam ao usuário anônimo acesso de gravação ao bucket. Você pode aplicar esse comportamento recomendado usando o Bloqueio de acesso público do Amazon S3.

Para obter mais informações sobre como bloquear o acesso público, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#). Para obter mais informações sobre ACLs, consulte [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#).

Important

Recomendamos que você não use credenciais de usuário raiz da Conta da AWS para fazer solicitações autenticadas. Em vez disso, crie um usuário do IAM e conceda acesso total a ele. Esses usuários são conhecidos como usuários administradores. As credenciais do usuário administrador podem ser usadas, em vez das credenciais do usuário raiz da Conta da AWS, para interagir com a AWS e executar tarefas, como criar um bucket, criar usuários e conceder permissões a eles. Para obter mais informações, consulte [Credenciais de usuário root da Conta da AWS e credenciais de usuário do IAM](#) na Referência geral da AWS e [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Operações de recurso

O Amazon S3 fornece um conjunto de operações para trabalhar com recursos do Amazon S3. Para ver uma lista das operações disponíveis, consulte [Ações definidas pelo Amazon S3 \(p. 458\)](#).

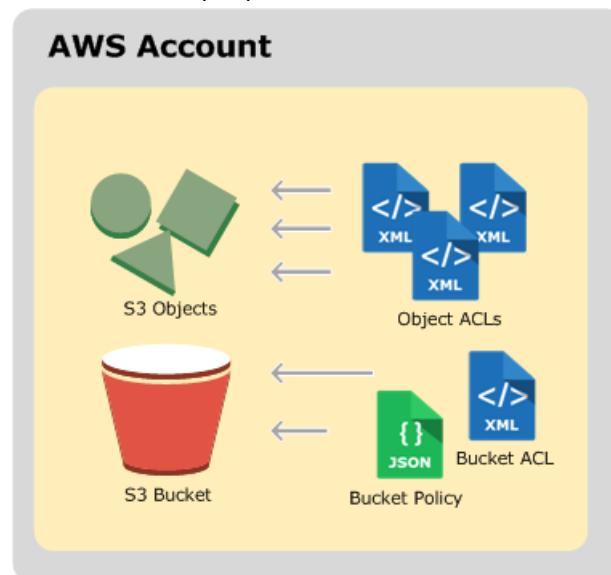
Gerenciar o acesso aos recursos

Gerenciar o acesso refere-se à concessão de permissões a outros (usuários e Contas da AWS) para realizar operações de recurso por meio da criação de uma política de acesso. Por exemplo, você pode conceder a permissão `PUT Object` a um usuário em uma Conta da AWS para que ele possa carregar 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.

Opções de política de acesso

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) são com base em recurso porque você as anexa aos recursos do Amazon S3.



- ACL – Cada bucket e objeto tem uma ACL associada. Uma ACL é uma lista de concessões que identifica o concessionário e a permissão concedida. Você usa ACLs para conceder permissões básicas de leitura/gravação 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>
```

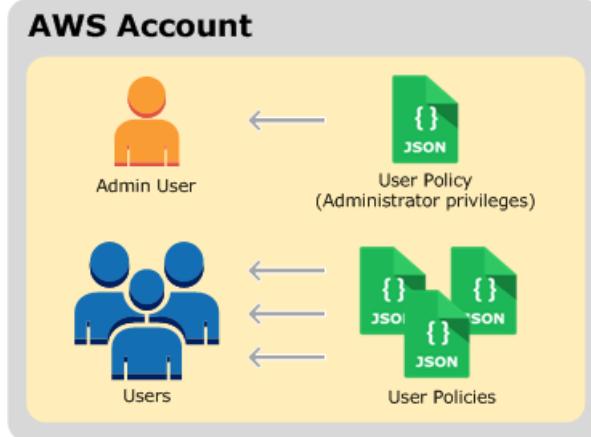
As ACLs de bucket e objeto usam o mesmo esquema XML.

- Política de bucket: 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. Tenha cuidado ao fazer isso. Por exemplo, a seguinte política de bucket tornaria os objetos publicamente acessíveis.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GrantAnonymousReadPermissions",
            "Effect": "Allow",
            "Principal": "*",
            "Action": ["s3:GetObject"],
            "Resource": ["arn:aws:s3:::awsexamplebucket1/*"]
        }
    ]
}
```

- 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.



Para obter mais informações sobre o IAM, consulte [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": "AssignUserActions",
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:GetObject",
            "s3>ListBucket",
            "s3>DeleteObject",
            "s3:GetBucketLocation"
        ],
        "Resource": [
            "arn:aws:s3:::awsexamplebucket1/*",
            "arn:aws:s3:::awsexamplebucket1"
        ]
    },
    {
        "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 \(p. 429\)](#).

Analisador de acesso para S3

No console do Amazon S3, você pode usar o Analisador de acesso para S3 para revisar todos os buckets com listas de controle de acesso (ACLs) de bucket, políticas de bucket ou políticas de ponto de acesso que concedem acesso público ou compartilhado. O Access Analyzer for S3 alerta você sobre buckets que são configurados para permitir o acesso a qualquer pessoa na Internet ou a outras Contas da AWS, incluindo Contas da AWS fora da organização. Para cada bucket público ou compartilhado, você recebe descobertas que relatam a origem e o nível de acesso público ou compartilhado.

No Analisador de acesso para S3, você pode bloquear todo o acesso público a um bucket com um único clique. Recomendamos que você bloquee todo o acesso aos buckets, a menos que exija acesso público para dar suporte a um caso de uso específico. Antes de bloquear todo o acesso público, certifique-se de que os aplicativos continuarão funcionando corretamente sem acesso público. Para obter mais informações, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

Você também pode detalhar as configurações de permissão no nível do bucket para definir níveis granulares de acesso. Para casos de uso específicos e verificados que exigem acesso público ou compartilhado, você pode confirmar e registrar sua intenção de que o bucket permaneça público ou compartilhado arquivando as descobertas do bucket. Você pode acessar novamente e modificar essas configurações de bucket a qualquer momento. Você também pode fazer download das descobertas como um relatório CSV para fins de auditoria.

O Analisador de acesso para S3 está disponível sem custo adicional no console do Amazon S3. O Access Analyzer for S3 usa tecnologia do AWS Identity and Access Management (IAM) Access Analyzer. Para usar o Analisador de acesso para S3 no console do Amazon S3, você deve visitar o console do IAM e criar um analisador no nível da conta no Analisador de acesso do IAM por região.

Para obter mais informações sobre o Access Analyzer for S3, consulte [Revisar o acesso de bucket usando o Access Analyzer for S3 \(p. 629\)](#).

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.

- [Diretrizes para políticas de acesso \(p. 425\)](#)
- [Como o Amazon S3 autoriza uma solicitação \(p. 429\)](#)
- [Demonstrações de exemplo: gerenciar o acesso aos recursos do Amazon S3 \(p. 553\)](#)
- [Práticas recomendadas de controle de acesso \(p. 24\)](#)

Diretrizes para políticas 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. Para obter mais informações, consulte [Gerenciar o acesso aos recursos \(p. 422\)](#). As políticas com base em recursos incluem políticas de bucket, listas de controle de acesso (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.

Tópicos

- [Quando usar uma política de acesso com base em ACL \(ACLs de bucket e objeto\) \(p. 425\)](#)
- [Quando usar uma política de bucket \(p. 427\)](#)
- [Quando usar uma política de usuário \(p. 427\)](#)
- [Tópicos relacionados \(p. 428\)](#)

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.

Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. É possível usar Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Como resultado, o controle de acesso para seus dados é baseado em políticas, como políticas do IAM, políticas de bucket do S3, políticas de endpoint da Virtual Private Cloud (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations.

A maioria dos casos de uso modernos no Amazon S3 não requer mais o uso de ACLs, e recomendamos desabilitar as ACLs, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com Object Ownership, é possível desabilitar ACLs e confiar em políticas para controle de acesso. Ao desabilitar ACLs, você pode facilmente manter um bucket com objetos carregados por diferentes Contas da AWS. Como proprietário do bucket, você possui todos os objetos do bucket e pode gerenciar o acesso a eles usando políticas. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Important

Se o bucket usar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Solicitações para definir ACLs ou atualizar ACLs falham e retornam o código de erro `AccessControlListNotSupported`. Ainda há suporte para solicitações de leitura de ACLs.

Quando usar uma ACL de objeto

Os cenários a seguir mostram quando você usa ACLs de objeto para gerenciar permissões.

Objetos não pertencentes ao proprietário do bucket

Uma ACL de objeto é a única maneira de gerenciar o acesso a objetos não pertencentes ao proprietário do bucket. Uma Conta da AWS que seja proprietária do bucket pode conceder permissão à outra Conta da AWS para carregar 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.

Você precisa gerenciar permissões no nível do objeto

Suponha que as permissões variem de acordo com o objeto e você precise gerenciar permissões no nível do objeto. É possível gravar uma única instrução de política que conceda a uma Conta da AWS permissão de leitura e milhões de objetos com um [prefixo específico de nome de chave](#). Por exemplo, você pode conceder permissão de leitura em objetos que começam com o prefixo de nome de chave `logs`. Contudo, se as permissões de acesso variam por objeto, conceder permissões para objetos individuais usando uma política de bucket talvez não seja prático. Além disso, o tamanho das políticas de bucket é limitado a 20 KB.

Nesse caso, você pode considerar o uso de ACLs de objeto como uma boa alternativa. No entanto, mesmo uma ACL de objeto é limitada a, no máximo, 100 concessões. Para obter mais informações, consulte [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#).

As ACLs de objeto controlam somente permissões no nível do objeto

Há uma 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 permissão a outra Conta da AWS para gerenciar a política de acesso. Isso permite que a conta altere algo na política. Para gerenciar permissões melhor, você poderá 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.

Para usar ACLs para gerenciar permissões no nível do objeto e ter propriedade sobre novos objetos gravados em seu bucket, você pode aplicar a configuração preferencial do proprietário do bucket em Object Ownership. Um bucket com a configuração preferencial do proprietário do bucket continua aceitando e respeitando as ACLs de bucket e objeto. Com essa configuração, novos objetos que são

escritos com a ACL `bucket-owner-full-control` pré-configurada serão automaticamente de propriedade do proprietário do bucket, e não do gravador de objeto. Todos os outros comportamentos da ACL permanecem em vigor. Para exigir que todas as operações PUT do Amazon S3 incluam a ACL `bucket-owner-full-control` pré-configurada, [adicone uma política de bucket \(p. 661\)](#) que permita somente o carregamento de objetos que usam essa ACL.

Alternativas ao uso de ACLs

Além de uma ACL de objeto, o proprietário de objeto pode gerenciar permissões de objeto de outras formas:

- Se a Conta da AWS que é proprietária do objeto também for proprietária do bucket, ela poderá criar uma política de bucket para gerenciar permissões de objeto.
- Se a Conta da AWS que é proprietária do objeto quiser conceder permissão a um usuário na respectiva conta, ela poderá usar uma política de usuário.
- Como proprietário do bucket, se você quiser ter automaticamente a propriedade e o controle total sobre cada objeto de seu bucket, aplique a configuração aplicada pelo proprietário do bucket em Object Ownership para desabilitar ACLs. Consequentemente, o controle de acesso para seus dados é baseado em políticas. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Quando usar uma ACL de bucket

O único caso de uso recomendado para ACLs de bucket é conceder permissões a determinados Serviços da AWS, como a conta de `awslogsdelivery` do Amazon CloudFront. Ao criar ou atualizar uma distribuição e habilitar o registro do CloudFront, o CloudFront atualiza a ACL do bucket para fornecer à conta `awslogsdelivery` permissões `FULL_CONTROL` para registrar logs em seu bucket. Para obter mais informações, consulte [Permissões necessárias para configurar o registro padrão e acessar seus arquivos de log](#) no Guia do desenvolvedor do Amazon CloudFront. Se o bucket que armazena os logs usar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership para desabilitar ACLs, o CloudFront não poderá gravar logs no bucket. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Quando usar uma política de bucket

Se uma Conta da AWS que é proprietária de um bucket quiser conceder permissão aos usuários na respectiva conta, ela poderá usar uma política de bucket ou de usuário. Porém, nos cenários a seguir, é necessário usar uma política de bucket.

Você deseja 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 são compatíveis apenas com um conjunto finito de permissões, e essas não incluem todas as permissões do Amazon S3. Para obter mais informações, consulte [Quais permissões posso conceder? \(p. 591\)](#). Por exemplo, você não pode conceder permissões em sub-recursos de bucket. Para obter mais informações, consulte [Identity and Access Management no Amazon S3 \(p. 419\)](#).

As políticas de bucket e de usuário são compatíveis com a concessão de permissão para todas as operações do Amazon S3. (Para obter mais informações, consulte [Ações do Amazon S3 \(p. 441\)](#).) No entanto, as políticas de usuário se destinam a 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

com a associação de políticas a usuários (ou grupos de usuários). Também é possível que você considere as políticas baseadas em recursos, como a política de bucket, funcionam melhor para o seu cenário.

Com o AWS Identity and Access Management (IAM), é possível criar vários usuários em sua Conta da AWS e gerenciar as permissões deles por meio de políticas de usuário. Um usuário do IAM deve ter permissões da conta pai a qual ele pertence e da Conta da AWS que é proprietária do recurso que ele 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 pertence a outra pessoa. Nesse caso, a criança precisa obter permissão de um responsável para brincar com o brinquedo e permissão do proprietário do brinquedo.

Para obter mais informações, consulte [Políticas de bucket e políticas de usuário \(p. 437\)](#).

Delegação de permissão

Se uma Conta da AWS for proprietária de um recurso, ela poderá conceder essas permissões a 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. Porém, uma conta que recebe permissões de outra conta não pode delegar permissão entre contas para outra Conta da AWS.

Tópicos relacionados

Recomendamos analisar, primeiro, todos os tópicos introdutórios que explicam como gerenciar o acesso aos recursos do Amazon S3 e as diretrizes relacionadas. Para obter mais informações, consulte [Identity and Access Management no Amazon S3 \(p. 419\)](#). Você pode usar os seguintes tópicos para obter mais informações sobre as opções específicas da política de acesso.

- [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#)
- [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#)

Como o Amazon S3 autoriza uma solicitação

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.

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.

Essa etapa será aplicável somente 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 essa 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.

Veja a seguir alguns exemplos de cenários que ilustram como o Amazon S3 autoriza uma solicitação.

Example O solicitante é uma entidade principal do IAM

Se o solicitante for uma entidade principal do IAM, o Amazon S3 deverá determinar se a Conta da AWS pai à qual a entidade principal pertence concedeu a permissão necessária para ele 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. Para executar uma operação específica em um recurso, uma entidade principal do IAM precisa de permissão da Conta da AWS pai à qual ele pertence e da Conta da AWS que é proprietária do recurso.

Example O solicitante é uma entidade principal do IAM: se a solicitação for para uma operação em um objeto que não pertence ao proprietário do bucket.

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. Um proprietário de 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.

Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. É possível usar Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Como resultado, o controle de acesso para seus dados é baseado em políticas, como políticas do IAM, políticas de bucket do S3, políticas de endpoint da Virtual Private Cloud (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Para obter mais informações sobre como o Amazon S3 avalia as políticas de acesso para autorizar ou negar solicitações para operações de bucket e operações de objeto, consulte os seguintes tópicos:

Tópicos

- [Como o Amazon S3 autoriza uma solicitação para uma operação de bucket \(p. 430\)](#)
- [Como o Amazon S3 autoriza uma solicitação para uma operação de objeto \(p. 433\)](#)

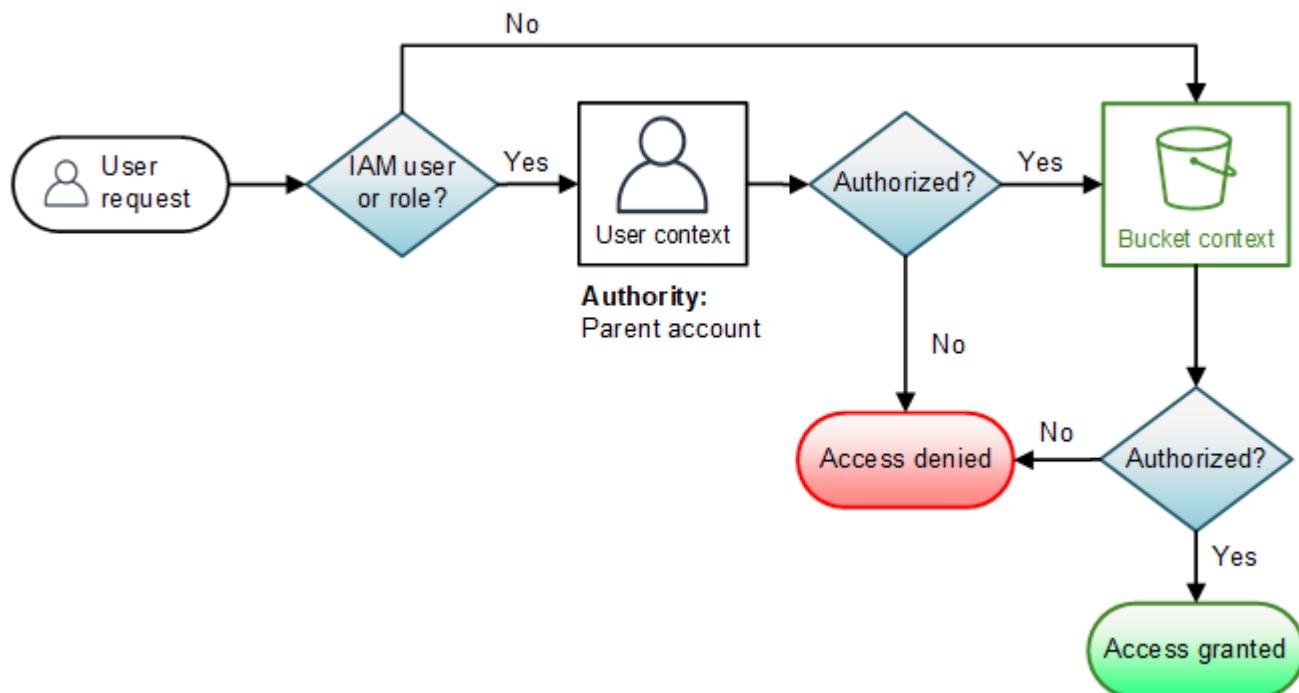
Como o Amazon S3 autoriza uma solicitação para uma operação de bucket

Quando o Amazon S3 recebe uma solicitação para uma operação de bucket, ele converte todas as permissões relevantes em um conjunto de políticas para avaliar em tempo de execução. As permissões relevantes incluem permissões baseadas em recursos (por exemplo, políticas de bucket e listas de controle de acesso de bucket) e políticas de usuário do IAM se a solicitação vier de um principal do IAM. Em seguida, o Amazon S3 avalia o conjunto de políticas resultante em uma série de etapas, de acordo com o contexto específico: contexto de usuário ou de bucket.

1. Contexto do usuário: se o solicitante for um principal do IAM, o principal deverá ter permissão da Conta da AWS pai à qual ele 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 principal. 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 solicitante. Para obter mais informações, consulte [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#).
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 é proprietária do 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 é proprietária do bucket também for a conta pai de uma entidade principal do IAM, ela poderá configurar permissões de bucket em uma política de usuário.

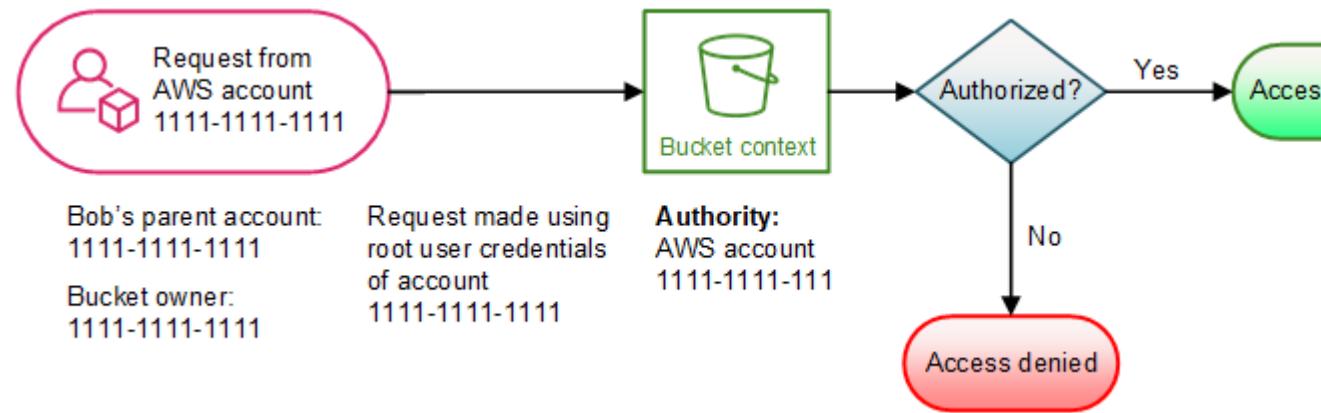
O seguinte é uma ilustração gráfica da avaliação baseada em contexto para a operação de bucket.



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 a uma operação de bucket usando as credenciais raiz da Conta da AWS.

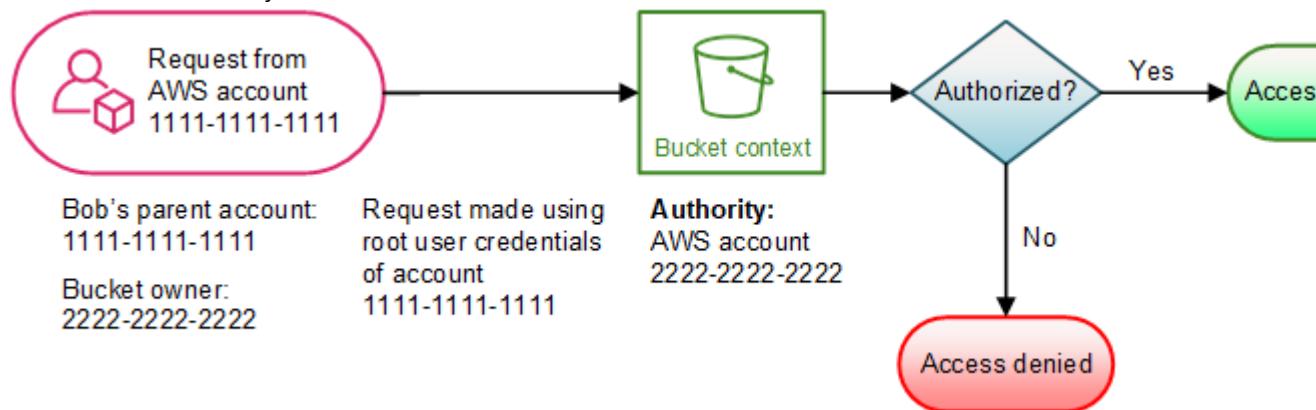


O Amazon S3 executa a avaliação de contexto da seguinte forma:

1. Como a solicitação é feita usando credenciais raiz de uma Conta da AWS, o contexto do 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 a uma operação de bucket pertencente à Conta da AWS 2222-2222-2222. Nenhum usuário do IAM está envolvido nessa solicitação.

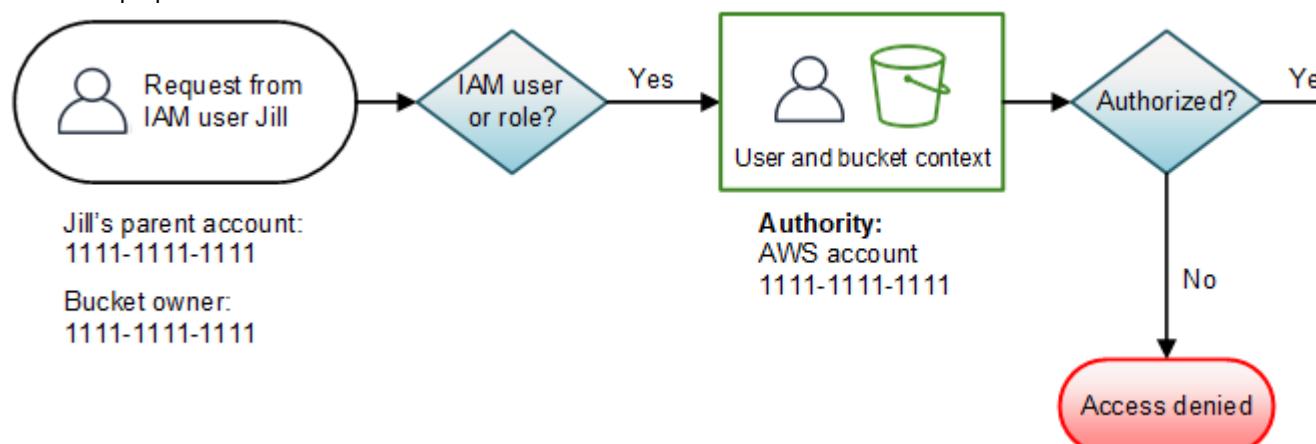


Neste caso, o Amazon S3 avalia o contexto da seguinte forma:

1. Como a solicitação é feita usando credenciais raiz de uma Conta da AWS, o contexto do 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 tiver autorizado 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 uma entidade principal do IAM cuja Conta da AWS pai 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 é proprietária do bucket.



O Amazon S3 executa a seguinte avaliação de contexto:

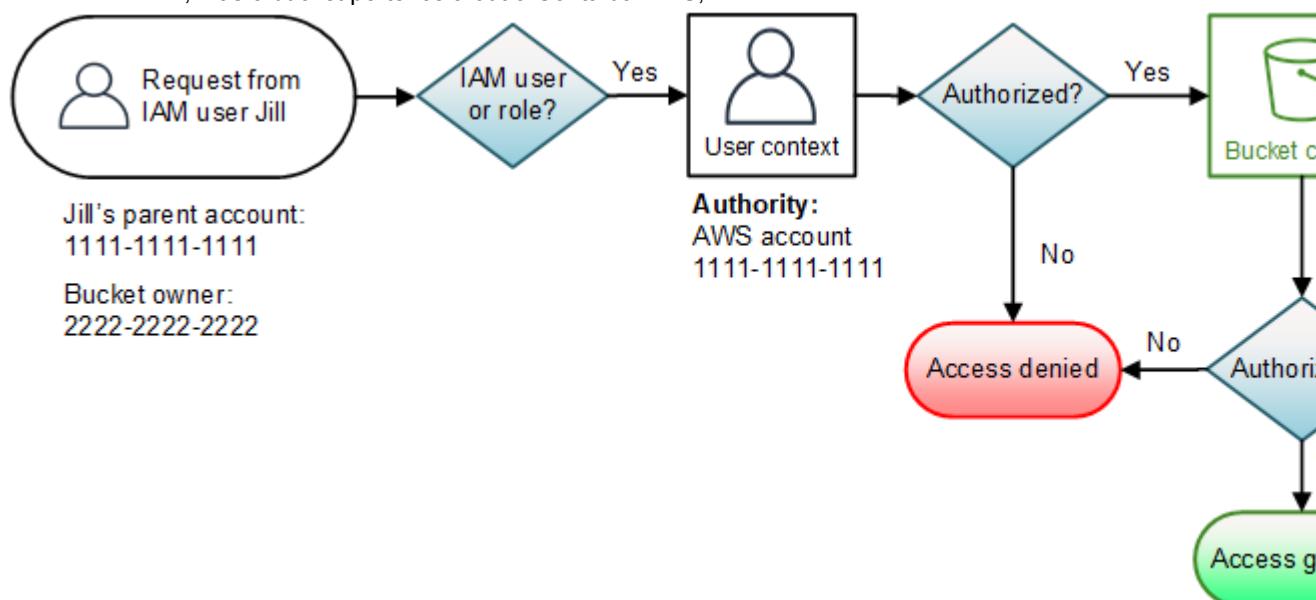
1. Como a solicitação é de uma entidade principal do IAM, no contexto de usuário, o Amazon S3 avalia todas as políticas que pertencem à Conta da AWS pai para determinar se Jill tem permissão para executar a operação.

Neste exemplo, a Conta da AWS pai 1111-1111-1111 à qual a entidade principal 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.

- 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 entidade principal do IAM cuja Conta da AWS pai não é a proprietária do bucket

Neste exemplo, a solicitação é enviada por Jill, uma usuária do IAM cuja Conta da AWS pai é 1111-1111-1111, mas o bucket pertence a outra Conta da AWS, 2222-2222-2222.



Jill precisará de permissões da Conta da AWS pai e do proprietário do bucket. O Amazon S3 avalia o contexto da seguinte forma:

- Como a solicitação é de um principal 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.
- No contexto de bucket, o Amazon S3 verifica se o proprietário do bucket 2222-2222-2222 concedeu a Jill (ou à Conta da AWS pai) 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

Ao receber uma solicitação para uma operação de objeto, o Amazon S3 converterá todas as permissões relevantes, permissões 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 do usuário: se o solicitante for um principal do IAM, o principal deverá ter permissão da Conta da AWS pai à qual ele 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 pai anexa ao principal. 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 da AWS pai for a proprietária do recurso (bucket ou objeto), ela poderá conceder permissões de recurso à entidade principal do IAM usando a política de usuário ou a política de recurso.

2. Contexto do 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 é proprietária do objeto na solicitação não for a mesma que é proprietária do bucket, no contexto de 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.

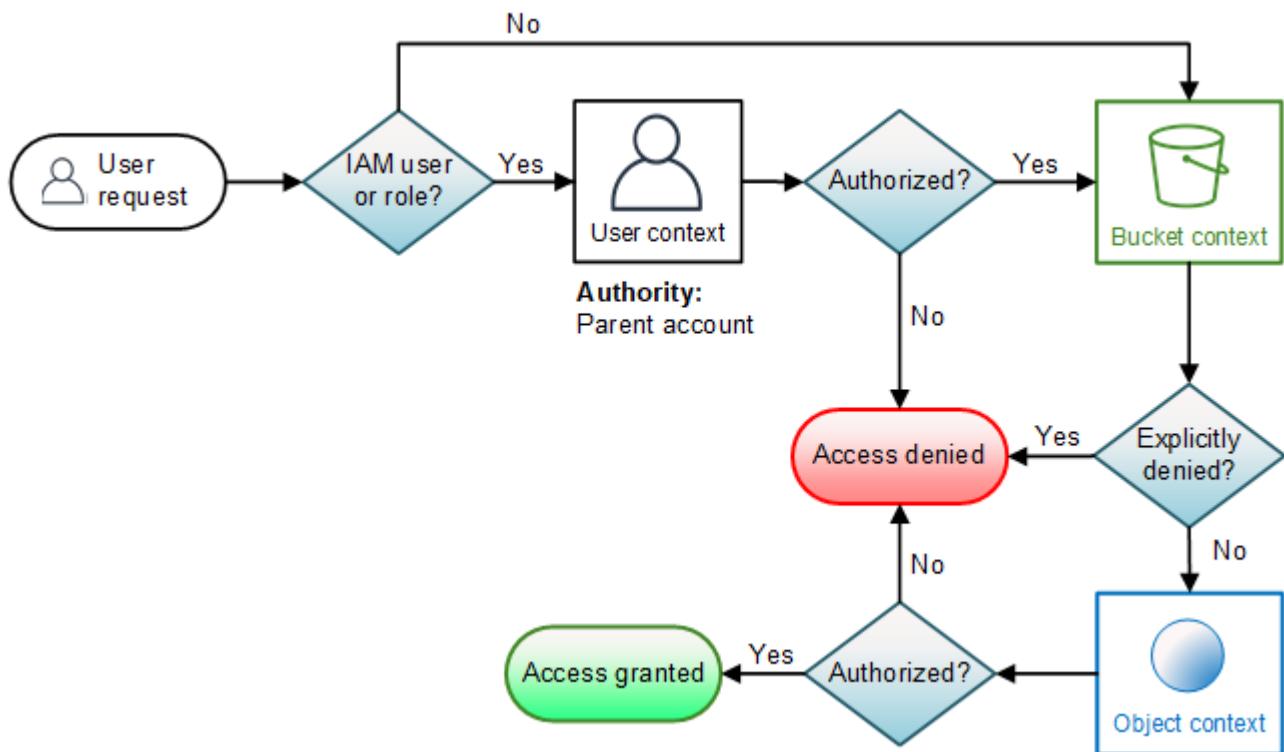
3. Contexto de objeto: o solicitante deve ter permissões do proprietário do objeto para executar uma operação específica de objeto. Nesta etapa, o Amazon S3 avalia a ACL do objeto.

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 é proprietária do objeto também for a conta pai à qual a entidade principal do IAM pertence, ela poderá configurar permissões de objeto em uma política 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 políticas de acesso \(p. 425\)](#).

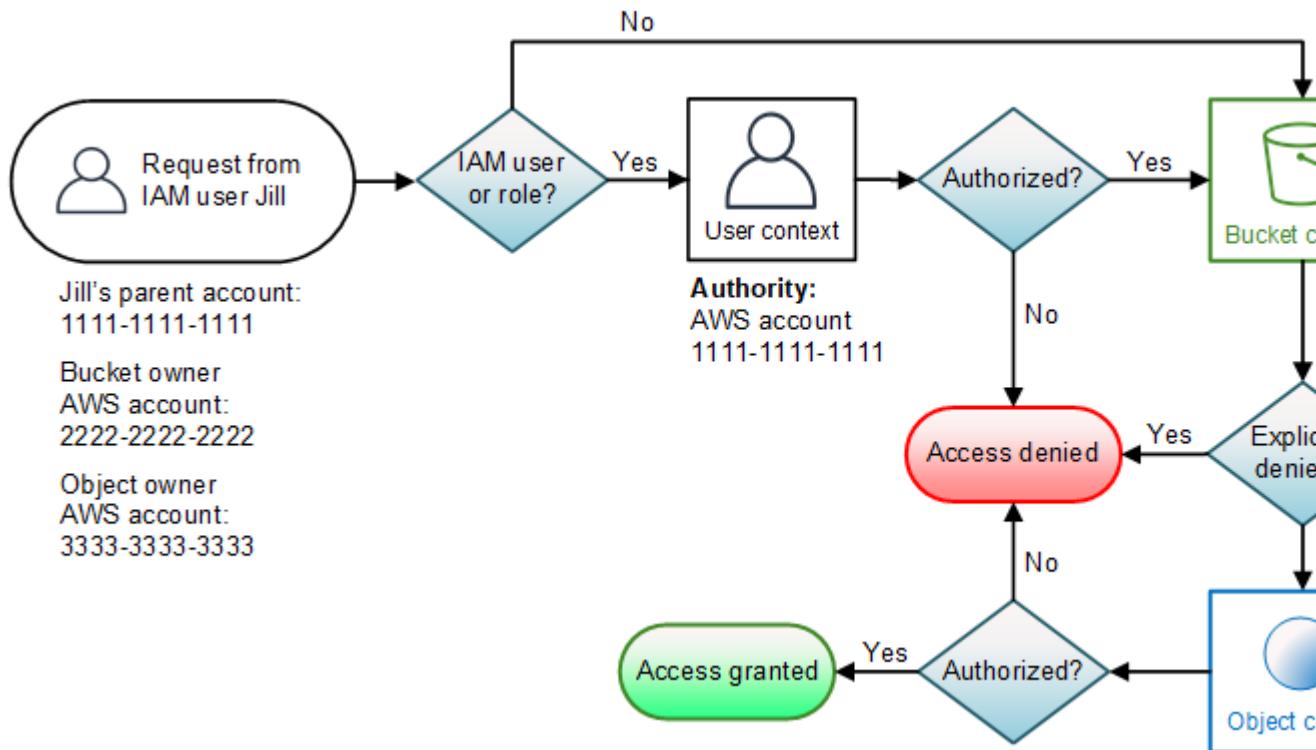
Como proprietário do bucket, se você quiser ter propriedade de todos os objetos de seu bucket e usar políticas de bucket ou políticas baseadas em IAM para gerenciar o acesso a esses objetos, aplique a configuração aplicada pelo proprietário do bucket em Object Ownership. Com essa configuração, você, como proprietário do bucket, automaticamente tem a propriedade e o controle total sobre todos os objetos de seu bucket. As ACLs de bucket e objeto não poderão ser editadas e não serão mais consideradas para acesso. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

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 da AWS pai é 1111-1111-1111, envia uma solicitação de operação de objeto (por exemplo, "Get object") a um objeto pertencente à Conta da AWS 3333-3333-3333 em um bucket pertencente à Conta da AWS 2222-2222-2222.



Jill precisará de permissões da Conta da AWS pai, 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 entidade principal do IAM, o Amazon S3 avalia o contexto de usuário para verificar se a Conta da AWS pai 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 do bucket, o proprietário do bucket, a 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 do 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.

Políticas de bucket e políticas de usuário

Políticas de bucket e políticas 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. 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 [Identity and Access Management no Amazon S3 \(p. 419\)](#).

Important

As políticas de bucket são limitadas a 20 KB.

Tópicos

- [Políticas e permissões no Amazon S3 \(p. 437\)](#)
- [Uso de políticas de bucket \(p. 515\)](#)
- [Uso de políticas de usuário do IAM \(p. 528\)](#)
- [Demonstrações de exemplo: gerenciar o acesso aos recursos do Amazon S3 \(p. 553\)](#)
- [Usar funções vinculadas a serviços para o Amazon S3 Storage Lens \(p. 582\)](#)

Políticas e permissões no Amazon S3

Esta página fornece uma visão geral de bucket e políticas de usuário no Amazon S3 e descreve os elementos básicos de uma política. Cada elemento listado vincula mais detalhes sobre esse elemento e exemplos de como usá-lo.

Para obter uma lista completa de ações, recursos e condições do Amazon S3, consulte [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#).

No sentido mais básico, uma política contém os seguintes elementos:

- **Recursos (p. 438)**: buckets, objetos, pontos de acesso e trabalhos 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. Para obter mais informações, consulte [Recursos do Amazon S3 \(p. 438\)](#).
- **Ações (p. 441)**: para cada recurso, o Amazon S3 oferece suporte a um conjunto de operações. Você identifica as operações de recursos que permitirão (ou negarão) usando palavras-chave de ação.

Por exemplo, a permissão s3 : ListBucket autoriza o usuário a empregar a operação [GET Bucket \(List Objects\)](#) do Amazon S3. Para obter mais informações sobre como usar ações do Amazon S3, consulte [Ações do Amazon S3 \(p. 441\)](#). Para obter uma lista completa das ações do Amazon S3, consulte [Ações](#).

- **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 acesso explicitamente a um recurso. Você poderia fazer isso para garantir que um usuário não possa acessar o recurso, mesmo se uma política diferente conceder acesso. Para obter mais informações, consulte [Elementos da política JSON do IAM: efeito](#).

- **Principal (p. 440)** — A conta ou usuário que tem permissão de acesso a ações e recursos na instrução. Em uma política de bucket, o principal é o usuário, a conta, o serviço ou outra entidade que receba essa permissão. Para obter mais informações, consulte [Principais \(p. 440\)](#).

- [Condição \(p. 447\)](#): condições para quando uma política está em vigor. Você pode usar chaves de toda a AWS e chaves específicas do Amazon S3 para especificar condições em uma política de acesso do Amazon S3. Para obter mais informações, consulte [Exemplos de chave de condição do Amazon S3 \(p. 447\)](#).

A política de bucket de exemplo a seguir mostra os elementos efeito, principal, ação e recurso. 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 awsexamplebucket1.

```
{  
    "Version": "2012-10-17",  
    "Id": "ExamplePolicy01",  
    "Statement": [  
        {  
            "Sid": "ExampleStatement01",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::123456789012:user/Dave"  
            },  
            "Action": [  
                "s3:GetObject",  
                "s3:GetBucketLocation",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::awsexamplebucket1/*",  
                "arn:aws:s3:::awsexamplebucket1"  
            ]  
        }  
    ]  
}
```

Para obter mais informações, consulte os tópicos abaixo. Para obter informações completas sobre linguagem de políticas, consulte [Políticas e permissões](#) e [Referência de política de JSON do IAM](#) no Guia do usuário do IAM.

Tópicos

- [Recursos do Amazon S3 \(p. 438\)](#)
- [Principais \(p. 440\)](#)
- [Ações do Amazon S3 \(p. 441\)](#)
- [Chaves de condição do Amazon S3 \(p. 444\)](#)
- [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#)

Recursos do Amazon S3

O formato comum de nome do recurso da Amazon (ARN) a seguir identifica recursos na AWS:

```
arn:partition:service:region:namespace:relative-id
```

Para obter informações sobre os ARNs, consulte [Amazon Resource Names \(ARNs\)](#) (Nomes de recurso da Amazon (ARNs)) na Referência geral da AWS.

Para obter informações sobre recursos, consulte [Elementos de política de JSON do IAM: recurso](#) no Guia do usuário do IAM.

Um ARN do Amazon S3 exclui a Região da AWS e o namespace, mas inclui o seguinte:

- Partição: aws é um nome de partição comum. Se os seus recursos estiverem na região China (Pequim), aws-cn será o nome da partição.
- Serviço - s3.
- ID relativo: bucket-name ou um bucket-name/object-key. Você pode usar curingas.

O formato do ARN para recursos do Amazon S3 se reduz a:

```
arn:aws:s3:::bucket_name/key_name
```

Para obter uma lista completa dos recursos do Amazon S3, consulte [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#).

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 os tópicos a seguir:

- [Adicionar uma política de bucket usando o console do Amazon S3 \(p. 516\)](#)
- [Configuração de CORS \(p. 608\)](#)

Exemplos de ARN do Amazon S3

Veja a seguir exemplos de ARNs de recurso do Amazon S3.

Nome do bucket e chave de objeto especificados

O ARN a seguir identifica o objeto /developers/design_info.doc no bucket examplebucket.

```
arn:aws:s3:::examplebucket/developers/design_info.doc
```

Curingas

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.

- O ARN a seguir usa o curinga * na parte do ARN relativa ao ID para identificar todos os objetos no bucket examplebucket.

```
arn:aws:s3:::examplebucket/*
```

- O ARN a seguir usa * para indicar todos os recursos do Amazon S3 (todos os buckets e objetos do S3).

```
arn:aws:s3::::*
```

- O ARN a seguir usa curingas, * e ?, na parte de relative-ID. Ele identifica todos os objetos em buckets, como example1bucket, example2bucket, example3bucket e assim por diante.

```
arn:aws:s3:::example?bucket/*
```

Variáveis de políticas

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}/
```

No 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.

Principais

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.

Conceder permissões a uma Conta da AWS

Para conceder permissões a uma Conta da AWS, identifique a conta usando o formato a seguir.

```
"AWS" : "account-ARN"
```

Veja os exemplos a seguir.

```
"Principal": {"AWS": "arn:aws:iam::AccountIDWithoutHyphens:root"}
```

```
"Principal": {"AWS": ["arn:aws:iam::AccountID1WithoutHyphens:root", "arn:aws:iam::AccountID2WithoutHyphens:root"]}
```

O Amazon S3 também é compatível com um ID de usuário canônico, que é uma forma complexa de ID de Conta da AWS. Você pode especificar esse ID usando o formato a seguir.

```
"CanonicalUser": "64-digit-alphanumeric-value"
```

Veja um exemplo a seguir.

```
"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.

Conceder permissões a um usuário do IAM

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 obter exemplos detalhados que fornecem instruções passo a passo, consulte [Exemplo 1: Proprietário do bucket que concede permissões do bucket aos usuários \(p. 557\)](#) e [Exemplo 3: O proprietário do bucket concede permissões para objetos que não possui \(p. 568\)](#).

Conceder permissões anônimas

Para conceder permissão a todos, também denominada acesso anônimo, defina o curinga ("*") como o valor Principal. Por exemplo, se você configura seu bucket como um site, quer que todos os objetos no bucket sejam publicamente acessíveis.

"Principal": "*"

"Principal": { "AWS": "*" }

Usar "Principal": "*" com um efeito Allow em uma política baseada em recursos permite que qualquer pessoa, mesmo que não esteja conectada à AWS, acesse seu recurso.

Usar `"Principal" : { "AWS" : "*" }` com um efeito Allow em uma política baseada em recursos permite que qualquer usuário raiz, usuário do IAM, sessão de função assumida ou usuário federado em qualquer conta na mesma partição acesse seu recurso. Para obter mais informações, consulte [Todas as entidades principais](#) no Guia do usuário do IAM.

Não é possível usar um curinga para fazer a correspondência de parte de um nome de entidade principal ou ARN.

Important

Como qualquer pessoa pode criar uma Conta da AWS, o nível de segurança desses dois métodos é equivalente, mesmo que funcionem de maneira diferente.

Warning

Tenha cuidado ao conceder acesso anônimo ao bucket do Amazon 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.

Exigir acesso por URLs do CloudFront

Você pode exigir que os usuários acessem seu 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. Em seguida, altere as permissões no bucket ou nos objetos em seu bucket. O formato para especificar a OAI em uma instrução Principal é o descrito a seguir.

```
"Principal": {"CanonicalUser": "Amazon S3 Canonical User ID assigned to origin access identity"}
```

Acções do Amazon S3

O Amazon S3 define um conjunto de permissões que você pode especificar em uma política. Estas são palavras-chave e cada uma mapeia para uma operação específica do Amazon S3. Para obter mais informações sobre as operações do Amazon S3, consulte [Ações](#) na Referência da API do Amazon Simple Storage Service.

Para ver como especificar permissões em uma política do Amazon S3, revise os exemplos de políticas a seguir. Para obter uma lista de ações, recursos e chaves de condição do Amazon S3 para uso em políticas, consulte [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#). Para obter uma lista completa das ações do Amazon S3, consulte [Ações](#).

Tópicos

- Exemplo: operações de objeto (p. 442)
- Exemplo: operações de bucket (p. 442)
- Exemplo: operações de sub-recursos de bucket (p. 443)
- Exemplo: operações de conta (p. 444)

Exemplo: operações 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. Estas são operações de objetos. Assim, a parte de `relative-id` do ARN do `Resource` identifica objetos (`awsexamplebucket1/*`). Para obter mais informações, consulte [Recursos do Amazon S3 \(p. 438\)](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "statement1",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::12345678901:user/Dave"  
            },  
            "Action": [  
                "s3:PutObject",  
                "s3:PutObjectAcl"  
            ],  
            "Resource": "arn:aws:s3:::awsexamplebucket1/*"  
        }  
    ]  
}
```

Permissões para todas as ações do Amazon S3

Você pode usar um curinga para conceder permissão para todas as ações do Amazon S3.

```
"Action": "*"
```

Exemplo: operações de bucket

O exemplo a seguir de política de usuário concede as permissões `s3:CreateBucket`, `s3>ListAllMyBuckets` e `s3:GetBucketLocation` a um usuário. Para todas essas permissões, defina a parte de `Resource` ARN de `relative-id` como `"*"`. Para todas as outras ações de bucket, você deve especificar um nome de bucket. Para obter mais informações, consulte [Recursos do Amazon S3 \(p. 438\)](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "statement1",  
            "Effect": "Allow",  
            "Action": [  
                "s3:CreateBucket",  
                "s3>ListAllMyBuckets",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": [  
                "arn:aws:s3:::/*"  
            ]  
        }  
    ]  
}
```

}

Política de acesso ao console

Se um usuário quiser usar o AWS Management Console para visualizar buckets e o conteúdo de qualquer um desses buckets, o usuário deverá ter as permissões `s3>ListAllMyBuckets` e `s3:GetBucketLocation`. Para obter um exemplo, consulte Política de acesso ao console no post de blog [Escrever políticas do IAM: como conceder acesso a um bucket do S3](#).

Exemplo: operações de sub-recursos de bucket

O seguinte exemplo de política de usuário concede a permissão `s3:GetBucketAcl` no bucket `DOC-EXAMPLE-BUCKET1` ao usuário Dave.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "statement1",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::123456789012:user/Dave"  
            },  
            "Action": [  
                "s3:GetBucketAcl"  
            ],  
            "Resource": [  
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"  
            ]  
        }  
    ]  
}
```

Permissões de DELETE Object

Você pode excluir objetos chamando explicitamente a API `DELETE` objeto ou configurando seu ciclo de vida (consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#)) 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`.

Negação explícita

Por padrão, os usuários não têm nenhuma permissão. Mas, à medida que você cria usuários, adiciona usuários a grupos e concede permissões a eles, eles podem obter certas permissões que você não pretendia conceder. Para evitar essas brechas de permissão, você pode elaborar uma política de acesso mais estrita adicionando uma negação explícita.

A política de bucket anterior concede a permissão `s3:GetBucketAcl` do bucket `DOC-EXAMPLE-BUCKET1` ao usuário Dave. Neste exemplo, você nega explicitamente ao usuário Dave permissões `DELETE` do objeto. A negação explícita sempre se sobrepõe a qualquer outra permissão concedida. Veja a seguir um exemplo de política de acesso revisada com a negação explícita adicionada.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "statement1",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::123456789012:user/Dave"  
            },  
            "Action": [  
                "s3:GetBucketAcl"  
            ],  
            "Resource": [  
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"  
            ]  
        },  
        {  
            "Sid": "statement2",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": ["s3:DeleteObject"],  
            "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*"]  
        }  
    ]  
}
```

```
    "Action": [
        "s3:GetObjectVersion",
        "s3:GetBucketAcl"
    ],
    "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET1",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*"
    ]
},
{
    "Sid": "statement2",
    "Effect": "Deny",
    "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Dave"
    },
    "Action": [
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:PutLifecycleConfiguration"
    ],
    "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET1",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*"
    ]
}
]
```

Exemplo: operações de conta

O exemplo a seguir de política de usuário concede a permissão `s3:GetAccountPublicAccessBlock` a um usuário. Para todas essas permissões, defina o valor `Resource` como `"*"`. Para obter mais informações, consulte [Recursos do Amazon S3 \(p. 438\)](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Action": [
                "s3:GetAccountPublicAccessBlock"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Chaves de condição do Amazon S3

A linguagem de políticas de acesso permite que especificar condições ao conceder permissões. Para especificar condições para quando uma política está em vigor, você pode usar o elemento opcional `Condition` ou o bloco `Condition`, para especificar condições para quando uma política está em vigor. Você pode usar chaves predefinidas de toda a AWS e chaves específicas do Amazon S3 para especificar condições em uma política de acesso do Amazon S3.

No elemento `Condition`, que é opcional, você cria expressões em que usa operadores boolianos (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:::awsexamplebucket1/*"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "s3:x-amz-acl": "public-read"  
                }  
            }  
        }  
    ]  
}
```

No exemplo, 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 pode ser usado 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 cada solicitação de objeto PUT.

Tópicos

- [Chaves de condição de toda a AWS \(p. 445\)](#)
- [Chaves de condição específicas do Amazon S3 \(p. 446\)](#)
- [Exemplos de chave de condição do Amazon S3 \(p. 447\)](#)

Chaves de condição 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 são chamadas chaves de toda a AWS e usam o prefixo `aws:`. Para obter uma lista completa de chaves de condição em toda a AWS, consulte [Chaves da AWS disponíveis para condições](#) no Guia do usuário do IAM. Você pode usar chaves de condição de toda a AWS no Amazon S3. 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 (192.0.2.0.*), a menos que o endereço IP seja 192.0.2.188. 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 de toda a 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, consulte <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:::awsexamplebucket1/*",  
            "Condition": {  
                "IpAddress": "192.0.2.188"  
            }  
        }  
    ]  
}
```

```
        "IpAddress" : {
            "aws:SourceIp": "192.0.2.0/24"
        },
        "NotIpAddress" : {
            "aws:SourceIp": "192.0.2.188/32"
        }
    }
]
```

Você também pode usar outras chaves de condição de toda a AWS nas políticas do Amazon S3. Por exemplo, você pode especificar o `aws:SourceVpc` e as chaves de condição `aws:SourceVpc` em políticas de bucket para VPC endpoints. Para obter exemplos específicos, consulte [Controlar o acesso a partir de VPC endpoints com políticas de bucket \(p. 517\)](#).

Note

Em relação a algumas chaves de condição globais da AWS, somente certos tipos de recurso são compatíveis. Portanto, confira se o Amazon S3 é compatível com a chave de condição global e o tipo de recurso que você deseja usar ou se você precisará usar uma chave de condição específica do Amazon S3. Para obter uma lista completa de tipos de recurso e chaves de condição compatíveis com o Amazon S3, consulte [Ações, recursos e condições](#).

Chaves de condição específicas do Amazon S3

Você pode usar chaves de condição do Amazon S3 com ações específicas do Amazon S3. Cada chave de condição mapeia para o mesmo cabeçalho de solicitação de nome permitido pela API na qual a condição pode ser definida. As chaves de condição específicas do Amazon S3 ditam o comportamento dos cabeçalhos de solicitação de mesmo nome. Para obter uma lista completa de chaves de condição específicas do Amazon S3, consulte [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#).

Por exemplo, a chave de condição `s3:x-amz-acl` usada 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 do objeto PUT 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.

A política de bucket a seguir concederá a permissão `s3:PutObject` a duas Contas da AWS se a solicitação incluir o cabeçalho `x-amz-acl`, tornando o objeto publicamente legível. 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:`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AddCannedAcl",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::Account1-ID:root",
                    "arn:aws:iam::Account2-ID:root"
                ]
            },
            "Action": "s3:PutObject",
```

```
    "Resource": [ "arn:aws:s3:::awsexamplebucket1/*" ],
    "Condition": {
        "StringEquals": {
            "s3:x-amz-acl": ["public-read"]
        }
    }
}
```

Important

Nem todas as condições fazem sentido para todas as ações. Por exemplo, faz sentido incluir uma condição `s3:LocationConstraint` em uma política que concede a permissão `s3:CreateBucket` do Amazon S3. No entanto, não faz sentido incluir essa condição em uma política que conceda a permissão `s3:GetObject`. O Amazon S3 pode testar erros de semântica para esse tipo que envolve 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.

Exemplos de chave de condição do Amazon S3

Você pode usar a linguagem de políticas de acesso para especificar condições ao conceder permissões. Você pode usar o elemento `Condition` opcional ou o bloco `Condition` para especificar condições para quando uma política está em vigor.

Para obter políticas que usam chaves de condição do Amazon S3 para operações de objeto e bucket, consulte os exemplos a seguir. Para obter mais informações sobre essas chaves de condição, consulte [Chaves de condição do Amazon S3 \(p. 444\)](#). Para obter uma lista completa de ações do Amazon S3, as chaves de condição e os recursos que você pode especificar em políticas, consulte [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#).

Exemplos: chaves de condição do Amazon S3 para operações de objeto

Esta seção fornece exemplos que mostram como você pode usar chaves de condição específicas do Amazon S3 para operações de objeto. Para obter uma lista completa de ações do Amazon S3, as chaves de condição e os recursos que você pode especificar em políticas, consulte [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#).

Várias das políticas de exemplo mostram como você pode usar chaves de condições com operações de [objeto PUT](#). As operações de objeto PUT permitem cabeçalhos específicos baseados em lista de controle de acesso (ACL) usados 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. Você também pode conceder permissões baseadas em ACL com a operação `PutObjectAcl`. Para obter mais informações, consulte [PutObjectAcl](#) na Referência da API do Amazon S3 Amazon Simple Storage Service. Para obter mais informações sobre ACLs, consulte [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#).

Tópicos

- [Exemplo 1: Concessão da permissão s3:PutObject com uma condição que exige que o proprietário do bucket obtenha controle total \(p. 448\)](#)
- [Exemplo 2: Concessão da permissão s3:PutObject que exige os objetos armazenados usando criptografia no lado do servidor \(p. 450\)](#)
- [Exemplo 3: Concessão da permissão s3:PutObject para copiar objetos com uma restrição na origem da cópia \(p. 450\)](#)
- [Exemplo 4: Concessão de acesso a uma versão específica de um objeto \(p. 451\)](#)
- [Exemplo 5: restrição de uploads de objetos com uma classe de armazenamento específica \(p. 452\)](#)

- Exemplo 6: conceder permissões com base em tags de objetos (p. 453)
- Exemplo 7: restringir o acesso pelo ID da Conta da AWS do proprietário do bucket (p. 453)
- Exemplo 8: exigir uma versão mínima do TLS (p. 453)

Exemplo 1: Concessão da permissão s3:PutObject com uma condição que exige que o proprietário do bucket obtenha controle total

A operação [PUT Object](#) permite cabeçalhos específicos da lista de controle de acesso (ACL) 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.

Suponha 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. Para obter mais informações, consulte [Objeto PUT](#).

[Exigir o cabeçalho x-amz-full-control](#)

Você pode exigir 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:::awsexamplebucket1/*",  
            "Condition": {  
                "StringEquals": {  
                    "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"  
                }  
            }  
        }  
    ]  
}
```

Note

Este exemplo trata da permissão entre contas. Contudo, se Dave (a quem é concedida a permissão) pertencer à Conta da AWS que é proprietária do bucket, essa permissão condicional não será necessária. Isso acontece porque a conta pai à qual o Dave pertence é proprietária dos objetos que o usuário carrega.

Adicionar negação explícita

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 com a negação explícita adicionada.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::awsexamplebucket1/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"
                }
            }
        },
        {
            "Sid": "statement2",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::awsexamplebucket1/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"
                }
            }
        }
    ]
}
```

Testar a política com a AWS CLI

Se você tiver duas Contas da AWS, teste a política usando a AWS Command Line Interface (AWS CLI). Anexe a política e use as credenciais de Dave para testar a permissão usando o seguinte comando `put-object` da AWS CLI. 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 [Desenvolvimento com o Amazon S3 usando a AWS CLI \(p. 1257\)](#).

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg --grant-full-control id="AccountA-CanonicalUserID" --profile AccountBUserProfile
```

Exigir o cabeçalho x-amz-acl

Você pode exigir 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 o exemplo abaixo.

```
"Condition": {
    "StringEquals": {
```

```
        "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 AccountAdmin
```

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 example1bucket --key HappyFace.jpg --body c:\HappyFace.jpg --server-side-encryption "AES256" --profile AccountAdmin
```

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 copiar objetos do bucket de origem e somente os objetos cujo prefixo de nome de chave começa com `public/` (por exemplo, `sourcebucket/public/*`).
- Permite copiar apenas um objeto específico do bucket de origem (por exemplo, `sourcebucket/example.jpg`).

A política de bucket a seguir concede a permissão s3:PutObject ao usuário (Dave). 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 `/awsexamplebucket1/public/*`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "cross-account permission to user in your own account",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::awsexamplebucket1/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-copy-source": "/awsexamplebucket1/public/*"
                }
            }
        }
    ]
}
```

```
        "Action": "s3:PutObject",
        "Resource": "arn:aws:s3:::awsexamplebucket1/*"
    },
    {
        "Sid": "Deny your user permission to upload object if copy source is not / bucket/folder",
        "Effect": "Deny",
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/Dave"
        },
        "Action": "s3:PutObject",
        "Resource": "arn:aws:s3:::awsexamplebucket1/*",
        "Condition": {
            "StringNotLike": {
                "s3:x-amz-copy-source": "awsexamplebucket1/public/*"
            }
        }
    }
]
```

Testar a política com a AWS CLI

Você pode testar a permissão usando o comando da AWS CLI `copy-object`. Você especifica a origem adicionando o parâmetro `--copy-source`; o prefixo de nome de chave que deve coincidir 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 [Desenvolvimento com o Amazon S3 usando a AWS CLI \(p. 1257\)](#).

```
aws s3api copy-object --bucket awsexamplebucket1 --key HappyFace.jpg
--copy-source examplebucket/public/PublicHappyFace1.jpg --profile AccountADave
```

Conceder permissão para copiar somente um objeto específico

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": "awsexamplebucket1/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 deseja conceder agora 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 mostrado abaixo. O par de chave-valor no bloco `Condition` especifica a chave de condição `s3:VersionId`. Neste caso, Dave precisa saber o ID de versão do objeto exato para recuperar o objeto.

Para obter mais informações, consulte [GetObject](#) na Referência da API do Amazon Simple Storage Service.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
```

```
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/Dave"
        },
        "Action": "s3:GetObjectVersion",
        "Resource": "arn:aws:s3:::examplebucketversionenabled/HappyFace.jpg"
    },
    {
        "Sid": "statement2",
        "Effect": "Deny",
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/Dave"
        },
        "Action": "s3:GetObjectVersion",
        "Resource": "arn:aws:s3:::examplebucketversionenabled/HappyFace.jpg",
        "Condition": {
            "StringNotEquals": {
                "s3:VersionId": "AaaHbAQitwiL_h47_44lR02DDfLlB05e"
            }
        }
    }
]
```

Testar a política com a AWS CLI

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_44lR02DDfLlB05e --profile AccountADave
```

Exemplo 5: restrição de uploads de objetos com uma classe de armazenamento específica

Suponha que a Conta A, representada pelo ID da conta 123456789012, possui 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`. Para restringir uploads de objetos a uma classe de armazenamento específica, o administrador da Conta A pode usar a chave de condição `s3:x-amz-storage-class`, conforme mostrado no exemplo de política de bucket a seguir.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-storage-class": [
                        "STANDARD_IA"
                    ]
                }
            }
        }
    ]
}
```

Exemplo 6: conceder permissões com base em tags de objetos

Para obter exemplos sobre como usar chaves de condição de tags de objetos com operações do Amazon S3, consulte [Marcação e políticas de controle de acesso \(p. 882\)](#).

Exemplo 7: restringir o acesso pelo ID da Conta da AWS do proprietário do bucket

É possível usar a chave `aws:ResourceAccount` ou `s3:ResourceAccount` para criar políticas do IAM ou do endpoint da nuvem privada virtual que restrinjam o acesso de usuários ou aplicações aos buckets do Amazon S3 pertencentes a um ID de Conta da AWS específica. Você pode usar essa chave de condição para restringir o acesso dos clientes dentro de sua VPC a buckets que você não possui.

No entanto, saiba que os serviços da AWS dependem do acesso a buckets gerenciados pela AWS. Portanto, usar a chave `aws:ResourceAccount` ou `s3:ResourceAccount` em sua política do IAM também pode afetar o acesso a esses recursos.

Para obter mais informações e exemplos, consulte os seguintes recursos:

- [Restringir o acesso a buckets em uma Conta da AWS especificada](#) no Guia do usuário do AWS PrivateLink
- [Restringir o acesso aos buckets que o Amazon ECR usa](#) no Guia do Amazon ECR
- [Fornecer acesso necessário ao Systems Manager para buckets do Amazon S3 gerenciados pela AWS](#) no Guia do AWS Systems Manager
- [Limit access to Amazon S3 buckets owned by specific Contas da AWS](#) (Limitar o acesso a buckets do Amazon S3 pertencentes a Contas da AWS específicas) no AWS Storage Blog

É possível usar a chave de condição `aws:ResourceAccount` ou `s3:ResourceAccount` para criar políticas do IAM ou do endpoint da nuvem privada virtual que restrinjam o acesso de usuários ou aplicações aos buckets do S3 pertencentes a um ID de uma Conta da AWS específica. Você pode usar essa chave de condição para restringir o acesso dos clientes dentro de sua VPC a buckets que você não possui.

Para obter informações e exemplos, consulte os seguintes recursos:

- [Restrição de acesso a buckets em uma Conta da AWS especificada](#) no Guia do usuário do AWS PrivateLink
- [Limitar o acesso a AWS pertencentes a Contas da AWS específicas](#) no Blog de armazenamento da AWS

Exemplo 8: exigir uma versão mínima do TLS

Você pode usar a chave de condição `s3:TlsVersion` para gravar IAM, Virtual Private Cloud Endpoint (VPCE) ou políticas de bucket que restringem o acesso do usuário ou aplicação aos buckets do Amazon S3 com base na versão TLS usada pelo cliente. Você pode usar essa chave de condição para gravar políticas que exigem uma versão mínima do TLS.

Example

Esse exemplo de política de bucket nega solicitações PutObject de clientes que tenham uma versão do TLS inferior a 1.2, por exemplo, 1.1 ou 1.0.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:PutObject",  
            "Condition": {  
                "s3:TlsVersion": "  
                    < 1.2  
                "}  
            }  
        }  
    ]  
}
```

```
"Resource": [
    "arn:aws:s3::::DOC-EXAMPLE-BUCKET1",
    "arn:aws:s3::::DOC-EXAMPLE-BUCKET1/*"
],
"Condition": {
    "NumericLessThan": {
        "s3:TlsVersion": 1.2
    }
}
]
```

Example

Essa política de bucket de exemplo permite solicitações PutObject de clientes que tenham uma versão do TLS superior a 1.1, por exemplo, 1.2, 1.3 ou acima.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3::::DOC-EXAMPLE-BUCKET1",
                "arn:aws:s3::::DOC-EXAMPLE-BUCKET1/*"
            ],
            "Condition": {
                "NumericGreaterThan": {
                    "s3:TlsVersion": 1.1
                }
            }
        }
    ]
}
```

Exemplos: chaves de condição do Amazon S3 para operações de bucket

Esta seção fornece exemplos de políticas que mostram como você pode usar chaves de condição específicas do Amazon S3 para operações de bucket.

Tópicos

- [Exemplo 1: conceder a um usuário permissão para criar um bucket somente em uma região específica \(p. 454\)](#)
- [Exemplo 2: obter uma lista de objetos em um bucket com um prefixo específico \(p. 456\)](#)
- [Exemplo 3: definir o número máximo de chaves \(p. 458\)](#)

[Exemplo 1: conceder a um usuário permissão para criar um bucket somente em uma região específica](#)

Suponha 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 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 de regiões do Amazon S3, consulte [Regiões e endpoints](#) na Referência geral da AWS.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "statement1",  
            "Effect": "Allow",  
            "Action": "s3:CreateBucket",  
            "Resource": "arn:aws:s3:::*",  
            "Condition": {  
                "StringLike": {  
                    "s3:LocationConstraint": "sa-east-1"  
                }  
            }  
        }  
    ]  
}
```

Adicionar negação explícita

A política acima impede que o usuário crie um bucket em qualquer outra Região, exceto sa-east-1. No entanto, alguma outra política pode conceder a esse usuário permissão para criar buckets em outra Região. Por exemplo, se ele pertencer a um grupo, o grupo pode ter uma política anexada que permita que todos os seus usuários tenham permissão para criar buckets em outra Região. Para garantir que o usuário não obtenha permissão para criar buckets em nenhuma outra Região, você pode adicionar uma instrução de negação explícita nessa política.

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. A política abaixo inclui uma instrução de negação explícita.

```
{  
    "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"  
                }  
            }  
        }  
    ]  
}
```

```
        }
    ]
}
```

Testar a política com a AWS CLI

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 [Desenvolvimento com o Amazon S3 usando a AWS CLI \(p. 1257\)](#).

```
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: obter uma lista de objetos em um bucket com um prefixo específico

Você pode usar a chave de condição `s3:prefix` para limitar a resposta da API [GET Bucket \(ListObjects\)](#) a nomes de chave com um prefixo específico. Se você for o proprietário do bucket, poderá restringir um usuário a listar o conteúdo de um prefixo específico no bucket. Essa chave de condição será útil se os objetos no bucket forem organizados por prefixes de nome de chave. O console do Amazon S3 usa prefixes de nomes de chaves para mostrar um conceito de pasta. Somente o console suporta o conceito de pastas; a API do Amazon S3 suporta somente buckets e objetos. Para obter mais informações sobre como usar prefixes e delimitadores para filtrar permissões de acesso, consulte [Controlar o acesso a um bucket com políticas de usuário \(p. 528\)](#).

Por exemplo, se você tiver dois objetos com nomes de chave `public/object1.jpg` e `public/object2.jpg`, o console mostrará os objetos na pasta `public`. Na API do Amazon S3, esses são objetos com prefixes, não objetos em pastas. No entanto, na API do Amazon S3, se você organizar suas chaves de objeto usando tais prefixes, poderá conceder a permissão `s3>ListBucket` com a condição `s3:prefix` que permitirá que o usuário obtenha uma lista de nomes de chave com um prefixo específico.

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. Para obter mais informações sobre outras chaves de condição que você pode usar com a API `GET Bucket (ListObjects)`, consulte [ListObjects](#).

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:::awsexamplebucket1",
            "Condition": {
                "StringEquals": {
                    "s3:prefix": "projects"
                }
            }
        }
    ]
}
```

```
        }
    },
{
    "Sid":"statement2",
    "Effect":"Deny",
    "Action": "s3>ListBucket",
    "Resource": "arn:aws:s3:::awsexamplebucket1",
    "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 objetos sem nenhuma restrição, tanto por meio de atualizações na política de usuário anterior quanto por meio de uma política de bucket. Como a negação explícita sempre prevalece, a solicitação do usuário para listar outras chaves além do prefixo `projects` é negada.

Política de bucket

Se você adicionar o elemento `Principal` à política de usuário acima, 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::123456789012:user/bucket-owner"
            },
            "Action": "s3>ListBucket",
            "Resource": "arn:aws:s3:::awsexamplebucket1",
            "Condition" : {
                "StringEquals" : {
                    "s3:prefix": "projects"
                }
            }
        },
        {
            "Sid":"statement2",
            "Effect":"Deny",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/bucket-owner"
            },
            "Action": "s3>ListBucket",
            "Resource": "arn:aws:s3:::awsexamplebucket1",
            "Condition" : {
                "StringNotEquals" : {
                    "s3:prefix": "projects"
                }
            }
        }
    ]
}
```

Testar a política com a AWS CLI

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 [Desenvolvimento com o Amazon S3 usando a AWS CLI \(p. 1257\)](#).

```
aws s3api list-objects --bucket awsexamplebucket1 --prefix examplefolder --profile AccountADave
```

Se o bucket tiver o versionamento habilitado, para listar os objetos no bucket, você deverá conceder a permissão `s3:ListBucketVersions` na política anterior, em vez da permissão `s3>ListBucket`. Essa permissão também oferece suporte à chave de condição `s3:prefix`.

Exemplo 3: definir o número máximo de chaves

Você pode usar a chave de condição `s3:max-keys` para definir o número máximo de chaves que o solicitante pode retornar em uma solicitação [GET Bucket \(ListObjects\)](#) ou [ListObjectVersions](#). Por padrão, a API retorna até 1.000 chaves. Para obter uma lista de operadores de condição numérica que você pode usar com `s3:max-keys` e exemplos associados, consulte [Operadores de condição numérica](#) no Guia do usuário do IAM.

Ações, recursos e chaves de condição do Amazon S3

O Amazon S3 (prefixo do serviço: `s3`) fornece os seguintes recursos, ações e chaves de contexto de condição específicos ao serviço para uso em políticas de permissão do IAM.

Note

Você pode usar as ações listadas abaixo nas políticas do IAM e políticas do bucket do Amazon S3 para conceder permissões para operações específicas da API do Amazon S3. A maioria das ações tem o mesmo nome que as operações de API às quais são associadas. No entanto, em alguns casos, os nomes da ação e da operação da API são diferentes. Além disso, uma única ação pode controlar o acesso a mais de uma operação e algumas operações exigem várias ações diferentes.

Referências:

- Saiba como [configurar este serviço](#).
- Visualize uma lista das [operações de API disponíveis para este serviço](#).
- Saiba como proteger este serviço e seus recursos [usando políticas de permissão do IAM](#).

Tópicos

- [Ações definidas pelo Amazon S3 \(p. 458\)](#)
- [Tipos de recursos definidos pelo Amazon S3 \(p. 512\)](#)
- [Chaves de condição do Amazon S3 \(p. 512\)](#)

Ações definidas pelo Amazon S3

Você pode especificar as seguintes ações no elemento `Action` de uma declaração de política do IAM. Use políticas para conceder permissões para executar uma operação na AWS. Quando usa uma ação em uma política, você geralmente permite ou nega acesso à operação da API ou ao comando da CLI com o mesmo nome. No entanto, em alguns casos, uma única ação controla o acesso a mais de uma operação. Como alternativa, algumas operações exigem várias ações diferentes.

A coluna `Resource types` (Tipos de recursos) indica se cada ação é compatível com permissões no nível do recurso. Se não houver valor para essa coluna, você deverá especificar todos os recursos ("*") no elemento `Resource` de sua declaração de política. Se a coluna incluir um tipo de recurso, você poderá especificar um ARN desse tipo em uma declaração com essa ação. Os recursos obrigatórios são indicados

na tabela com um asterisco (*). Se você especificar um ARN de permissão no nível do recurso em uma instrução que esteja usando essa ação, ele deverá ser desse tipo. Algumas ações oferecem suporte a vários tipos de recursos. Se o tipo de recurso for opcional (não indicado como obrigatório), você poderá optar por usar um, mas não o outro.

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
AbortMultipartUpload	Concede permissão para anular upload multipart upload	Write	object* (p. 512)	s3:DataAccessPointArn (p. 513) s3:DataAccessPointAccount (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
BypassGovernanceRetention	Concede permissão para controlar as configurações de retenção de objetos no modo de governança	Permissions management	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:RequestObjectTag<key> (p. 513) s3:RequestObjectTagKeys (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-acl (p. 514) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
				s3:x-amz-copy-source (p. 514) s3:x-amz-grant-full-control (p. 514) s3:x-amz-grant-read (p. 514) s3:x-amz-grant-read-acp (p. 514) s3:x-amz-grant-write (p. 514) s3:x-amz-grant-write-acp (p. 515) s3:x-amz-metadata-directive (p. 515) s3:x-amz-server-side-encryption (p. 515) s3:x-amz-server-side-encryption-aws-kms-key-id (p. 515) s3:x-amz-server-side-encryption-customer-algorithm (p. 515) s3:x-amz-storage-class (p. 515)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
				s3:x-amz-website-redirect-location (p. 515) s3:object-lock-mode (p. 514) s3:object-lock-retain-until-date (p. 514) s3:object-lock-remaining-retention-days (p. 514) s3:object-lock-legal-hold (p. 514)	
CreateAccessPoint	Concede permissão para criar um novo ponto de acesso	Write	accesspoint* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:locationconstraint (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-acl (p. 514) s3:x-amz-content-sha256 (p. 514)	
CreateAccessPointForObjectLambda		Write	objectlambdaaccesspoint* (p. 512)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
	Concede permissão para criar um ponto de acesso habilitado para lambda objeto			s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
CreateBucket	Concede permissão para criar um novo bucket	Write	bucket* (p. 512)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
				s3:authType (p. 514) s3:locationconstraint (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-acl (p. 514) s3:x-amz-content-sha256 (p. 514) s3:x-amz-grant-full-control (p. 514) s3:x-amz-grant-read (p. 514) s3:x-amz-grant-read-acp (p. 514) s3:x-amz-grant-write (p. 514) s3:x-amz-grant-write-acp (p. 515)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
CreateJob	Concede permissão para criar um novo trabalho de operações em lote do Amazon S3	Write		s3:authType (iam:PassRole) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514) s3:RequestJobPriority (p. 513) s3:RequestJobOperation (p. 513) aws:TagKeys (p. 513) aws:RequestTag/\${TagKey} (p. 513)	
CreateMultipartUpload	Inicia um multipart upload e retorna um ID de upload	Write	object* (p. 512)	s3:DataAccessPointArn (p. 513) s3:DataAccessPointAccount (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
DeleteAccessPoint		Write	accesspoint* (p. 512)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
	Concede permissão para excluir o ponto de acesso nomeado no URI			s3:DataAccessPointArn (p. 513) s3:DataAccessPointAccount (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
DeleteAccessPointFromObjectLambda	Concede permissão para excluir o ponto de acesso habilitado para o objeto lambda nomeado no URI	Write	objectlambdaaccesspoint* (p. 512)	s3:DataAccessPointArn (p. 513) s3:DataAccessPointAccount (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
DeleteAccessPointPolicy		Permissions management	accesspoint* (p. 512)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
	Concede permissão para excluir a política em um ponto de acesso especificado			s3:DataAccessPointArn (p. 513) s3:DataAccessPointAccount (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
DeleteAccessPointPolicyForObject	Concede permissão para excluir a política em um ponto de acesso habilitado para o objeto lambda especificado	Permissions management	objectlambdaaccesspoint* (p. 512)	s3:DataAccessPointArn (p. 513) s3:DataAccessPointAccount (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
DeleteBucket	Concede permissão para excluir o bucket nomeado no URI	Write	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
DeleteBucketOwner	Concede permissão para excluir controle de propriedade em um bucket	Write	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
DeleteBucketPolicy	Concede permissão para excluir a política em um bucket especificado	Permissions management	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
DeleteBucketWebsite	Concede permissão para remover a configuração do site de um bucket	Write	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
DeleteJobTagging	Concede permissão para remover etiquetas de um trabalho existente de operações em lote do Amazon S3	Atribuição de tags (tagging)	job* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514) s3:ExistingJobPriority (p. 513) s3:ExistingJobOperation (p. 513)	
DeleteObject	Concede permissão para remover a versão nula de um objeto e inserir um marcador de exclusão, que se torna a versão atual do objeto	Write	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
DeleteObjectTags	Concede permissão para usar o sub-recurso de marcação para remover todo o conjunto de etiquetas do objeto especificado	Marcação	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:ExistingObjectTag/<key> (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
DeleteObjectVersion	Concede permissão para remover uma versão específica de um objeto	Write	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:versionid (p. 514) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
DeleteObjectVersion	Concede permissão para removendo o conjunto de etiquetas para uma versão específica do objeto	Marcação	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:ExistingObjectTag/<key> (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:versionid (p. 514) s3:x-amz-content-sha256 (p. 514)	
DeleteStorageLensConfiguration	Concede permissão para excluir uma configuração existente do Amazon S3 Storage Lens	Write	storagelensconfiguration* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
DeleteStorageLensConfiguration	Concede permissão para remover etiquetas de uma configuração existente do Amazon S3 Storage Lens	Marcação	storagelensconfiguration* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
DescribeJob	Concede permissão para recuperar os parâmetros de configuração e o status de um trabalho de operações em lote	Read	job* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetAccelerateConfiguration	Concede permissão para usar o recurso de aceleração para retornar o estado de Transfer Acceleration de um bucket, que é Habilido ou Suspensão	Read	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
GetAccessPoint	Concede permissão para retornar informações de configuração sobre o ponto de acesso especificado	Read		s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetAccessPointConfigurations	Concede permissão para recuperar a configuração de um ponto de acesso habilitado para o objeto lambda	Read	objectlambdaaccesspoint* (p. 512)	s3:DataAccessPointArn (p. 513) s3:DataAccessPointAccount (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetAccessPointForObjectLambda		Read	objectlambdaaccesspoint* (p. 512)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
	Concede permissão para criar um ponto de acesso habilitado para lambda objeto			s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetAccessPointPolicy	Concede permissão para retornar a política de ponto de acesso associada ao ponto de acesso especificado	Read	accesspoint* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
GetAccessPointPolicy	Concede permissão para retornar a política de ponto de acesso associada ao ponto de acesso habilitado para o objeto lambda especificado	Read	objectlambdaaccesspoint* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetAccessPointPolicyStatistic	Concede permissão para retornar a política para uma política de ponto de acesso específica	Read	accesspoint* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
GetAccessPointPolicy	Concede permissão para retornar a política de ponto de acesso para uma política de ponto de acesso específica de objeto lambda	Read	objectlambdaaccesspoint* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetAccountPublicAccessBlock	Concede permissão para recuperar a configuração PublicAccessBlock para uma Conta da AWS	Read		s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetAnalyticsConfiguration	Concede permissão para obter a configuração de análise de um bucket do Amazon S3, identificada pelo ID de configuração de análise	Read	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
GetBucketAcl	Concede permissão para usar o sub-recurso acl para retornar a lista de controle de acesso (ACL) de um bucket do Amazon S3	Read	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetBucketCORS	Concede permissão para retornar as informações de configuração de CORS definidas para um bucket do Amazon S3	Read	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetBucketLocation	Concede permissão para retornar a região em que um bucket do Amazon S3 reside	Read	bucket* (p. 512)		
GetBucketLogging	Concede permissão para retornar o status do registro em log de um bucket do Amazon S3 e as permissões que os usuários têm para visualizar ou modificar esse status	Read	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetBucketNotification		Read	bucket* (p. 512)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
	Concede permissão para obter a configuração de notificação de um bucket do Amazon S3			s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetBucketObjectLegalHold	Concede permissão para obter a configuração de bloqueio de objetos de um bucket do Amazon S3	Read	bucket* (p. 512)		
				s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:signatureversion (p. 514)	
GetBucketOwnershipControls	Concede permissão para manipular controles de propriedade em um bucket	Read	bucket* (p. 512)		
				s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetBucketPolicy	Concede permissão para retornar a política do bucket especificado	Read	bucket* (p. 512)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
				s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetBucketPolicyStatus	Concede permissão para recuperar o status da política de um bucket específico do Amazon S3, o que indica se o bucket é público	Read	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetBucketPublicAccessBlock	Concede permissão para recuperar a configuração PublicAccessBlock para um bucket do Amazon S3	Read	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
GetBucketRequestPayment	Concede permissão para retornar a configuração de pagamento da solicitação para um bucket do Amazon S3	Read	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetBucketTagging	Concede permissão para retornar o conjunto de etiquetas associado a um bucket do Amazon S3	Read	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetBucketVersioning	Concede permissão para retornar o estado de versionamento de um bucket do Amazon S3	Read	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetBucketWebsite		Read	bucket* (p. 512)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
	Concede permissão para retornar a configuração do site para um bucket do Amazon S3			s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetEncryptionConfiguration	Concede permissão para retornar a configuração de criptografia padrão de um bucket do Amazon S3	Read	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetIntelligentTieringConfiguration	Concede permissão para obter ou configurar todas a configuração do Amazon S3 Intelligent Tiering em um bucket do S3	Read	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
GetInventoryConfiguration	Concede permissão para retornar a configuração de inventário de um bucket do Amazon S3, identificado pelo ID de configuração de inventário	Read	bucket* (p. 512)		
				s3:authType (p. 514)	
				s3:ResourceAccount (p. 513)	
				s3:signatureAge (p. 514)	
				s3:signatureversion (p. 514)	
				s3:TlsVersion (p. 513)	
				s3:x-amz-content-sha256 (p. 514)	
GetJobTagging	Concede permissão para retornar o conjunto de etiquetas de um trabalho existente de operações em lote do Amazon S3	Read	job* (p. 512)		
				s3:authType (p. 514)	
				s3:ResourceAccount (p. 513)	
				s3:signatureAge (p. 514)	
				s3:signatureversion (p. 514)	
				s3:TlsVersion (p. 513)	
				s3:x-amz-content-sha256 (p. 514)	
GetLifecycleConfiguration	Concede permissão para retornar as informações de configuração de ciclo de vida definidas em um bucket do Amazon S3	Read	bucket* (p. 512)		
				s3:authType (p. 514)	
				s3:ResourceAccount (p. 513)	
				s3:signatureAge (p. 514)	
				s3:signatureversion (p. 514)	
				s3:TlsVersion (p. 513)	
				s3:x-amz-content-sha256 (p. 514)	
GetMetricsConfiguration		Read	bucket* (p. 512)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
	Concede permissão para obter uma configuração de métricas de um bucket do Amazon S3			s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetObject	Concede permissão para recuperar objetos do Amazon S3	Read	object* (p. 512)		
				s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:ExistingObjectTag/<key> (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
GetObjectAcl	Concede permissão para retornar a lista de controle de acesso (ACL) de um objeto	Read	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:ExistingObjectTag/<key> (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetObjectLegalHold	Concede permissão para obter o status atual de retenção legal de um objeto	Read	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetObjectRetention		Read	object* (p. 512)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
	Concede permissão para recuperar as configurações de retenção de um objeto			s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetObjectTagging	Concede permissão para retornar o conjunto de etiquetas de um objeto	Read	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:ExistingObjectTag/<key> (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)				

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
GetObjectVersion	Concede permissão para recuperar uma versão específica de um objeto	Read	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:ExistingObjectTag/<key> (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:versionid (p. 514) s3:x-amz-content-sha256 (p. 514)	
GetObjectVersion	Concede permissão para retornar a lista de controle de acesso (ACL) de uma versão específica de objeto	Read	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:ExistingObjectTag/<key> (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:versionid (p. 514) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
GetObjectVersion	Concede permissão para replicar objetos não criptografados e objetos criptografados com SSE-S3 ou SSE-KMS	Read	object* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
GetObjectVersionTagging	Concede permissão para retornar o conjunto de etiquetas para uma versão específica do objeto	Read	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:ExistingObjectTag/<key> (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:versionid (p. 514) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
GetReplicationConfiguration	Concede permissão para obter informações de configuração de replicação definidas em um bucket do Amazon S3	Read	bucket* (p. 512)		
				s3:authType (p. 514)	
				s3:ResourceAccount (p. 513)	
				s3:signatureAge (p. 514)	
				s3:signatureversion (p. 514)	
				s3:TlsVersion (p. 513)	
				s3:x-amz-content-sha256 (p. 514)	
GetStorageLensConfiguration	Concede permissão para obter a configuração do Amazon S3 Storage Lens	Read	storagelensconfiguration* (p. 512)		
				s3:authType (p. 514)	
				s3:ResourceAccount (p. 513)	
				s3:signatureAge (p. 514)	
				s3:signatureversion (p. 514)	
				s3:TlsVersion (p. 513)	
				s3:x-amz-content-sha256 (p. 514)	
GetStorageLensConfigurationForTemplate	Concede permissão para obter conjuntos de templates de uma configuração existente do Amazon S3 Storage Lens	Read	storagelensconfiguration* (p. 512)		
				s3:authType (p. 514)	
				s3:ResourceAccount (p. 513)	
				s3:signatureAge (p. 514)	
				s3:signatureversion (p. 514)	
				s3:TlsVersion (p. 513)	
				s3:x-amz-content-sha256 (p. 514)	
GetStorageLensDashboard		Read	storagelensconfiguration* (p. 512)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
	Concede permissão para obter um painel do Amazon S3 Storage Lens			s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
ListAccessPoints	Concede permissão para listar pontos de acesso	Read		s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
ListAccessPointsForLambda	Concede permissão para listar os pontos de acesso habilitados para objeto lambda	Read		s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
ListAllMyBuckets	Concede permissão para listar todos os buckets de propriedade do remetente autenticado da solicitação	List		s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
ListBucket	Concede permissão para listar alguns ou todos os objetos em um bucket do Amazon S3 (até 1.000)	List	bucket* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:delimiter (p. 514) s3:max-keys (p. 514) s3:prefix (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
ListBucketMultipartUploads	Concede permissão para listar uploads em andamento	List	bucket* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
ListBucketVersion	Concede permissão para listar metadados sobre todas as versões de objetos em um bucket do Amazon S3	List	bucket* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:delimiter (p. 514) s3:max-keys (p. 514) s3:prefix (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
ListJobs	Concede permissão para listar trabalhos atuais e trabalhos que terminaram recentemente	List		s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
ListMultipartUploadParts	Concede permissão para listar as partes que foram carregadas para um multipart upload específico	List	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
ListStorageLensConfigurations	Concede permissão para listar configurações do Amazon S3 Storage Lens	List		s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
ObjectOwnerOverridePreventsDelete	Concede permissão para alterar as propriedades da réplica	Permissions management	object* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
PutAccelerateConfiguration	Concede permissão para usar o recurso de aceleração para definir o estado de Transfer Acceleration de um bucket existente do S3	Write	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
PutAccessPointConfiguration	Concede permissão para definir a configuração do ponto de acesso habilitado para objeto lambda	Write	objectlambdaaccesspoint* (p. 512)	s3:DataAccessPointArn (p. 513) s3:DataAccessPointAccount (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
PutAccessPointPolicy	Concede permissão para associar uma política de acesso a um ponto de acesso especificado	Permissions management	accesspoint* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
PutAccessPointPolicyForObjectLambda	Concede permissão para associar uma política de acesso a um ponto de acesso especificado habilitado para objeto lambda	Permissions management	objectlambdaaccesspoint* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
PutAccountPublicAccessBlock	Concede permissão para criar a configuração PublicAccessBlock para uma Conta da AWS	Permissions management		s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
PutAnalyticsConfiguration	Concede permissão para definir a configuração de análise para o bucket, especificada pelo ID de configuração de análise	Write	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
PutBucketAcl	Concede permissão para definir as permissões em um bucket existente usando listas de controle de acesso (ACLs)	Permissions management	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-acl (p. 514) s3:x-amz-content-sha256 (p. 514) s3:x-amz-grant-full-control (p. 514) s3:x-amz-grant-read (p. 514) s3:x-amz-grant-read-acp (p. 514) s3:x-amz-grant-write (p. 514) s3:x-amz-grant-write-acp (p. 515)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
PutBucketCORS	Concede permissão para definir a configuração de CORS para um bucket do Amazon S3	Write	bucket* (p. 512)		
				s3:authType (p. 514)	
				s3:ResourceAccount (p. 513)	
				s3:signatureAge (p. 514)	
				s3:signatureversion (p. 514)	
				s3:TlsVersion (p. 513)	
				s3:x-amz-content-sha256 (p. 514)	
PutBucketLogging	Concede permissão para definir os parâmetros de registro em log para um bucket do Amazon S3	Write	bucket* (p. 512)		
				s3:authType (p. 514)	
				s3:ResourceAccount (p. 513)	
				s3:signatureAge (p. 514)	
				s3:signatureversion (p. 514)	
				s3:TlsVersion (p. 513)	
				s3:x-amz-content-sha256 (p. 514)	
PutBucketNotification	Concede permissão para receber notificações quando determinados eventos acontecem em um bucket do Amazon S3	Write	bucket* (p. 512)		
				s3:authType (p. 514)	
				s3:ResourceAccount (p. 513)	
				s3:signatureAge (p. 514)	
				s3:signatureversion (p. 514)	
				s3:TlsVersion (p. 513)	
				s3:x-amz-content-sha256 (p. 514)	
PutBucketObjectLockConfiguration		Write	bucket* (p. 512)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
	Concede permissão para colocar a configuração de bloqueio de objetos em um bucket específico			s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:TlsVersion (p. 513) s3:signatureversion (p. 514)	
PutBucketOwnershipControls	Concede permissão para adicionar ou substituir controles de propriedade em um bucket	Write	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
PutBucketPolicy	Concede permissão para adicionar ou substituir uma política de bucket em um bucket	Permissions management	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
PutBucketPublicAccessBlock	Concede permissão para criar ou modificar a configuração PublicAccessBlock para um bucket específico do Amazon S3	Permissions management	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
PutBucketRequestPayment	Concede permissão para definir a configuração de pagamento de solicitação de um bucket	Write	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
PutBucketTagging	Concede permissão para adicionar um conjunto de etiquetas a um bucket existente do Amazon S3	Marcação	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
PutBucketVersioning	Concede permissão para definir o estado de versionamento de um bucket existente do Amazon S3	Write	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
PutBucketWebsite	Concede permissão para definir a configuração do site especificado no sub-recurso do site	Write	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
PutEncryptionConfiguration	Concede permissão para definir a configuração de criptografia para um bucket do Amazon S3	Write	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
PutIntelligentTieringConfiguration	Concede permissão para criar, modificar, atualizar ou excluir uma configuração do Amazon S3 Intelligent Tiering	Write	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
PutInventoryConfiguration	Concede permissão para adicionar uma configuração de inventário ao bucket, identificada pelo ID de inventário	Write	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
PutJobTagging	Concede permissão para substituir etiquetas em um trabalho existente de operações em lote do Amazon S3	Marcação	job* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514) s3:ExistingJobPriority (p. 513) s3:ExistingJobOperation (p. 513) aws:TagKeys (p. 513) aws:RequestTag/\${TagKey} (p. 513)	
PutLifecycleConfiguration	Concede permissão para criar uma configuração de ciclo de vida para o bucket ou substituir uma configuração de ciclo de vida existente	Write	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
PutMetricsConfiguration	Concede permissão para definir ou atualizar uma configuração de métricas para as métricas de solicitação do CloudWatch de um bucket do Amazon S3	Write	bucket* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
PutObject	Concede permissão para adicionar um objeto a um bucket	Write	object* (p. 512)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
					s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:RequestObjectTag/ <key> (p. 513) s3:RequestObjectTagKeys (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-acl (p. 514) s3:x-amz-content-sha256 (p. 514) s3:x-amz-copy-source (p. 514) s3:x-amz-grant-full-control (p. 514) s3:x-amz-grant-read (p. 514) s3:x-amz-grant-read-acp (p. 514) s3:x-amz-grant-write (p. 514) s3:x-amz-grant-write-acp (p. 515) s3:x-amz-metadata-directive (p. 515)

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
				s3:x-amz-server-side-encryption (p. 515) s3:x-amz-server-side-encryption-aws-kms-key-id (p. 515) s3:x-amz-server-side-encryption-customer-algorithm (p. 515) s3:x-amz-storage-class (p. 515) s3:x-amz-website-redirect-location (p. 515) s3:object-lock-mode (p. 514) s3:object-lock-retain-until-date (p. 514) s3:object-lock-remaining-retention-days (p. 514) s3:object-lock-legal-hold (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes	
PutObjectAcl	Concede permissão para definir as permissões de lista de controle de acesso (ACL) para objetos novos ou existentes em um bucket do S3.	Permissions management	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:ExistingObjectTag/<key> (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-acl (p. 514) s3:x-amz-content-sha256 (p. 514) s3:x-amz-grant-full-control (p. 514) s3:x-amz-grant-read (p. 514) s3:x-amz-grant-read-acp (p. 514) s3:x-amz-grant-write (p. 514) s3:x-amz-grant-write-acp (p. 515) s3:x-amz-storage-class (p. 515)		
PutObjectLegalHold		Write	object* (p. 512)			

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
	Concede permissão para aplicar uma configuração de retenção legal ao objeto especificado			s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514) s3:object-lock-legal-hold (p. 514)	
PutObjectRetention		Write	object* (p. 512)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
	Concede permissão para colocar uma configuração de retenção de objetos em um objeto				s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514) s3:object-lock-mode (p. 514) s3:object-lock-retain-until-date (p. 514) s3:object-lock-remaining-retention-days (p. 514)

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
PutObjectTagging	Concede permissão para definir o conjunto de etiquetas fornecido para um objeto que já existe em um bucket ou ao carregar pela primeira vez o objeto.	Marcação	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:ExistingObjectTag/<key> (p. 513) s3:RequestObjectTag/<key> (p. 513) s3:RequestObjectTagKeys (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes	
PutObjectVersion	Concede permissão para usar o sub-recurso acl para definir as permissões de lista de controle de acesso (ACL) para um objeto que já existe em um bucket	Permissions management	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:ExistingObjectTag/<key> (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:versionid (p. 514) s3:x-amz-acl (p. 514) s3:x-amz-content-sha256 (p. 514) s3:x-amz-grant-full-control (p. 514) s3:x-amz-grant-read (p. 514) s3:x-amz-grant-read-acp (p. 514) s3:x-amz-grant-write (p. 514) s3:x-amz-grant-write-acp (p. 515) s3:x-amz-storage-class (p. 515)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
PutObjectVersionTagging	Concede permissão para definir o conjunto de etiquetas fornecido para uma versão específica de um objeto	Marcação	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:ExistingObjectTag/<key> (p. 513) s3:RequestObjectTag/<key> (p. 513) s3:RequestObjectTagKeys (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:versionid (p. 514) s3:x-amz-content-sha256 (p. 514)	
PutReplicationConfiguration	Concede permissão para criar ou substituir uma existente configuração de replicação	Write	bucket* (p. 512)	iam:PassRole	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
PutStorageLensConfiguration	Concede permissão para criar ou atualizar uma configuração do Amazon S3 Storage Lens	Write		s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514) aws:TagKeys (p. 513) aws:RequestTag/\${TagKey} (p. 513)	
PutStorageLensConfigurationSubstitutionTags	Concede permissão para colocar substituições de tags em uma configuração existente do Amazon S3 Storage Lens	Marcação	storagelensconfiguration*	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514) aws:TagKeys (p. 513) aws:RequestTag/\${TagKey} (p. 513)	
ReplicateDelete	Concede permissão para replicar marcadores de exclusão no bucket de destino	Write	object*	object* (p. 512) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
ReplicateObject	Concede permissão para replicar objetos e etiquetas de objetos no bucket de destino	Write	object* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514) s3:x-amz-server-side-encryption (p. 515) s3:x-amz-server-side-encryption-aws-kms-key-id (p. 515) s3:x-amz-server-side-encryption-customer-algorithm (p. 515)	
ReplicateTags	Concede permissão para replicar etiquetas de objetos no bucket de destino	Marcação	object* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
RestoreObject	Concede permissão para restaurar uma cópia arquivada de um objeto novamente no Amazon S3	Write	object* (p. 512)	s3:DataAccessPointAccount (p. 513) s3:DataAccessPointArn (p. 513) s3:AccessPointNetworkOrigin (p. 513) s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514)	
UpdateJobPriority	Concede permissão para atualizar a prioridade de um trabalho existente	Write	job* (p. 512)	s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514) s3:RequestJobPriority (p. 513) s3:ExistingJobPriority (p. 513) s3:ExistingJobOperation (p. 513)	
UpdateJobStatus		Write	job* (p. 512)		

Ações	Descrição	Nível de acesso	Tipos de recursos (*necessário)	Chaves de condição	Ações dependentes
	Concede permissão para atualizar o status do trabalho especificado			s3:authType (p. 514) s3:ResourceAccount (p. 513) s3:signatureAge (p. 514) s3:signatureversion (p. 514) s3:TlsVersion (p. 513) s3:x-amz-content-sha256 (p. 514) s3:ExistingJobPriority (p. 513) s3:ExistingJobOperation (p. 513) s3:JobSuspendedCause (p. 513)	

Tipos de recursos definidos pelo Amazon S3

Os seguintes tipos de recursos são definidos por este serviço e podem ser usados no elemento `Resource` de declarações de políticas de permissão do IAM. Cada ação na [Tabela de ações \(p. 458\)](#) identifica os tipos de recursos que podem ser especificados com essa ação. Um tipo de recurso também pode definir quais chaves de condição você pode incluir em uma política. Essas chaves são exibidas na última coluna da tabela.

Tipos de recursos	ARN	Chaves de condição
accesspoint	arn:\${Partition}:s3:\${Region}: \${Account}:accesspoint/\${AccessPointName}	
bucket	arn:\${Partition}:s3:::\${BucketName}	
object	arn:\${Partition}:s3:::\${BucketName} / \${ObjectName}	
job	arn:\${Partition}:s3:\${Region}: \${Account}:job/\${JobId}	
storagelensconfig	arn:\${Partition}:s3:\${Region}: \${Account}:storage-lens/\${ConfigId}	aws:ResourceTag/\${TagKey} (p. 513)
objectlambdaaccesspoint	arn:\${Partition}:s3-object-lambda:\${Region}: \${Account}:accesspoint/\${AccessPointName}	

Chaves de condição do Amazon S3

O Amazon S3 define as seguintes chaves de condição que podem ser usadas no elemento `Condition` de uma política do IAM. É possível usar essas chaves para refinar ainda mais as condições sob as quais a declaração de política se aplica.

Para visualizar as chaves de condição globais disponíveis para todos os serviços, consulte [Chaves de condição globais disponíveis](#).

Chaves de condição	Descrição	Type
aws:RequestTag/\${TagKey}	Filtre ações com base nas etiquetas transmitidas na solicitação	String
aws:ResourceTag/\${TagKey}	Filtre as ações com base nas etiquetas associadas ao recurso	String
aws:TagKeys	Filtre ações com base nas chaves da etiqueta transmitidas na solicitação	String
s3:AccessPointNetworkOrigin	Filtre o acesso pela origem de rede (Internet ou VPC)	String
s3:DataAccessPointAdopter	Filtre o acesso pelo ID da Conta da AWS que é proprietária do ponto de acesso	String
s3:DataAccessPointOwner	Filtre o acesso pelo nome de recurso da Amazon (ARN) de um ponto de acesso	String
s3:ExistingJobOperation	Filtre o acesso à atualização da prioridade de trabalho por operação	String
s3:ExistingJobPriority	Filtre o acesso ao cancelamento de trabalhos existentes por intervalo de prioridade	Numeric
s3:ExistingObjectTag/existente <key>	Filtre o acesso por chave e valor da etiqueta de um objeto	String
s3:JobSuspendedCause	Filtre o acesso ao cancelamento de trabalhos suspensos por uma causa específica do trabalho suspenso (por exemplo, AWAITING_CONFIRMATION)	String
s3:LocationConstraint	Filtre o acesso por uma região específica	String
s3:RequestJobOperation	Filtre o acesso à criação de trabalhos por operação	String
s3:RequestJobPriority	Filtre o acesso à criação de novos trabalhos por intervalo de prioridade	Numeric
s3:RequestObjectTag/adicionados <key>	Filtre o acesso pelas chaves e valores da etiqueta a serem adicionados aos objetos	String
s3:RequestObjectTag/objectos	Filtre o acesso pelas chaves de tag a serem adicionadas aos objetos	String
s3:ResourceAccount	Filtre o acesso pelo ID da Conta da AWS do proprietário do recurso	String
s3:TlsVersion	Filtre o acesso pela versão TLS usada pelo cliente	Numeric

Chaves de condição	Descrição	Type
s3:VersionId	Filtrar o acesso por uma versão de objeto específica	String
s3:authType	Filtrar o acesso por método de autenticação	String
s3:delimiter	Filtrar o acesso por parâmetro delimitador	String
s3:locationconstraint	Filtrar o acesso por uma região específica	String
s3:max-keys	Filtrar o acesso pelo número máximo de chaves retornadas em uma solicitação ListBucket	Numeric
s3:object-lock-legal-hold	Filtrar o acesso pelo status de obtenção legal do objeto	String
s3:object-lock-mode	Filtrar o acesso por modo de retenção do objeto (COMPLIANCE ou GOVERNANCE, conformidade ou governança)	String
s3:object-lock-remaining-retention-days	Filtrar o acesso pelos dias restantes da retenção do objeto	String
s3:object-lock-retain-until-date	Filtrar o acesso pela data de término de retenção do objeto	String
s3:prefix	Filtrar o acesso pelo prefixo do nome da chave	String
s3:signatureAge	Filtrar o acesso pela idade em milissegundos da assinatura da solicitação	Numeric
s3:signatureversion	Filtrar o acesso pela versão do AWS Signature usada na solicitação	String
s3:versionid	Filtrar o acesso por uma versão de objeto específica	String
s3:x-amz-acl	Filtrar o acesso pela ACL padrão no cabeçalho x-amz-acl da solicitação	String
s3:x-amz-content-sha256	Filtrar o acesso ao conteúdo não assinado no bucket	String
s3:x-amz-copy-source	Filtrar o acesso a solicitações com bucket, prefixo ou objeto específico como a fonte de cópia	String
s3:x-amz-grant-full-control	Filtrar o acesso a solicitações com o cabeçalho x-amz-grant-full-control (controle total)	String
s3:x-amz-grant-read	Filtrar o acesso a solicitações com o cabeçalho x-amz-grant-read (acesso de leitura)	String
s3:x-amz-grant-read-acp	Filtrar o acesso a solicitações com o cabeçalho x-amz-grant-read-acp (permissões de leitura para a ACL)	String
s3:x-amz-grant-write	Filtrar o acesso a solicitações com o cabeçalho x-amz-grant-write (acesso de gravação)	String

Chaves de condição	Descrição	Type
s3:x-amz-grant-write-acp	Filtre o acesso a solicitações com o cabeçalho x-amz-grant-write-acp (permissões de gravação para a ACL)	String
s3:x-amz-metadata-directive	Filtre acesso pelo comportamento de metadados do objeto (COPY ou REPLACE, copiar ou substituir) quando objetos são copiados	String
s3:x-amz-server-side-encryption	Filtre o acesso pela criptografia do lado do servidor	String
s3:x-amz-server-side-encryption-aws-kms-key-id	Filtre o acesso pela criptografia do lado do servidor do AWS KMS key	String
s3:x-amz-server-side-encryption-customer-algorithm	Filtre o acesso por algoritmo especificado pelo cliente para criptografia no lado do servidor	String
s3:x-amz-storage-class	Filtre o acesso por classe de armazenamento	String
s3:x-amz-website-redirect-location	Filtre o acesso por um local de redirecionamento de site específico para buckets configurados como sites estáticos	String

Uso de políticas de bucket

Você pode criar e configurar políticas de bucket para conceder permissão aos seus recursos do Amazon S3.

Uma política de bucket é baseada em recursos que você pode usar para conceder permissões de acesso ao bucket e aos objetos contidos nele. Só o proprietário do bucket pode associar uma política a um bucket. As permissões anexadas ao bucket se aplicam a todos os objetos do bucket que pertencem ao proprietário do bucket. Essas permissões não se aplicam a objetos de propriedade de outras Contas da AWS.

Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. É possível usar Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Como resultado, o controle de acesso para seus dados é baseado em políticas, como políticas do IAM, políticas de bucket do S3, políticas de endpoint da Virtual Private Cloud (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

As políticas de bucket usam uma linguagem de política de acesso baseada em JSON. Você pode usar políticas de bucket para adicionar ou negar permissões para os objetos em um bucket. As políticas de bucket permitem ou negam solicitações com base nos elementos da política, incluindo o solicitante, ações do S3, recursos e aspectos ou condições da solicitação (por exemplo, o endereço IP usado para fazer a solicitação). Por exemplo, você pode criar uma política de bucket que conceda permissões entre contas para carregar objetos em um bucket do S3 enquanto garante que o proprietário do bucket tenha controle total dos objetos carregados. Para obter mais informações, consulte [Exemplos de políticas de bucket \(p. 519\)](#).

Na política de bucket, você pode usar caracteres curinga nos nomes de recursos da Amazon (ARNs) e outros valores para conceder permissões a um subconjunto de objetos. Por exemplo, você pode controlar

o acesso a grupos de objetos que começam com um [prefixo](#) ou termine com uma determinada extensão, como `.html`.

Os tópicos nesta seção fornecem exemplos e mostram como adicionar uma política de bucket no console do S3. Para obter informações sobre as políticas de usuário do IAM, consulte [Uso de políticas de usuário do IAM \(p. 528\)](#). Para obter informações sobre a linguagem da política de bucket, consulte [Políticas e permissões no Amazon S3 \(p. 437\)](#)

Important

As políticas de bucket são limitadas a 20 KB.

Tópicos

- [Adicionar uma política de bucket usando o console do Amazon S3 \(p. 516\)](#)
- [Controlar o acesso a partir de VPC endpoints com políticas de bucket \(p. 517\)](#)
- [Exemplos de políticas de bucket \(p. 519\)](#)

Adicionar uma política de bucket usando o console do Amazon S3

Use o console do Amazon S3 para adicionar uma nova política de bucket ou editar uma política de bucket existente. Uma política de bucket é uma política do AWS Identity and Access Management (IAM) baseada em recursos. Você adiciona uma política de bucket a um bucket para conceder permissões de acesso ao bucket, e aos objetos contidos nele, a outras Contas da AWS ou usuários do IAM. As permissões de objeto aplicam-se somente aos objetos criados pela proprietário do bucket. Para obter mais informações sobre políticas de bucket, consulte [Visão geral do gerenciamento de acesso \(p. 419\)](#).

Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. É possível usar Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Como resultado, o controle de acesso para seus dados é baseado em políticas, como políticas do IAM, políticas de bucket do S3, políticas de endpoint da Virtual Private Cloud (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Solucionar avisos de segurança, erros, avisos gerais e sugestões de AWS Identity and Access Management Access Analyzer antes de salvar a política. O IAM Access Analyzer executa verificações de política para validar sua política em relação à [gramática das políticas](#) e às [práticas recomendadas](#) do IAM. Essas verificações geram descobertas e fornecem recomendações práticas que ajudam a criar políticas que sejam funcionais e estejam em conformidade com as práticas recomendadas de segurança. Para saber mais sobre a validação de políticas usando o IAM Access Analyzer, consulte [Validação de políticas do IAM Access Analyzer](#) no Guia do usuário IAM. Para visualizar uma lista de avisos, erros e sugestões retornados pelo IAM Access Analyzer, consulte [Referência de verificação de políticas do IAM Access Analyzer](#).

Para criar ou editar uma política de bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket para o qual você deseja criar uma política de bucket ou cuja política de bucket você deseja editar.
3. Escolha Permissions (Permissões).
4. Em Bucket policy (Política de bucket), escolha Edit (Editar). Isso abre a página Edit bucket policy (Editar política de bucket).
5. Na página Edit bucket policy (Editar política de buckets), explore Exemplos de políticas no Guia do usuário do Amazon S3, escolha Policy generator (Gerador de políticas) para gerar uma política automaticamente ou edite o JSON na seção Policy (Política).

Se você escolher Policy generator (Gerador de políticas), o AWS Policy Generator será aberto em uma nova janela:

- a. Na página AWS Policy Generator, em Select Type of Policy (Selecionar tipo de política), S3 Bucket Policy (Política de bucket do S3).
- b. Adicione uma instrução inserindo as informações nos campos fornecidos e escolha Add Statement (Adicionar instrução). Repita para todas as instruções que deseja adicionar. Para obter mais informações sobre esses campos, consulte a [Referência de elementos de política JSON do IAM](#) no Guia do usuário do IAM.

Note

Por conveniência, a página Edit bucket policy (Editar política do bucket) exibe o nome do recurso da Amazon (ARN) do bucket atual acima do campo de texto Policy (Política). Você pode copiar este ARN para usá-lo nas instruções da página do AWS Policy Generator.

- c. Depois de terminar de adicionar as instruções, escolha Generate Policy (Gerar política).
- d. Copie o texto da política gerada, escolha Close (Fechar) e retorne à página Edit bucket policy (Editar política de bucket) no console do Amazon S3.
6. Na caixa Policy (Política), edite a política existente ou cole a política de bucket do gerador de políticas. Solucione avisos de segurança, erros, avisos gerais e sugestões antes de salvar a política.
7. (Opcional) Pré-visualize como sua nova política afeta o acesso público e entre contas ao seu recurso. Antes de salvar sua política, você pode verificar se ela introduz novas descobertas do IAM Access Analyzer ou resolve as descobertas existentes. Se você não vir um analisador ativo, [crie um analisador de conta](#) no IAM Access Analyzer. Para obter mais informações, consulte [Visualizar acesso](#) no Guia do usuário do IAM.
8. Escolha Save changes (Salvar alterações), o que o levará à página Bucket Permissions (Permissões do bucket).

Controlar o acesso a partir de VPC endpoints com políticas de bucket

Você pode usar políticas de bucket do Amazon S3 para controlar o acesso a buckets de VPC endpoints específicos ou VPCs específicas. Esta seção contém políticas de bucket de exemplo que podem ser usadas para controlar o acesso ao bucket do Amazon S3 de VPC endpoints. Para saber como configurar VPC endpoints, consulte [VPC Endpoints](#) no Guia do usuário da VPC.

A VPC permite executar os recursos da AWS em uma rede virtual definida por você. Um endpoint permite que você crie uma conexão privada entre sua VPC e outro produto da AWS sem exigir acesso pela Internet por meio de uma conexão VPN, de uma instância NAT ou do AWS Direct Connect.

Um endpoint de VPC para Amazon S3 é uma entidade lógica em uma VPC que oferece conectividade apenas ao 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 alteram somente a forma como as solicitações são roteadas. Os endpoints públicos do Amazon S3 e os nomes de DNS continuarão funcionando com VPC endpoints. Para obter informações importantes sobre o uso de VPC endpoints com o Amazon S3, consulte [VPC Endpoints de gateway](#) e [Endpoints para Amazon S3](#) no Guia do usuário da VPC.

Os VPC endpoints para 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 o acesso aos serviços com VPC Endpoints](#) no Guia do usuário da VPC.
- É possível controlar quais VPCs ou VPC endpoints têm acesso aos seus buckets usando as políticas de bucket do Amazon 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 \(p. 518\)](#)
- [Restringir o acesso a uma VPC específica \(p. 518\)](#)

Important

Ao aplicar políticas de bucket do Amazon S3 para os VPC endpoints descritos nesta seção, talvez você bloquee o acesso ao bucket inadvertidamente. As permissões de bucket destinadas especificamente a 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 [Minha política de bucket tem o ID da VPC ou do endpoint da VPC incorreto.](#) [Como corrigir a política para que eu possa acessar o bucket?](#) na Central de conhecimento do AWS Support.

Restringir o acesso a um VPC endpoint específico

O seguinte é um exemplo de um política de bucket do Amazon S3 que restringe o acesso a um bucket específico, awsexamplebucket1, 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 nome de recurso da Amazon (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 [Exemplos de chave de condição do Amazon S3 \(p. 447\)](#).

Important

- Antes de usar a política de exemplo a seguir, substitua o ID do VPC endpoint por um valor apropriado para o caso de uso. Caso contrário, não será possível acessar o bucket.
- Essa política desabilita o acesso do console ao bucket especificado, pois as solicitações do console não se originam do VPC endpoint especificado.

```
{  
    "Version": "2012-10-17",  
    "Id": "Policy1415115909152",  
    "Statement": [  
        {  
            "Sid": "Access-to-specific-VPCE-only",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Effect": "Deny",  
            "Resource": ["arn:aws:s3:::awsexamplebucket1",  
                        "arn:aws:s3:::awsexamplebucket1/*"],  
            "Condition": {  
                "StringNotEquals": {  
                    "aws:SourceVpce": "vpce-1a2b3c4d"  
                }  
            }  
        }  
    ]  
}
```

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 Amazon S3 para todos os endpoints. A seguir encontra-se um exemplo de política que permite que a VPC vpc-111bbb22 acesse awsexamplebucket1 e seus objetos. Essa

política negará todo acesso ao bucket se o endpoint a VPC especificada não estiver sendo usada. A chave de condição `vpc-111bbb22` não requer um ARN para o recurso da VPC, somente o ID da VPC.

Important

- Antes de usar a política de exemplo a seguir, substitua o ID da VPC por um valor apropriado para o caso de uso. Caso contrário, não será possível acessar o bucket.
- Essa política desabilita o acesso do console ao bucket especificado, pois as solicitações do console não se originam da VPC especificada.

```
{  
    "Version": "2012-10-17",  
    "Id": "Policy1415115909153",  
    "Statement": [  
        {  
            "Sid": "Access-to-specific-VPC-only",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Effect": "Deny",  
            "Resource": ["arn:aws:s3:::awsexamplebucket1",  
                        "arn:aws:s3:::awsexamplebucket1/*"],  
            "Condition": {  
                "StringNotEquals": {  
                    "aws:SourceVpc": "vpc-111bbb22"  
                }  
            }  
        }  
    ]  
}
```

Exemplos de políticas de bucket

Esta seção apresenta alguns exemplos de casos de uso típicos de políticas de bucket. As políticas usam strings `DOC-EXAMPLE-BUCKET` no valor do recurso. Para testar essas políticas, será necessário substituir essas sequências pelo nome do bucket. Para obter informações sobre políticas de bucket, consulte [Uso de políticas de bucket \(p. 515\)](#). Para obter mais informações sobre o texto das políticas, consulte [Políticas e permissões no Amazon S3 \(p. 437\)](#).

Uma política de bucket é baseada em recursos que você pode usar para conceder permissões de acesso ao bucket e aos objetos contidos nele. Só o proprietário do bucket pode associar uma política a um bucket. As permissões anexadas ao bucket se aplicam a todos os objetos do bucket que pertencem ao proprietário do bucket. Essas permissões não se aplicam a objetos de propriedade de outras Contas da AWS.

Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. É possível usar Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Como resultado, o controle de acesso para seus dados é baseado em políticas, como políticas do IAM, políticas de bucket do S3, políticas de endpoint da Virtual Private Cloud (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Para obter mais informações sobre políticas de bucket, consulte [Uso de políticas de bucket \(p. 515\)](#).

Note

As políticas de bucket são limitadas a 20 KB.

Você pode usar o [AWS Policy Generator](#) para criar uma política de bucket para o bucket do Amazon S3. Depois, use o documento gerado para definir a política de bucket usando o [console do Amazon S3](#), por meio de várias ferramentas de terceiros ou pela sua aplicação.

Important

Ao testar as permissões usando o console do Simple Storage Service (Amazon S3), você deve conceder as permissões adicionais que o console exige: 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 [Controlar o acesso a um bucket com políticas de usuário \(p. 528\)](#).

Tópicos

- [Conceder permissões a várias contas com condições adicionadas \(p. 520\)](#)
- [Conceder permissão somente leitura para um usuário anônimo \(p. 521\)](#)
- [Limitar o acesso a endereços IP específicos \(p. 521\)](#)
- [Restringir o acesso a um indicador HTTP específico \(p. 523\)](#)
- [Conceder permissão para uma OAI do Amazon CloudFront \(p. 523\)](#)
- [Adicionar uma política de bucket para exigir MFA \(p. 524\)](#)
- [Conceder permissões entre contas para fazer upload de objetos garantindo que o proprietário do bucket tenha controle total \(p. 526\)](#)
- [Conceder permissões para o Amazon S3 Inventory e a análise do Simple Storage Service \(Amazon S3\) \(p. 526\)](#)
- [Conceder permissões para o Amazon S3 Storage Lens \(p. 527\)](#)

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 as solicitações para essas operações incluam a lista de controle de acesso (ACL) public-read predefinida. Para obter mais informações, consulte [Ações do Amazon S3 \(p. 441\)](#) e [Exemplos de chave de condição do Amazon S3 \(p. 447\)](#).

Warning

Tenha cautela ao conceder acesso anônimo ao seu bucket do Amazon S3 ou ao desabilitar as configurações de bloqueio de acesso público. Quando você concede acesso anônimo, qualquer pessoa no mundo pode acessar seu bucket. Recomendamos nunca conceder acesso anônimo ao seu bucket do Amazon S3, a menos que seja especificamente necessário, como com a [hospedagem de site estático \(p. 1188\)](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AddCannedAcl",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": [  
                    "arn:aws:iam::111122223333:root",  
                    "arn:aws:iam::444455556666:root"  
                ]  
            },  
            "Action": [  
                "s3:PutObject",  
                "s3:PutObjectAcl"  
            ],  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",  
            "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 [Ações do Amazon S3 \(p. 441\)](#). 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. Antes de usar uma política de bucket para conceder permissão somente leitura a um usuário anônimo, é necessário desabilitar as configurações de bloqueio de acesso público ao seu bucket. Para obter mais informações, consulte [Configuração de permissões para acesso ao site \(p. 1199\)](#).

Warning

Tenha cautela ao conceder acesso anônimo ao seu bucket do Amazon S3 ou ao desabilitar as configurações de bloqueio de acesso público. Quando você concede acesso anônimo, qualquer pessoa no mundo pode acessar seu bucket. Recomendamos nunca conceder acesso anônimo ao seu bucket do Amazon S3, a menos que seja especificamente necessário, como com a [hospedagem de site estático \(p. 1188\)](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicRead",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ]
        }
    ]
}
```

Limitar o acesso a endereços IP específicos

O exemplo a seguir nega permissões a qualquer usuário para executar quaisquer operações do Amazon S3 em objetos no bucket do S3 especificado, a menos que a solicitação tenha origem no intervalo de endereços IP especificados na condição.

A condição nessa instrução identifica **54.240.143.0/24** como o intervalo de endereços IP do Internet Protocol versão 4 (IPv4) permitidos.

O bloco Condition usa a condição 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 [Exemplos de chave de condição do Amazon S3 \(p. 447\)](#). Os valores IPv4 aws:SourceIp usam a notação CIDR padrão. Para obter mais informações, consulte [Referência de elementos de política JSON do IAM](#) no Guia do usuário do IAM.

Warning

Antes de usar essa política, substitua o intervalo de endereços IP **54.240.143.0/24** desse exemplo por um valor apropriado para o seu caso de uso. Caso contrário, você perderá a capacidade de acessar seu bucket.

```
{  
    "Version": "2012-10-17",  
    "Id": "S3PolicyId1",  
    "Statement": [  
        {  
            "Sid": "IPAllow",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Resource": [  
                "arn:aws:s3::::DOC-EXAMPLE-BUCKET",  
                "arn:aws:s3::::DOC-EXAMPLE-BUCKET/*"  
            ],  
            "Condition": {  
                "NotIpAddress": {  
                    "aws:SourceIp": "54.240.143.0/24"  
                }  
            }  
        }  
    ]  
}
```

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 permitiria acesso aos endereços IP **54.240.143.1** e **2001:DB8:1234:5678::1** e negaria acesso aos 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 IPv6 (por exemplo `2032001:DB8:1234:5678::/64`). Para obter mais informações, consulte [Operadores de condição de endereço IP](#) no Guia do usuário do IAM.

Warning

Substitua os intervalos de endereços IP neste exemplo por valores apropriados para o seu caso de uso antes de usar esta política. Caso contrário, você pode perder a capacidade de acessar seu bucket.

```
{  
    "Id": "PolicyId2",  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowIPmix",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Resource": [  
                "arn:aws:s3::::DOC-EXAMPLE-BUCKET",  
                "arn:aws:s3::::DOC-EXAMPLE-BUCKET/*"  
            ],  
            "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 tenha um site com nome de domínio (www.example.com ou example.com) com links para fotos e vídeos armazenados no bucket do Amazon S3, **DOC-EXAMPLE-BUCKET**. Por padrão, todos os recursos do Amazon S3 são privados e, 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 de condição `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", "s3:GetObjectVersion"],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
            "Condition": {
                "StringLike": {"aws:Referer": ["http://www.example.com/\*", "http://example.com/\*"]}
            }
        }
    ]
}
```

Certifique-se de que os navegadores que você usa incluem o cabeçalho HTTP `referer` na solicitação.

Warning

Recomendamos cuidado ao usar a chave de condição `aws:Referer`. É perigoso incluir um valor de cabeçalho do indicador conhecido publicamente. Partes não autorizadas podem usar navegadores personalizados ou modificados para fornecer qualquer valor de `aws:Referer` que escolherem. Assim sendo, não use `aws:Referer` para impedir que partes não autorizadas façam solicitações diretas à AWS.

A chave de condição `aws:Referer` é oferecida apenas para permitir que os clientes protejam o conteúdo digital, como o conteúdo armazenado no Simple Storage Service (Amazon S3), evitando que seja referenciado em sites de terceiros não autorizados. Para obter mais informações, consulte [aws:Referer](#) no Guia do usuário do IAM.

Conceder permissão para uma OAI do Amazon CloudFront

A seguinte política de bucket de exemplo concede uma permissão de identidade de acesso de origem (OAI – Origin Access Identity) do CloudFront para obter (ler) todos os objetos em seu bucket do Amazon S3. Você pode usar uma OAI do CloudFront para permitir que os usuários acessem objetos em seu bucket por meio do CloudFront, mas não diretamente pelo Amazon S3. Para obter mais informações, consulte [Restringir acesso ao conteúdo do Amazon S3 usando uma identidade de acesso de origem](#) no Guia do desenvolvedor do Amazon CloudFront.

A política a seguir usa o ID da OAI como Principal da política. Para obter mais informações sobre o uso de políticas de bucket do S3 para conceder acesso a uma OAI do CloudFront, consulte [Usar políticas de bucket do Amazon S3](#) no Guia do desenvolvedor do Amazon CloudFront.

Para usar este exemplo:

- Substitua ***EH1HDMB1FH2TC*** pelo ID da OAI. Para localizar o ID da OAI, consulte a [página Origin Access Identity \(Identidade de acesso de origem\)](#) no console do CloudFront ou use `ListCloudFrontOriginAccessIdentities` na API do CloudFront.
- Substitua ***DOC-EXAMPLE-BUCKET*** pelo nome do bucket do Amazon S3.

```
{  
    "Version": "2012-10-17",  
    "Id": "PolicyForCloudFrontPrivateContent",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity EH1HDMB1FH2TC"  
            },  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"  
        }  
    ]  
}
```

Adicionar uma política de bucket para exigir MFA

O Amazon S3 oferece suporte a acesso à API protegido por MFA, um recurso que pode impor a autenticação multifator 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 [Autenticação multifator da AWS](#). É possível exigir a autenticação MFA para todas as solicitações de acesso a seus recursos do Amazon S3.

Você pode impor o requisito MFA usando a chave `aws:MultiFactorAuthAge` em uma política de bucket. AWS Identity and Access Management Os usuários do IAM podem acessar os recursos do Amazon S3 usando credenciais temporárias emitidas pelo AWS Security Token Service (AWS STS). Forneça o código da MFA no momento da solicitação do AWS STS.

Quando o Amazon S3 recebe uma solicitação com autenticação multifator, 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. O exemplo de política nega qualquer operação do Simple Storage Service (Amazon S3) na pasta `/taxdocuments` do bucket ***DOC-EXAMPLE-BUCKET*** se a solicitação não for autenticada usando MFA. Para saber mais sobre MFA, consulte [Uso da autenticação multifator \(MFA\) na AWS](#) no Guia do usuário do IAM.

```
{  
    "Version": "2012-10-17",  
    "Id": "123",  
    "Statement": [  
        {  
            "Sid": "",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Condition": {  
                "StringNotEquals": {  
                    "aws:MultiFactorAuthAge": "0"  
                }  
            }  
        }  
    ]  
}
```

```
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/taxdocuments/*",
        "Condition": { "Null": { "aws:MultiFactorAuthAge": true } }
    }
]
```

A condição Null no bloco Condition é avaliada como true a se o valor da chave aws:MultiFactorAuthAge for nulo, o que indica que as credenciais de segurança temporárias da 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 instrução concede a permissão s3:GetObject em um bucket (**DOC-EXAMPLE-BUCKET**) a todos. Outra instrução restringe ainda mais o acesso à pasta **DOC-EXAMPLE-BUCKET/taxdocuments** no bucket ao exigir MFA.

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/taxdocuments/*",
            "Condition": { "Null": { "aws:MultiFactorAuthAge": true } }
        },
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": "*",
            "Action": ["s3:GetObject"],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        }
    ]
}
```

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:::DOC-EXAMPLE-BUCKET/taxdocuments/*",
            "Condition": { "Null": { "aws:MultiFactorAuthAge": true } }
        },
        {
            "Sid": "",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/taxdocuments/*",
            "Condition": { "NumericLessThan": { "aws:MultiFactorAuthAge": 3600 } }
        }
    ]
}
```

```
        "Condition": {"NumericGreaterThan": {"aws:MultiFactorAuthAge": 3600 }},
    },
    {
        "Sid": "",
        "Effect": "Allow",
        "Principal": "*",
        "Action": ["s3:GetObject"],
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    }
]
```

Conceder permissões entre contas para fazer upload de objetos garantindo que o proprietário do bucket tenha controle total

O exemplo a seguir mostra como permitir que outra Conta da AWS carregue objetos no bucket e, ao mesmo tempo, obter controle total sobre os objetos carregados. Essa política impõe que só seja concedida a uma Conta da AWS específica ([111122223333](#)) a capacidade de carregar objetos se essa conta incluir a ACL pré-configurada de controle total do proprietário do bucket no carregamento. A condição `StringEquals` da política especifica a chave `s3:x-amz-acl` condition para expressar essa exigência (consulte [Exemplos de chave de condição do Amazon S3 \(p. 447\)](#)).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PolicyForAllowUploadWithACL",
            "Effect": "Allow",
            "Principal": {"AWS": "111122223333" },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
            "Condition": {
                "StringEquals": {"s3:x-amz-acl": "bucket-owner-full-control"}
            }
        }
    ]
}
```

Conceder permissões para o Amazon S3 Inventory e a análise do Simple Storage Service (Amazon S3)

O Amazon S3 Inventory cria listas dos objetos em um bucket do Amazon 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 Amazon S3 e ao configurar a exportação da análise. Para obter mais informações, consulte [Inventário do Amazon S3 \(p. 786\)](#) e [Análise do Amazon S3 – Análise de classe de armazenamento \(p. 1116\)](#).

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. Use uma política de bucket como esta no bucket de destino ao configurar o Amazon S3 Inventory 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:::destinationbucket/*"
        ],
        "Condition": {
            "ArnLike": {
                "aws:SourceArn": "arn:aws:s3:::sourcebucket"
            },
            "StringEquals": {
                "aws:SourceAccount": "111122223333",
                "s3:x-amz-acl": "bucket-owner-full-control"
            }
        }
    }
}
```

Conceder permissões para o Amazon S3 Storage Lens

O Amazon S3 Storage Lens agrega suas métricas de uso e atividade e exibe as informações no snapshot da conta na página inicial do console (Buckets) do Amazon S3, em painéis interativos ou por meio de uma exportação de métricas que você pode obter por download no formato CSV ou Parquet. Você pode usar o painel para visualizar insights e tendências, sinalizar discrepâncias e receber recomendações para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados. Você pode usar o S3 Storage Lens por meio do AWS Management Console, da AWS CLI, de AWS SDKs ou da API REST.

O S3 Storage Lens pode agregar seu uso de armazenamento às exportações de métricas em um bucket do Amazon S3 para análise posterior. O bucket em que o S3 Storage Lens coloca as exportações de métricas dele é conhecido como bucket de destino. Você deve ter uma política de buckets para o bucket de destino quando configurar a exportação de métricas do S3 Storage Lens. Para obter mais informações, consulte [Avaliação de sua atividade de armazenamento e uso com o Amazon S3 Storage Lens \(p. 1121\)](#).

O exemplo de política de bucket a seguir concede ao Amazon S3 permissão para gravar objetos (PUTs) em um bucket de destino. Você usa uma política de bucket como essa no bucket de destino ao configurar uma exportação de métricas do S3 Storage Lens.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3StorageLensExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "storage-lens.s3.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::destination-bucket/destination-prefix/StorageLens/111122223333/*"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": "111122223333",
                    "aws:SourceArn": "arn:aws:s3:Região da AWS:111122223333:storage-lens/storage-lens-dashboard-configuration-id"
                }
            }
        }
    ]
}
```

Use a modificação a seguir no recurso de política de bucket anterior ao configurar uma exportação de métricas no nível da organização do S3 Storage Lens.

```
"Resource": "arn:aws:s3:::destination-bucket/destination-prefix/StorageLens/your-organization-id/*",
```

Uso de políticas de usuário do IAM

Você pode criar e configurar políticas de usuário do IAM para controlar o acesso do usuário ao Amazon S3. As políticas de usuário usam linguagem de política de acesso baseada em JSON.

Esta seção mostra diversas políticas de usuário do IAM para controle de acesso de usuário ao Amazon S3. Por exemplo, políticas de bucket, consulte [Uso de políticas de bucket \(p. 515\)](#). Para obter informações sobre a linguagem de políticas de acesso, consulte [Políticas e permissões no Amazon S3 \(p. 437\)](#).

Tópicos

- [Controlar o acesso a um bucket com políticas de usuário \(p. 528\)](#)
- [Exemplos de política de usuário \(p. 545\)](#)

Controlar o acesso a um bucket com políticas de usuário

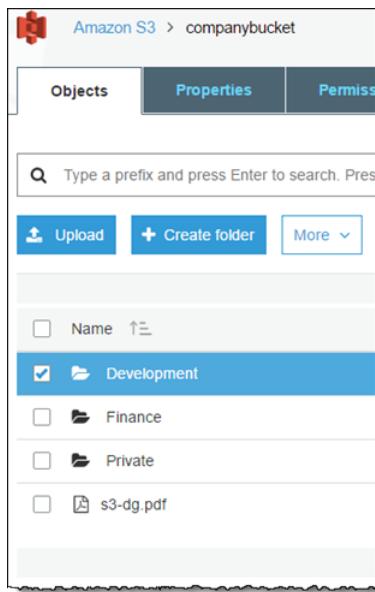
Esta demonstração explica como as permissões de usuário funcionam com o Amazon S3. Nesse exemplo, você cria um bucket com pastas. Em seguida, crie usuários do AWS Identity and Access Management (IAM) na sua Conta da AWS e conceda a eles permissões incrementais no bucket do Amazon S3 e nas pastas dele.

Tópicos

- [Elementos básicos de buckets e pastas \(p. 528\)](#)
- [Resumo da demonstração \(p. 530\)](#)
- [Preparar para a demonstração \(p. 531\)](#)
- [Etapa 1: Criar um bucket \(p. 531\)](#)
- [Etapa 2: criar usuários do IAM e um grupo \(p. 532\)](#)
- [Etapa 3: verificar se os usuários do IAM não têm nenhuma permissão \(p. 532\)](#)
- [Etapa 4: Conceder permissões no nível do grupo \(p. 533\)](#)
- [Etapa 5: conceder permissões específicas do usuário do IAM Alice \(p. 539\)](#)
- [Etapa 6: conceder permissões específicas do usuário do IAM Bob \(p. 543\)](#)
- [Etapa 7: Proteger a pasta Private \(p. 543\)](#)
- [Etapa 8: Limpar \(p. 545\)](#)
- [Recursos relacionados \(p. 545\)](#)

Elementos básicos de buckets e pastas

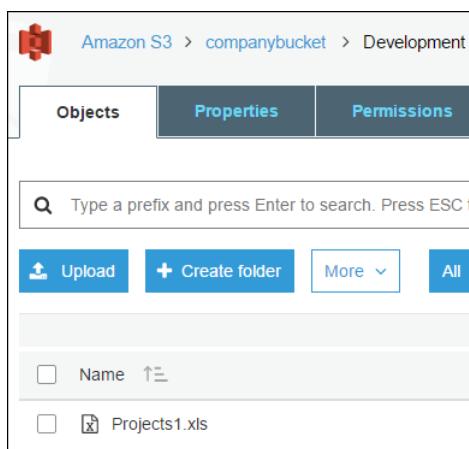
O modelo de dados do Amazon S3 é uma estrutura plana: você cria um bucket e o bucket armazena objetos. Não há hierarquia de buckets ou de subpastas, mas você pode emular uma hierarquia de pastas. Ferramentas como o console do Amazon S3 podem apresentar uma exibição dessas pastas e subpastas lógicas em seu bucket, como mostrado na imagem abaixo:



O console mostra que o bucket nomeado 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 (/) no final.
- Quando você faz upload de um objeto chamado Projects1.xls na pasta Development, o console sobe o objeto e concede a ele 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 abre a pasta Development, o console lista os objetos nela. No exemplo a seguir, a pasta Development contém um objeto.



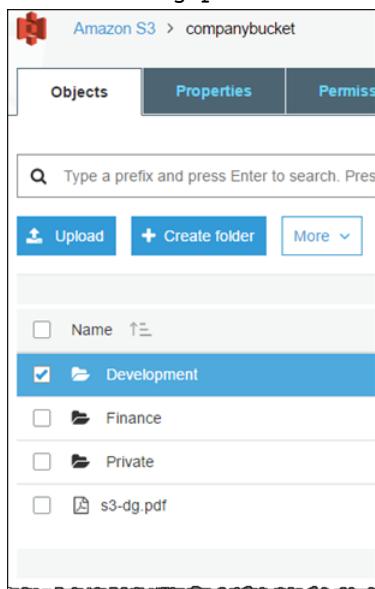
Quando o console lista a pasta **Development** no bucket **companybucket**, ele envia uma solicitação para o Amazon S3 na qual especifica o prefixo **Development** e o delimitador / na solicitação. A resposta do console parece uma lista de pastas no sistema de arquivos de seu computador. O exemplo anterior mostra que o bucket **companybucket** tem um objeto com a chave **Development/Projects1.xls**.

O console está usando chaves de objeto para inferir uma hierarquia lógica. O Amazon S3 não tem hierarquia física; ele só tem buckets que contêm objetos em uma estrutura de arquivos simples. Ao criar objetos usando a API do Amazon S3, você pode usar chaves de objeto que implicam uma hierarquia lógica. Ao criar uma hierarquia lógica de objetos, você pode gerenciar o acesso a pastas individuais, como este passo a passo ensina.

Antes de começar, você precisa conhecer o conceito de conteúdo no nível raiz do bucket. 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 objeto no nível raiz. Quando você escolhe o nome do bucket no console do Amazon S3, os itens no nível raiz são exibidos como mostrado na imagem abaixo. O console exibe os prefixes no nível superior (**Private/**, **Development/** e **Finance/**) como pastas no nível raiz. A chave de **s3-dg.pdf** não tem um prefixo, então, ela aparece como um item no nível raiz.



Resumo da demonstração

Neste passo a passo, você cria um bucket com três pastas (**Private**, **Development** e **Finance**).

Você tem dois usuários, Alice e Bob. Você quer que a Alice tenha acesso somente à pasta `Development`, e Bob tenha acesso à `Finance`. Você quer manter o conteúdo da pasta `Private` privado. No passo a passo, você gerencia o acesso criando usuários do IAM (o exemplo usa 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. Neste exercício, Alice e Bob precisam ter algumas permissões comuns. Então, você também cria um grupo chamado `Consultants` e adiciona Alice e Bob a ele. Primeiro, você concede permissões anexando uma política de grupo ao grupo. Depois, adiciona permissões específicas do usuário anexando políticas a usuários específicos.

Note

O passo a passo usa `companybucket` como o nome do bucket, Alice e Bob como os usuários do `Consultants`, e IAM como o nome do grupo. Como o Amazon S3 exige que os nomes de bucket sejam globalmente exclusivos, você precisará substituir o nome do bucket pelo nome que criar.

Preparar para a demonstração

Neste exemplo, use suas credenciais da Conta da AWS para criar usuários do IAM. Inicialmente, esses usuários não têm nenhuma permissão. Você concede gradualmente a esses usuários as permissões para executar ações específicas do Amazon S3. Para testar essas permissões, você entra no console com as credenciais de cada usuário. À medida que concede 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 o teste em apenas um navegador, mas o processo é mais rápido se você usar dois navegadores diferentes. AWS Management Console Use um navegador para se conectar ao seu console com suas credenciais da Conta da AWS e outro para se conectar com as credenciais de usuário do IAM.

Para fazer login no AWS Management Console com as credenciais da sua Conta da AWS, acesse <https://console.aws.amazon.com/>. Um usuário do IAM não pode fazer login com 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, consulte a página sobre login no AWS Management Console no Guia do usuário do IAM.

Para fornecer um link de login para usuários do IAM

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel Navigation (Navegação), escolha IAM Dashboard (Painel do IAM).
3. Observe o URL em IAM users sign in link: (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 faz login no console do Amazon S3 com suas credenciais da Conta da AWS, cria um bucket, adiciona pastas (`Development`, `Finance` e `Private`) ao bucket e carrega um ou dois documentos de exemplo em cada pasta.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Crie um bucket.
Para obter instruções detalhadas, consulte [Criação de um bucket \(p. 129\)](#).
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 nomeadas `Private`, `Finance` e `Development` ao bucket.

Para obter instruções detalhadas para criar uma pasta, consulte [Organizar objetos no console do Amazon S3 usando pastas \(p. 269\)](#) no Manual do usuário do Amazon Simple Storage Service.

5. Faça upload de um ou dois documentos em cada pasta.

Neste exercício, suponha que você tenha carregado alguns documentos em cada pasta, resultando que o bucket tenha 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 detalhadas, consulte [Fazer upload de objetos \(p. 171\)](#).

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 `Consultants` e adicione os dois usuários a ele.

Warning

Quando você adicionar usuários e um grupo, não anexe políticas que concedem permissões a esses usuários. No início, esses usuários não terão permissões. Nas próximas seções, você concederá permissões incrementalmente. Primeiro verifique se atribuiu senhas a esses usuários do IAM. Você utilizará essas credenciais de usuário para testar as ações do Amazon S3 e verificar se as permissões funcionam como esperado.

Para obter instruções passo a passo para criar um novo usuário do IAM, consulte [Criação de um usuário do IAM em sua Conta da AWS](#) no Guia do usuário do IAM. Ao criar usuários seguindo este passo a passo, selecione AWS Management Console access (Acesso ao console) e desmarque Programmatic access (Acesso programático).

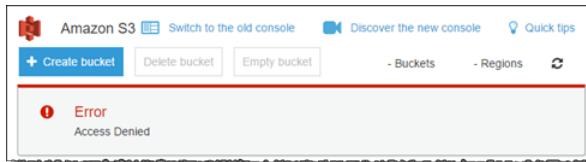
Para obter instruções passo a passo para criar um grupo administrativo, consulte [Criar seu primeiro usuário administrador e grupo do IAM](#) no Guia do usuário do IAM.

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 \(p. 531\)](#)), entre no AWS Management Console 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.



Agora, você pode começar a conceder permissões incrementais aos usuários. Antes, você anexará uma política de grupo concedendo as permissões de que os dois usuários precisam.

Etapa 4: Conceder permissões no nível do grupo

Você quer que 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.

Primeiro, você cria uma política que concede essas permissões e, depois, anexa-a ao grupo Consultants.

Etapa 4.1: Conceder permissão para listar todos os buckets

Nesta etapa, você cria uma política gerenciada que concede aos usuários as permissões mínimas para listar todos os buckets de propriedade da conta pai. Depois, anexa a política ao grupo Consultants. Ao anexar a política gerenciada a um usuário ou um grupo, você concede ao usuário ou grupo a permissão para obter uma lista de buckets de propriedade da Conta da AWS pai.

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.

Note

Como você está concedendo permissões de usuário, faça login com suas credenciais da Conta da AWS, e não como um usuário do IAM.

2. Crie a política gerenciada.
 - a. No painel de navegação à esquerda, escolha Policies (Políticas) e Create Policy (Criar política).
 - b. Selecione a 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 uma descrevendo uma permissão usando uma coleção de pares de valores de nome. A política anterior descreve uma permissão específica. A Action especifica o tipo de acesso. Na

política, `s3:ListAllMyBuckets` é uma ação do Amazon S3 predefinida. Esta ação abrange a operação GET serviço do Amazon S3, que retorna uma lista de todos os buckets no remetente autenticado. O valor do elemento `Effect` determina se a permissão específica é permitida ou negada.

- d. Escolha Review Policy (Revisar política). Na próxima página, insira `AllowGroupToSeeBucketListInTheConsole` no campo Name (Nome) e escolha em Create policy (Criar política).

Note

A entrada Summary (Resumo) exibe uma mensagem afirmando que a política não concede permissão nenhuma. Para esta apresentação, você pode, de maneira segura, ignorar essa mensagem.

3. Anexe a política gerenciada `AllowGroupToSeeBucketListInTheConsole` que você criou para o grupo `Consultants`.

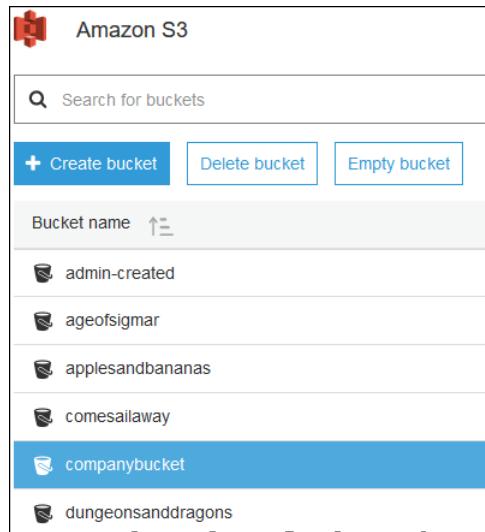
Para obter instruções passo a passo para anexar uma política gerenciada, consulte [Adicionar e remover permissões de identidade do IAM](#) no Guia do usuário do IAM.

Anexe documentos de política a usuários e grupos do IAM no console do IAM. Como você quer que os dois usuários possam listar os buckets, você anexa a política ao grupo.

4. Teste a permissão.

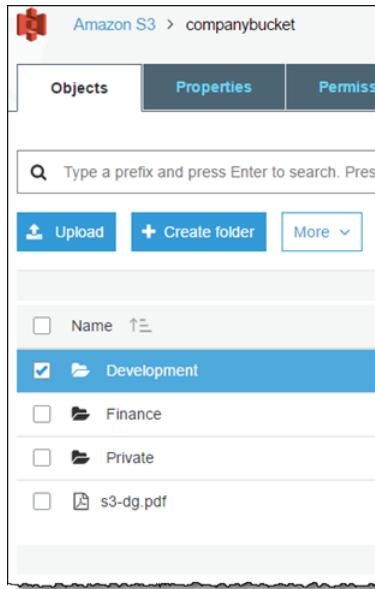
- a. Usando o link de login de usuário do IAM (consulte [Para fornecer um link de login para usuários do IAM \(p. 531\)](#)), entre no console usando qualquer uma das credenciais de usuário do IAM.
- b. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.

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, você permite que todos os usuários no grupo `Consultants` listem itens no nível raiz do bucket `companybucket`. Quando um usuário escolhe o bucket da empresa no console do Amazon S3, ele pode visualizar os itens no nível raiz do bucket.



Note

Esse exemplo usa `companybucket` como ilustração. Use o nome do bucket que você criou.

Para entender a solicitação que o console envia ao Amazon S3 quando você escolhe o nome de um bucket, a resposta que o Amazon S3 retorna e como o console a interpreta, é necessário entendê-la um pouco mais.

Quando você clica no nome de um bucket, o console envia a solicitação `GET Bucket` ao Amazon S3. Essa solicitação inclui os parâmetros a seguir:

- O parâmetro `prefix` com uma string vazia como valor.
- O parâmetro `delimiter` com `/` como 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=
```

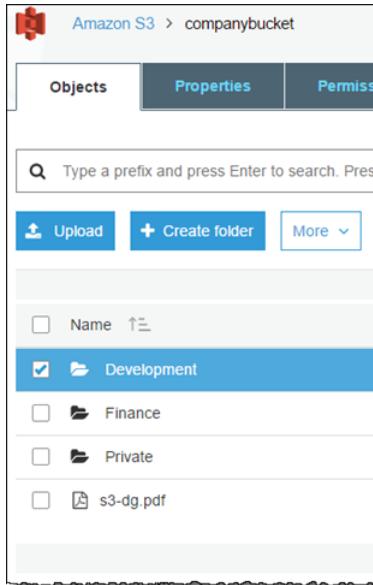
O Amazon S3 retorna uma resposta que inclui o seguinte elemento `<ListBucketResult>`.

```
<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>
```

O objeto s3-dg.pdf da chave 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.



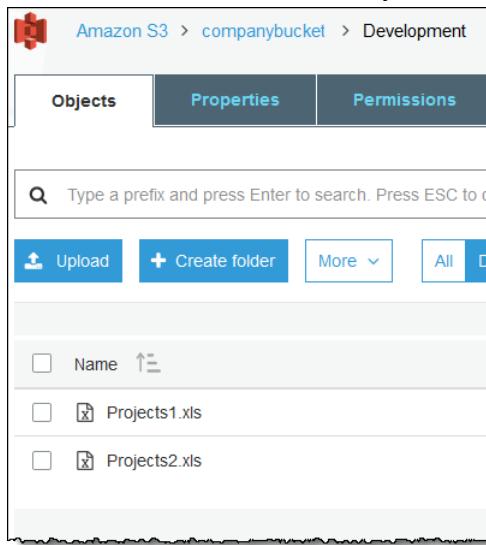
Se Bob ou Alice abrirem a pasta Development (Desenvolvimento), o console enviará a solicitação [GET Bucket \(Listar objetos\)](#) ao Amazon S3 com os parâmetros prefix e delimiter definidos como os seguintes valores:

- O parâmetro prefix com valor Development/.
- O parâmetro delimiter com 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, volte a conceder aos usuários a permissão para listar itens no nível raiz do bucket. 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, você adiciona uma condição de que os usuários devem especificar um prefixo vazio na solicitação, isto é, eles não podem clicar duas vezes em nenhuma das pastas no nível raiz. Finalmente, você adiciona uma condição para solicitar o acesso de pasta exigindo que as solicitações de usuário incluam o parâmetro `delimiter` com valor `/`.

```
{  
    "Sid": "AllowRootLevelListingOfCompanyBucket",  
    "Action": ["s3>ListBucket"],  
    "Effect": "Allow",  
    "Resource": ["arn:aws:s3:::companybucket"],  
    "Condition":{  
        "StringEquals":{  
            "s3:prefix":[""], "s3:delimiter":["/"]  
        }  
    }  
}
```

Quando você escolhe o bucket no console do Amazon S3, o console primeiro envia a solicitação [GET Bucket location](#) para encontrar a Região da AWS em que o bucket está implantado. Depois, o console usa o endpoint específico da região para o bucket enviar a solicitação [GET Bucket \(listar objetos\)](#). Como resultado, se os usuários forem usar 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

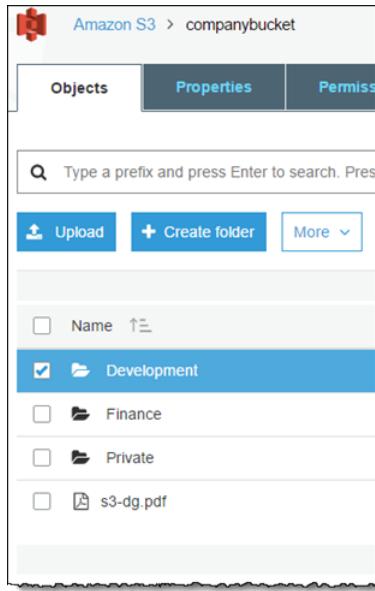
1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.

- Use as credenciais da sua Conta da AWS, e não as credenciais de um usuário do IAM, para fazer login no console.
2. Substitua a política gerenciada existente `AllowGroupToSeeBucketListInTheConsole` que está anexada ao grupo `Consultants` 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 do IAM](#) no Guia do usuário do IAM. Ao seguir as instruções detalhadas, siga as etapas para aplicar as 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 \(p. 531\)](#)), acesse o AWS Management Console.
- Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
- b. Escolha o bucket que você criou, e o console mostrará os itens no nível raiz do bucket. Se você escolher alguma pasta no bucket, não verá o conteúdo dela, porque ainda não concedeu essas permissões.



Esse teste é bem-sucedido quando os usuários usam o console do Amazon S3. Ao escolher 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 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 muita coisa. Agora, conceda as permissões específicas do usuário da seguinte maneira:

- Permite que a Alice obtenha e coloque objetos na pasta `Development`.
- Permite que o 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, conceda permissão para a 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 você concede permissões adicionais a Alice para que ela possa ver o conteúdo da pasta `Development` e obter e colocar objetos nela.

Etapa 5.1: conceder permissão do usuário do IAM Alice para listar o conteúdo da pasta `Development`

Para que a Alice possa listar o conteúdo da pasta `Development`, você deve aplicar uma política ao usuário Alice concedendo permissão para a ação `s3>ListBucket` no bucket `companybucket`, contanto que a solicitação inclua o prefixo `Development/`. Você quer que essa política seja aplicada somente ao

usuário Alice, então, usa uma política em linha. Para obter mais informações sobre políticas em linha, consulte [Políticas gerenciadas e políticas em linha](#) no Guia do usuário do IAM.

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.

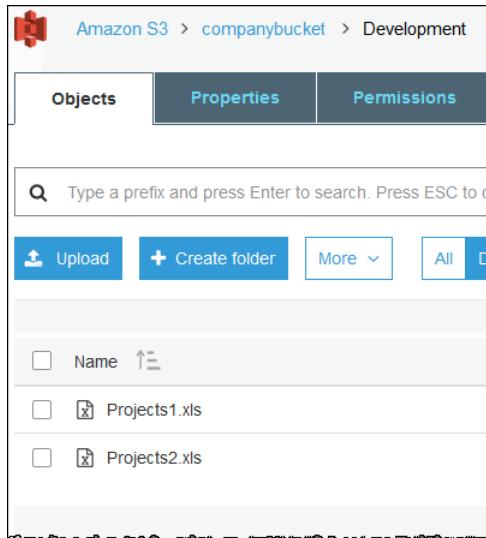
Use as credenciais da sua Conta da AWS, e não as credenciais de um usuário do IAM, para fazer login no console.

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, escolha **Uses (Usuários)**.
 - b. Clique no nome de usuário Alice.
 - c. Na página de detalhes do usuário, selecione a guia **Permissions (Permissões)** e escolha **Add inline policy (Adicionar política em linha)**.
 - d. Selecione a guia **JSON**.
 - e. Copie a política a seguir e cole-a no campo de 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. Escolha **Review Policy (Revisar política)**. Na próxima página, insira um nome no campo **Name (Nome)** e escolha **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 \(p. 531\)](#)), acesse o AWS Management Console.
 - 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 escolhe a 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 exibe a lista.



Etapa 5.2: conceder permissões do usuário do IAM Alice para obter e colocar objetos na pasta Development

Para que a Alice possa obter e colocar 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 AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.

Use as credenciais da sua Conta da AWS, e não as credenciais de um usuário do IAM, para fazer login no console.

2. Edite a política em linha que você criou na etapa anterior.
 - a. No painel de navegação à esquerda, escolha Users (Usuários).
 - b. Clique no nome de usuário Alice.
 - c. Na página de detalhes do usuário, selecione a guia Permissions (Permissões) e depois expanda a seção Inline Policies (Políticas em linha).
 - d. Escolha Edit Policy (Editar política) ao lado do nome da política que você criou na etapa anterior.
 - e. Copie a política a seguir e cole no campo do 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 \(p. 531\)](#)), acesse o AWS Management Console.
- 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 de usuário Alice que exige que todas as solicitações que Alice envia 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}
    }
}
```

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* é classificado como verdadeiro, os usuários não podem 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 será classificada como verdadeiro e, portanto, a *Condition* inteira será classificada como verdadeira. Você deve permitir uma

string vazia como o valor do parâmetro `prefix`. Com a discussão anterior, lembre-se 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 \(p. 534\)](#).

- 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 que você criou para o usuário Alice.

Copie a política a seguir e cole no campo do 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"],  
            "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 ao 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 \(p. 539\)](#).

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, você não quer que nenhum dos usuários neste exemplo acessasse o conteúdo da pasta `Private`. Como garantir que você não vai conceder acidentalmente uma permissão ao 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 abrir a pasta **Private**, essa política fará o 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 **Consultants** 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** no seu bucket.

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

Use as credenciais da sua Conta da AWS, e não as credenciais de um usuário do IAM, para fazer login no console.

2. Substitua a política gerenciada `AllowGroupToSeeBucketListInTheConsole` existente que está anexada ao grupo **Consultants** 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": {  
                "StringLike": {"s3:prefix": ["Private/*"]} }  
        }  
    ]  
}
```

```
        "Condition":{  
            "StringEquals":{"s3:prefix":[""]}  
        }  
    },  
    {  
        "Sid": "RequireFolderStyleList",  
        "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/"]}  
        }  
    }  
]
```

Etapa 8: Limpar

Para limpar, abra o console do IAM e remova os usuários Alice e Bob. Para obter instruções passo a passo, consulte [Excluir 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 neste exercício.

Recursos relacionados

- [Gerenciar políticas do IAM](#) no Guia do usuário do IAM.

Exemplos de política de usuário

Esta seção mostra diversos exemplos de políticas de usuário do AWS Identity and Access Management (IAM) para controle de acesso de usuários ao Amazon S3. Por exemplo, políticas de bucket, consulte [Uso de políticas de bucket \(p. 515\)](#). Para obter informações sobre a linguagem de políticas do IAM, consulte [Políticas de bucket e políticas de usuário \(p. 437\)](#).

Os exemplos de política a seguir funcionarão se você testá-los de forma programática. No entanto, para usá-los com o console do Amazon S3, você precisará conceder permissões adicionais solicitadas pelo console. Para obter informações sobre o uso de políticas como essas com o console do Amazon S3, consulte [Controlar o acesso a um bucket com políticas de usuário \(p. 528\)](#).

Tópicos

- [Permitir que um usuário do IAM acesse um dos seus buckets \(p. 546\)](#)
- [Permitir que cada usuário do IAM acesse uma pasta em um bucket \(p. 546\)](#)
- [Permitir que um grupo tenha uma pasta compartilhada no Amazon S3 \(p. 549\)](#)

- Permitir que todos os seus usuários leiam objetos em uma parte de um bucket (p. 549)
- Permitir que um parceiro solte arquivos em uma parte específica de um bucket (p. 550)
- Restringir o acesso a buckets do Amazon S3 em uma Conta da AWS específica (p. 550)
- Restringir o acesso a buckets do Amazon S3 em sua unidade organizacional (UO) (p. 551)
- Restringir o acesso a buckets do Amazon S3 em sua unidade organizacional (p. 552)

Permitir que um usuário do IAM acesse um dos seus buckets

Neste exemplo, você deseja conceder a um usuário do IAM em sua Conta da AWS acesso a um de seus buckets, [DOC-EXAMPLE-BUCKET1](#), e permitir que ele 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 [Controlar o acesso a um bucket com políticas de usuário \(p. 528\)](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "s3>ListAllMyBuckets",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": ["s3>ListBucket", "s3:GetBucketLocation"],  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject",  
                "s3:PutObjectAcl",  
                "s3:GetObject",  
                "s3:GetObjectAcl",  
                "s3>DeleteObject"  
            ],  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*"  
        }  
    ]  
}
```

Permitir que cada usuário do IAM acesse uma pasta em um bucket

Neste exemplo, você deseja que dois usuários do IAM, Alice e Carlos, tenham acesso ao seu bucket, [DOC-EXAMPLE-BUCKET1](#), para que possam adicionar, atualizar e excluir objetos. No entanto, você quer restringir o acesso de todos os usuários a um único prefixo (pasta) no bucket. É recomendável criar pastas com nomes correspondentes aos nomes dos usuários.

```
DOC-EXAMPLE-BUCKET1  
Mary/  
Carlos/
```

Para conceder a cada usuário acesso apenas à pasta dele, crie uma política para cada um deles e anexe-a individualmente. Por exemplo, você pode anexar a política a seguir ao usuário Alice para conceder a ela permissões específicas do Amazon S3 na pasta [DOC-EXAMPLE-BUCKET1/Mary](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject",  
                "s3:GetObject",  
                "s3:GetObjectVersion",  
                "s3:DeleteObject",  
                "s3:DeleteObjectVersion"  
            ],  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/Mary/*"  
        }  
    ]  
}
```

Você pode anexar uma política semelhante ao usuário Carlos, especificando a pasta **Carlos** no valor **Resource**.

Em vez de anexar políticas a usuários individuais, você pode criar uma única política que use uma variável de política e anexá-la a um grupo. Primeiro, você precisará criar um grupo e adicionar os usuários Alice e Carlos a ele. O exemplo de política a seguir concede um conjunto de permissões do Amazon S3 na pasta **DOC-EXAMPLE-BUCKET1**/\${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 **DOC-EXAMPLE-BUCKET1/Mary**.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject",  
                "s3:GetObject",  
                "s3:GetObjectVersion",  
                "s3:DeleteObject",  
                "s3:DeleteObjectVersion"  
            ],  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/${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 do IAM, 2008-10-17, não é compatível com variáveis de política.

Se quiser testar a política anterior no console do Amazon S3, o console exigirá permissões adicionais, conforme exibido na política a seguir. Para obter informações sobre como o console usa essas permissões, consulte [Controlar o acesso a um bucket com políticas de usuário](#) (p. 528).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowGroupToSeeBucketListInTheConsole",  
            "Effect": "Allow",  
            "Action": "s3:ListBucket",  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"  
        }  
    ]  
}
```

```

    "Action": [
        "s3>ListAllMyBuckets",
        "s3:GetBucketLocation"
    ],
    "Effect": "Allow",
    "Resource": "arn:aws:s3:::/*"
},
{
    "Sid": "AllowRootLevelListingOfTheBucket",
    "Action": "s3>ListBucket",
    "Effect": "Allow",
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1",
    "Condition": {
        "StringEquals": {
            "s3:prefix": [""], "s3:delimiter": ["/"]
        }
    }
},
{
    "Sid": "AllowListBucketOfASpecificUserPrefix",
    "Action": "s3>ListBucket",
    "Effect": "Allow",
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1",
    "Condition": { "StringLike": {"s3:prefix": ["${aws:username}/*"] } }
},
{
    "Sid": "AllowUserSpecificActionsOnlyInTheSpecificUserPrefix",
    "Effect": "Allow",
    "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3>DeleteObject",
        "s3>DeleteObjectVersion"
    ],
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/${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 a chave de objeto (nome do objeto) incluir um \$.

Para evitar esse conflito, especifique o caractere \$ usando \${\$}. Por exemplo, para incluir a chave de objeto my\$file em uma política, especifique-a como my\${\$}file.

Embora os nomes de usuários do IAM sejam identificadores amigáveis e legíveis, eles não precisam ser globalmente exclusivos. Por exemplo, se o usuário Carlos deixar a empresa e outro Carlos ingressar, este poderá acessar as informações do Carlos anterior.

Em vez de usar nomes de usuário, você pode criar pastas baseadas nos IDs de usuário do IAM. O ID de cada usuário do IAM é exclusivo. 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:::DOC-EXAMPLE-BUCKET1/home/${aws:userid}/*"
}
]
```

Permitir que usuários que não são do IAM (usuários de aplicativos móveis) acessem 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 deseja limitar o acesso de todos os usuários à sua própria pasta, mas não pode criar pastas antes que alguém baixe seu aplicativo e comece a jogar, pois não tem o ID do 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 no 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 permitam 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.

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: **DOC-EXAMPLE-BUCKET1**/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:::DOC-EXAMPLE-BUCKET1/share/marketing/*"
        }
    ]
}
```

Permitir que todos os seus usuários leiam objetos em uma parte de um bucket

Neste exemplo, você cria 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 **DOC-EXAMPLE-BUCKET1**/readonly.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetObject",  
                "s3:GetObjectVersion"  
            ],  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/readonly/*"  
        }  
    ]  
}
```

Permitir que um parceiro solte arquivos em uma parte específica de um bucket

Neste exemplo, crie um grupo chamado **AnyCompany** que representa uma empresa parceira. Crie um usuário do IAM para a pessoa ou aplicação específica na empresa parceira que precisa de acesso. Depois, coloque o usuário no grupo.

Depois, anexe uma política que conceda ao grupo PutObject acesso à seguinte pasta em um bucket:

DOC-EXAMPLE-BUCKET1/uploads/anycompany

É preciso impedir que o grupo **AnyCompany** faça qualquer outra coisa no bucket e, portanto, você adiciona uma declaração negando explicitamente outras permissões a ações do Amazon S3, exceto PutObject, em qualquer recurso do Amazon S3 na Conta da AWS.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/uploads/anycompany/*"  
        },  
        {  
            "Effect": "Deny",  
            "Action": "s3:*",  
            "NotResource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/uploads/anycompany/*"  
        }  
    ]  
}
```

Restringir o acesso a buckets do Amazon S3 em uma Conta da AWS específica

Se quiser garantir que suas entidades principais do Amazon S3 acessem apenas os recursos que estão em uma Conta da AWS confiável, você poderá restringir o acesso. Por exemplo, esta [política do IAM baseada em identidade](#) usa um efeito Deny para bloquear o acesso às ações do Amazon S3, a menos que o recurso do Amazon S3 que está sendo acessado esteja na conta **222222222222**. Para impedir que uma entidade principal do IAM em uma Conta da AWS acesse objetos do Amazon S3 fora da conta, anexe a seguinte política do IAM:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "DenyS3AccessOutsideMyBoundary",  
            "Effect": "Deny",  
            "Action": [  
                "s3:*"
```

```
],
"Resource": "*",
"Condition": {
    "StringNotEquals": {
        "aws:ResourceAccount": [
            "222222222222"
        ]
    }
}
]
```

Note

Essa política não substitui seus controles de acesso do IAM existentes, pois não concede nenhum acesso. Em vez disso, essa política funciona como uma barreira de proteção adicional para suas outras permissões do IAM, independentemente das permissões concedidas por meio de outras políticas do IAM.

Substitua o ID da conta **222222222222** na política pela sua própria Conta da AWS. Para aplicar uma política a várias contas e ainda manter essa restrição, substitua o ID da conta pela chave de condição `aws:PrincipalAccount`. Essa condição exige que a entidade principal e o recurso estejam na mesma conta.

Restringir o acesso a buckets do Amazon S3 em sua unidade organizacional (UO)

Se você tiver uma [unidade organizacional \(UO\)](#) configurada no AWS Organizations, é recomendável restringir o acesso ao bucket do Amazon S3 a uma parte específica de sua organização. Neste exemplo, usaremos a chave `aws:ResourceOrgPaths` para restringir o acesso ao bucket do Amazon S3 a uma UO em sua organização. Neste exemplo, o [ID da UO](#) é **ou-acroot-exampleou**. Substitua esse valor em sua própria política por seus próprios IDs de UO.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowS3AccessOutsideMyBoundary",
            "Effect": "Allow",
            "Action": [
                "s3:*"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringNotLike": {
                    "aws:ResourceOrgPaths": [
                        "o-acorg/r-acroot/ou-acroot-exampleou/"
                    ]
                }
            }
        ]
    }
}
```

Note

Essa política não concede nenhum acesso. Em vez disso, essa política funciona como uma barreira para suas outras permissões do IAM, impedindo que suas entidades principais acessem objetos do Amazon S3 fora de um limite definido pela UO.

A política nega acesso às ações do Amazon S3, a menos que o objeto do Amazon S3 que está sendo acessado esteja na UO **ou-acroot-exampleou** em sua organização. A [condição da política do IAM](#)

exige `aws:ResourceOrgPaths`, uma chave de condição com vários valores, para conter qualquer um dos caminhos de UO listados. A política usa o operador `ForAllValues:StringNotLike` para comparar os valores de `aws:ResourceOrgPaths` com as UOs listadas sem distinção de maiúsculas e minúsculas.

Restringir o acesso a buckets do Amazon S3 em sua unidade organizacional

Para restringir o acesso a objetos do Amazon S3 em sua organização, anexe uma política do IAM à raiz da organização, aplicando-a a todas as contas. Para exigir que suas entidades principais do IAM sigam essa regra, use uma [política de controle de serviços \(SCP\)](#). Se você optar por usar uma SCP, [teste-a](#) antes de anexar a política à raiz da organização.

No exemplo de política a seguir, o acesso é negado às ações do Amazon S3, a menos que o objeto do Amazon S3 que está sendo acessado esteja na mesma organização que a entidade principal do IAM que está acessando:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "DenyS3AccessOutsideMyBoundary",  
            "Effect": "Deny",  
            "Action": [  
                "s3:*"  
            ],  
            "Resource": "arn:aws:s3:::/*",  
            "Condition": {  
                "StringNotEquals": {  
                    "aws:ResourceOrgID": "${aws:PrincipalOrgID}"  
                }  
            }  
        }  
    ]  
}
```

Note

Essa política não concede nenhum acesso. Essa política funciona como uma barreira para suas outras permissões do IAM, impedindo que suas entidades principais acessem quaisquer objetos do Amazon S3 de fora de sua organização. Essa política também se aplica aos recursos do Amazon S3 que são criados depois que a política é implementada.

Neste exemplo, a [condição da política do IAM](#) exige que `aws:ResourceOrgID` e `aws:PrincipalOrgID` sejam equivalentes. Com esse requisito, a entidade principal que faz a solicitação e o recurso que está sendo acessado devem estar na mesma organização.

Demonstrações de 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 AWS Management Console 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 tanto a AWS Command Line Interface (CLI) quanto o AWS Tools for Windows PowerShell.

- [Exemplo 1: Proprietário do bucket que concede permissões do bucket aos usuários \(p. 557\)](#)

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 \(p. 562\)](#)

Neste exercício, um proprietário do bucket, conta A, concede permissões entre contas a outra Conta da AWS, conta B. Em seguida, a conta B delega essas permissões aos 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 proprietária 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 os 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 para objetos que não possui \(p. 568\)](#)

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 \(p. 573\)](#)

Depois de receber as permissões do proprietário do objeto, o proprietário do bucket não pode delegar permissões a outras Contas da AWS, já que a delegação entre contas não é compatível (consulte [Delegação de permissão \(p. 428\)](#)). 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 AWS Management Console para criar recursos e conceder permissões. Para testar permissões, os exemplos usam ferramentas da linha de comando, a AWS Command Line Interface (CLI), e o AWS Tools for 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 de exemplo \(p. 554\)](#).

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 sua 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 fazer solicitações. Esse usuário é conhecido como usuário administrador. Para obter mais informações, consulte [Credenciais da conta de root vs. credenciais de usuário do IAM](#) na Referência geral da AWS e as [Práticas recomendadas 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 AWS Management Console usando as credenciais de usuário, você deverá usar o URL de login do usuário do IAM. O console do IAM fornece esse URL para sua Conta da AWS. Os tópicos mostram como obter o URL.

Configurar as ferramentas para as demonstrações de exemplo

Os exemplos introdutórios (consulte [Demonstrações de exemplo: gerenciar o acesso aos recursos do Amazon S3 \(p. 553\)](#)) usam o AWS Management Console para criar recursos e conceder permissões. Para testar permissões, os exemplos usam ferramentas da linha de comando, a AWS Command Line Interface (CLI), e o AWS Tools for 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 seguintes tópicos no Guia do usuário do AWS Command Line Interface.

[Começar a usar a AWS Command Line Interface](#)

[Instalar o AWS Command Line Interface](#)

[Configurar a AWS Command Line Interface](#)

2. Defina o perfil padrão.

Você armazenará as credenciais de usuário no arquivo de configuração da AWS CLI. Crie um perfil padrão no arquivo de configuração usando as credenciais da sua 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 AWS CLI 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 AWS CLI sem o parâmetro `--profile`, a AWS CLI usará o perfil definido na variável de ambiente como perfil padrão.

```
$ export AWS_DEFAULT_PROFILE=AccountAadmin
```

Como configurar o AWS Tools for Windows PowerShell

1. Faça download e configure a AWS Tools for Windows PowerShell. Para obter instruções, acesse [Fazer download e instalar o AWS Tools for Windows PowerShell](#) no Guia do usuário do AWS Tools for Windows PowerShell.

Note

Para carregar o módulo AWS Tools for Windows PowerShell, você precisará habilitar a execução do script PowerShell. Para obter mais informações, acesse [Habilitar a execução de scripts](#) no Guia do usuário do AWS Tools for 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 para testar os exercícios. Siga os links fornecidos no início da seção.

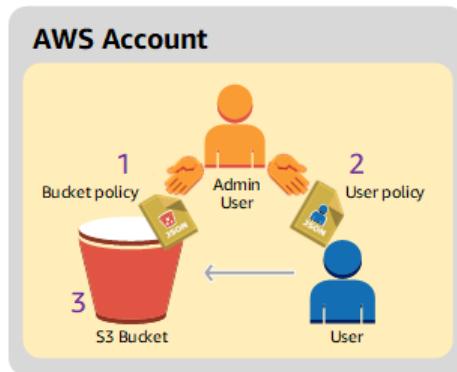
Exemplo 1: Proprietário do bucket que concede permissões do bucket aos usuários

Tópicos

- [Etapa 0: Preparar-se para a demonstração \(p. 558\)](#)
- [Etapa 1: Criar recursos \(um bucket e um usuário do IAM\) na Conta A e conceder permissões \(p. 558\)](#)
- [Etapa 2: Testar permissões \(p. 560\)](#)

Neste exercício, uma Conta da AWS é a proprietária de um bucket e há um usuário do IAM na conta. Por padrão, o usuário não tem nenhuma permissão. Para que o usuário realize execute qualquer tarefa, a conta pai deve conceder permissão a ele. O proprietário do bucket e a conta pai são os mesmos. Portanto, para conceder ao usuário permissões no bucket, a Conta da AWS pode usar uma política de bucket, uma política de usuário ou ambas. O proprietário da conta concederá algumas permissões usando uma política de bucket e 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.

Neste 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 \(p. 554\)](#)). Fazemos referência à Conta da AWS e ao usuário administrador da seguinte forma:

ID da conta	Conta referida como	Usuário administrador na conta
1111-1111-1111	Conta A	AccountAdmin

Note

O usuário administrador neste exemplo é AccountAdmin, que se refere à conta A, e não à AccountAdmin.

Todas as tarefas de criar usuários e conceder permissões são feitas no AWS Management Console. Para verificar permissões, a demonstração usa as ferramentas de linha de comando, a AWS Command Line Interface (CLI) e o AWS Tools for Windows PowerShell. Portanto, você não precisa escrever nenhum código para verificar as permissões.

Etapa 0: Preparar-se para a demonstração

1. Verifique se você tem uma Conta da AWS que tenha 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 notificará você 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 [Criação de um usuário do IAM na 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 [Como trabalhar com políticas](#) no Guia do usuário do IAM.

 - iii. Anote o IAM User Sign-In URL (URL de login do usuário do IAM) para AccountAdmin. Você precisará usar esse URL para fazer login no AWS Management Console. 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 AWS Command Line Interface (CLI) ou o AWS Tools for Windows PowerShell. Certifique-se de ter salvado as credenciais de usuário do administrador deste modo:
 - Se estiver usando a AWS CLI, crie um perfil, AccountAdmin, no arquivo de configuração.
 - Se estiver usando o AWS Tools for Windows PowerShell, armazene as credenciais para a sessão como AccountAdmin.

Para obter instruções, consulte [Configurar as ferramentas para as demonstrações de exemplo \(p. 554\)](#).

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 AWS Management Console 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 [Criação de um bucket \(p. 129\)](#).
 - b. No console do IAM, faça o seguinte:
 - i. Crie um usuário Dave.

Para obter instruções, consulte [Criação de usuários do IAM \(AWS Management Console\)](#) 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 iguais, 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 `awsexamplebucket1`.

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 [Políticas de bucket e políticas de usuário \(p. 437\)](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "statement1",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountA-ID:user/Dave"  
            },  
            "Action": [  
                "s3:GetBucketLocation",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::awsexamplebucket1"  
            ]  
        },  
        {  
            "Sid": "statement2",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountA-ID:user/Dave"  
            },  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::awsexamplebucket1/*"  
            ]  
        }  
    ]  
}
```

- b. 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": [  
        {  
            "Effect": "Allow",  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::YourBucketName/*"  
        }  
    ]  
}
```

```
        "Sid": "PermissionForObjectOperations",
        "Effect": "Allow",
        "Action": [
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::awsexamplebucket1/*"
        ]
    }
}
```

Para instruções, consulte [Como 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 AWS CLI

- Atualize o arquivo de configuração da AWS CLI adicionando o perfil de UserDaveAccountA a seguir. Para obter mais informações, consulte [Configurar as ferramentas para as demonstrações de exemplo \(p. 554\)](#).

```
[profile UserDaveAccountA]
aws_access_key_id = access-key
aws_secret_access_key = secret-access-key
region = us-east-1
```

- Verifique se Dave pode executar as operações conforme concedido na política de usuário. Carregue um objeto de exemplo usando o comando put-object da AWS CLI a seguir.

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 awsexamplebucket1 --key HappyFace.jpg --
body HappyFace.jpg --profile UserDaveAccountA
```

Execute o comando da AWS CLI a seguir para obter o objeto.

```
aws s3api get-object --bucket awsexamplebucket1 --key HappyFace.jpg OutputFile.jpg --
profile UserDaveAccountA
```

Testar usando a AWS Tools for Windows PowerShell

- 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
```

- Carregue um objeto de exemplo usando o comando Write-S3Object do AWS Tools for Windows PowerShell com as credenciais armazenadas do usuário Dave.

```
Write-S3Object -bucketname awsexamplebucket1 -key HappyFace.jpg -file HappyFace.jpg -  
StoredCredentials AccountADave
```

Faça download do objeto anteriormente carregado.

```
Read-S3Object -bucketname awsexamplebucket1 -key HappyFace.jpg -file Output.jpg -  
StoredCredentials AccountADave
```

Exemplo 2: Proprietário do bucket concedendo permissões de bucket entre contas

Tópicos

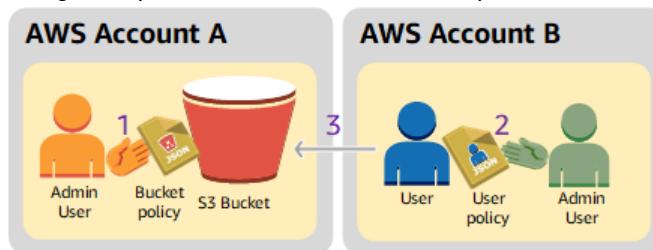
- [Etapa 0: Preparar-se para a demonstração \(p. 563\)](#)
- [Etapa 1: Fazer as tarefas da Conta A \(p. 564\)](#)
- [Etapa 2: Fazer as tarefas da Conta B \(p. 565\)](#)
- [Etapa 3: \(opcional\) tentar negação explícita \(p. 566\)](#)
- [Etapa 4: Limpeza \(p. 567\)](#)

Uma Conta da AWS, por exemplo, a conta A, pode conceder a outra Conta da AWS, a conta B, permissão para acessar os respectivos 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 \(p. 554\)](#)), 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
1111-1111-1111	Conta A	AccountAdmin
2222-2222-2222	Conta B	AccountBadmin

Todas as tarefas de criar usuários e conceder permissões são feitas no AWS Management Console. Para verificar permissões, a demonstração usa as ferramentas de linha de comando, a AWS Command Line Interface (CLI), e o AWS Tools for Windows PowerShell. Portanto, você não precisa escrever nenhum código.

Etapa 0: Preparar-se para a demonstração

1. Você precisa ter duas Contas da AWS, e cada conta deverá ter um usuário administrador, conforme mostrado na tabela na seção anterior.
 - a. Cadastre-se para obter uma Conta da AWS, se necessário.
 - i. Acesse <https://aws.amazon.com/s3/> e clique em Crie uma conta da AWS.
 - ii. Siga as instruções da tela.

A AWS notificará você 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 [Criação de um usuário do IAM na 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 [Como 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 AWS Management Console.

Para obter mais informações, consulte [Como os usuários fazem login em sua conta](#) no Guia do usuário do IAM.
 - d. Repita a etapa anterior usando as credenciais da Conta B e crie um usuário administrador AccountBAdmin.
2. Configure a AWS Command Line Interface (CLI) ou o AWS Tools for Windows PowerShell. Certifique-se de ter salvado as credenciais de usuário do administrador deste modo:
 - Se estiver usando a AWS CLI, crie dois perfis, AccountAdmin e AccountBAdmin, no arquivo de configuração.
 - Se estiver usando o AWS Tools for Windows PowerShell, armazene as credenciais para a sessão como AccountAdmin e AccountBAdmin.

Para obter instruções, consulte [Configurar as ferramentas para as demonstrações de exemplo \(p. 554\)](#).

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 de exemplo \(p. 554\)](#).
 - 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 o AWS Tools for Windows PowerShell

```
set-awscredentials -AccessKey AcctA-access-key-ID -SecretKey AcctA-secret-access-
key -storeas AccountAadmin
set-awscredentials -AccessKey AcctB-access-key-ID -SecretKey AcctB-secret-access-
key -storeas AccountBadmin
```

Etapa 1: Fazer as tarefas da Conta A

Etapa 1.1: Fazer login no AWS Management Console

Usando a URL de login do usuário do IAM para a Conta A, primeiro faça login no AWS Management Console como o usuário AccountAadmin. 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 US East (N. Virginia) e que o nome é DOC-EXAMPLE-BUCKET.

Para obter instruções, consulte [Criação de um bucket \(p. 129\)](#).

2. Faça upload de um objeto de exemplo no bucket.

Para obter instruções, vá para [Etapa 2: fazer upload de um objeto para o seu bucket \(p. 17\)](#).

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:GetLifecycleConfiguration e s3>ListBucket para a Conta B. Supõe-se que você ainda esteja conectado no console usando as credenciais do usuário AccountAadmin.

1. Anexe a política de bucket a seguir ao DOC-EXAMPLE-BUCKET. A política concede à Conta B permissão para as ações s3:GetLifecycleConfiguration e s3>ListBucket.

Para obter instruções, consulte [Adicionar uma política de bucket usando o console do Amazon S3 \(p. 516\)](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Example permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:root"
            },
            "Action": [
                "s3:GetLifecycleConfiguration",
                "s3>ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
            ]
        }
    ]
}
```

2. Verifique se a Conta B (e, portanto, do usuário administrador) pode executar as operações.

- Uso do AWS CLI

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET --profile AccountBadmin
aws s3api get-bucket-lifecycle-configuration --bucket DOC-EXAMPLE-BUCKET --profile
AccountBadmin
```

- Uso do AWS Tools for Windows PowerShell

```
get-s3object -BucketName DOC-EXAMPLE-BUCKET -StoredCredentials AccountBadmin
get-s3bucketlifecycleconfiguration -BucketName DOC-EXAMPLE-BUCKET -StoredCredentials
AccountBadmin
```

Etapa 2: Fazer 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 AWS Management Console

Usando a URL de login do usuário do IAM da Conta B, primeiro faça login no AWS Management Console 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 \(AWS Management Console\)](#) 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:::DOC-EXAMPLE-BUCKET"
            ]
        }
    ]
}
```

Para instruções, consulte [Como 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 DOC-EXAMPLE-BUCKET de propriedade da Conta A. Você pode verificar as permissões usando um dos seguintes procedimentos.

Testar usando a AWS CLI

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 de exemplo \(p. 554\)](#)

```
[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 comando da AWS CLI a seguir para verificar se agora Dave pode obter uma lista de objetos do DOC-EXAMPLE-BUCKET pertencente à conta A. Observe que o comando especifica o perfil UserDave.

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET --profile UserDave
```

Dave não tem nenhuma outra permissão. Portanto, se ele tentar realizar qualquer outra operação (por exemplo, a configuração seguinte de ciclo de vida de get bucket), o Amazon S3 retornará a recusa da permissão.

```
aws s3api get-bucket-lifecycle-configuration --bucket DOC-EXAMPLE-BUCKET --profile UserDave
```

Testar usando o AWS Tools for 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 DOC-EXAMPLE-BUCKET -StoredCredentials AccountBDave
```

Dave não tem nenhuma outra permissão. Portanto, se ele tentar realizar qualquer outra operação (por exemplo, a configuração seguinte de ciclo de vida de get bucket), o Amazon S3 retornará a recusa da permissão.

```
get-s3bucketlifecycleconfiguration -BucketName DOC-EXAMPLE-BUCKET -StoredCredentials AccountBDave
```

Etapa 3: (opcional) tentar 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 DOC-EXAMPLE-BUCKET.

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:GetLifecycleConfiguration",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET"  
            ]  
        },  
        {  
            "Sid": "Deny permission",  
            "Effect": "Deny",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountB-ID:root"  
            },  
            "Action": [  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET"  
            ]  
        }  
    ]  
}
```

2. Agora, se você tentar obter uma lista de bucket usando as credenciais de AccountBadmin, você terá o acesso negado.

- Usar a AWS CLI:

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET --profile AccountBadmin
```

- Usar a AWS Tools for Windows PowerShell:

```
get-s3object -BucketName DOC-EXAMPLE-BUCKET -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 AWS Management Console ([AWS Management Console](#)) usando as credenciais da Conta A e faça o seguinte:
 - No console do Amazon S3, remova a política de bucket anexada a **DOC-EXAMPLE-BUCKET**. 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 AccountAadmin.
2. Faça login no AWS Management Console ([AWS Management Console](#)) usando as credenciais da Conta B. No console do IAM, exclua o usuário AccountBadmin.

Exemplo 3: O proprietário do bucket concede permissões para objetos que não possui

Tópicos

- [Etapa 0: Preparar-se para a demonstração \(p. 569\)](#)
- [Etapa 1: Fazer as tarefas da Conta A \(p. 570\)](#)
- [Etapa 2: Fazer as tarefas da Conta B \(p. 571\)](#)
- [Etapa 3: Testar permissões \(p. 572\)](#)
- [Etapa 4: Limpeza \(p. 573\)](#)

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. 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 a carregar objetos. Por padrão, o proprietário do bucket não possui objetos gravados em um bucket por outra Conta da AWS. Os objetos são de propriedade das contas que os gravam em um bucket do S3. Se o proprietário do bucket não for proprietário do objeto no bucket, o proprietário do objeto deverá primeiro conceder permissão ao proprietário do bucket usando uma ACL de objeto. Em seguida, o proprietário do bucket pode conceder permissões a um objeto que ele não possui. Para obter mais informações, consulte [Propriedade de bucket e objeto do Amazon S3 \(p. 420\)](#).

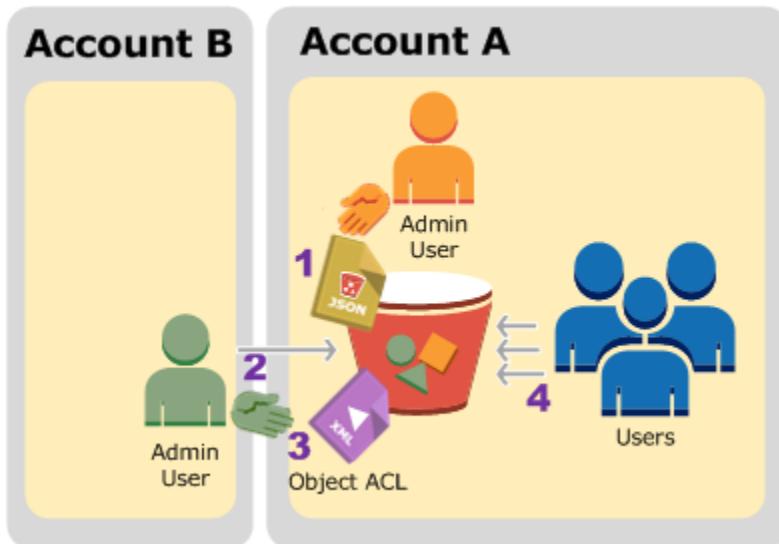
Se o proprietário do bucket adotar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership para o bucket, o proprietário do bucket terá propriedade de todos os objetos do bucket, inclusive de objetos gravados por outra Conta da AWS. Isso resolverá o problema de objetos que não pertencem ao proprietário do bucket. Em seguida, é possível delegar permissão aos usuários em sua própria conta ou para outras Contas da AWS.

Note

Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. É possível usar Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Como resultado, o controle de acesso para seus dados é baseado em políticas, como políticas do IAM, políticas de bucket do S3, políticas de endpoint da Virtual Private Cloud (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations.

A maioria dos casos de uso modernos no Amazon S3 não requer mais o uso de ACLs, e recomendamos desabilitar as ACLs, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com Object Ownership, é possível desabilitar ACLs e confiar em políticas para controle de acesso. Ao desabilitar ACLs, você pode facilmente manter um bucket com objetos carregados por diferentes Contas da AWS. Como proprietário do bucket, você possui todos os objetos do bucket e pode gerenciar o acesso a eles usando políticas. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Neste exemplo, assumimos que o proprietário do bucket não aplicou a configuração aplicada pelo proprietário do bucket em Object Ownership. 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. Nesta demonstração, você não usa as credenciais raiz da conta, de acordo com as diretrizes recomendadas do IAM. Para obter mais informações, consulte [Sobre o uso de um usuário administrador para criar recursos e conceder permissões \(p. 554\)](#). Em vez disso, você cria um administrador em cada conta e usa essas credenciais para criar recursos e conceder permissões a eles.

ID da Conta da AWS	Conta referida como	Administrador na conta
1111-1111-1111	Conta A	AccountAdmin
2222-2222-2222	Conta B	AccountBadmin

Todas as tarefas de criar usuários e conceder permissões são feitas no AWS Management Console. Para verificar permissões, a demonstração usa as ferramentas de linha de comando, a AWS Command Line Interface (AWS CLI), e o AWS Tools for Windows PowerShell. Portanto, você não precisa escrever nenhum código.

Etapa 0: Preparar-se para a demonstração

1. Você precisa ter duas Contas da AWS, e cada conta deverá ter um administrador, conforme mostrado na tabela na seção anterior.
 - a. Cadastre-se para obter uma Conta da AWS, se necessário.
 - i. Abra a [página do Amazon S3](#) e escolha Crie uma conta da AWS.

- ii. Siga as instruções na tela. A AWS notificará você 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 o usuário AccountAdmin e anote as credenciais de segurança. Para obter mais informações sobre como adicionar usuários, consulte [Criação de um usuário do IAM na sua Conta da AWS](#) no Guia do usuário do IAM.
 - Conceda permissões de administrador AccountAdmin anexando uma política de usuário com pleno acesso. Para obter instruções, consulte [Gerenciar políticas do IAM](#) 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 AWS Management Console. 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 AWS CLI ou as Tools for Windows PowerShell. Salve as credenciais de administrador da seguinte forma:
- Se estiver usando a AWS CLI, crie dois perfis, AccountAdmin e AccountBadmin, no arquivo de configuração.
 - Se estiver usando o Tools for Windows PowerShell, armazene as credenciais da sessão como AccountAdmin e AccountBadmin.

Para obter instruções, consulte [Configurar as ferramentas para as demonstrações de exemplo \(p. 554\)](#).

Etapa 1: Fazer as tarefas da Conta A

Execute as seguintes etapas para a Conta A:

Etapa 1.1: Fazer login no console

Usando o URL de login de usuário do IAM para a Conta A, primeiro faça login no AWS Management Console como usuário AccountAdmin. Esse usuário criará um bucket e anexará uma política a ele.

Etapa 1.2: Criar um bucket e um usuário e adicionar uma política de bucket para conceder permissões ao usuário

1. No console do Amazon S3, crie um bucket. Este exercício supõe que o bucket foi criado na região Leste dos EUA (Norte da Virgínia) e que o nome é [DOC-EXAMPLE-BUCKET1](#).
Para obter instruções, consulte [Criação de um bucket \(p. 129\)](#).
2. No console do IAM, crie um usuário Dave.
Para obter instruções, consulte [Criação de usuários do IAM \(console\)](#) no Guia do usuário do IAM.
3. Anote as credenciais de Dave.
4. No console do Amazon S3, anexe a seguinte política do bucket ao bucket [DOC-EXAMPLE-BUCKET1](#). Para obter instruções, consulte [Adicionar uma política de bucket usando o console do Amazon S3 \(p. 516\)](#). Siga as etapas para adicionar uma política de bucket. Para obter informações sobre como encontrar IDs de conta, consulte [Como encontrar o 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",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*",  
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"  
            ]  
        },  
        {  
            "Sid": "Statement3",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountA-ID:user/Dave"  
            },  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*"  
            ]  
        }  
    ]  
}
```

Etapa 2: Fazer 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.

Uso do AWS CLI

1. Usando o comando da CLI `put-object`, faça 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 DOC-EXAMPLE-BUCKET1 --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 [Como encontrar o ID de usuário canônico da Conta da AWS \(p. 594\)](#).

```
aws s3api put-object-acl --bucket DOC-EXAMPLE-BUCKET1 --key HappyFace.jpg --grant-full-control id="AccountA-CanonicalUserID" --profile AccountBadmin
```

Usar o Tools for Windows PowerShell

1. Usando o comando `Write-S3Object` do Tools for Windows PowerShell, faça upload de um objeto.

```
Write-S3Object -BucketName DOC-EXAMPLE-BUCKET1 -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 DOC-EXAMPLE-BUCKET1 -Key HappyFace.jpg -CannedACLName "bucket-owner-full-control" -StoredCreden
```

Etapa 3: Testar permissões

Agora verifique se o usuário Dave na conta A pode acessar o objeto de propriedade da conta B.

Uso do AWS CLI

1. Adicione as credenciais do usuário Dave ao arquivo de configuração da AWS CLI e crie um novo perfil, `UserDaveAccountA`. Para obter mais informações, consulte [Configurar as ferramentas para as demonstrações de exemplo \(p. 554\)](#).

```
[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 CLI para fazer download do arquivo `HappyFace.jpg` e salve-o localmente. Você fornece credenciais ao usuário Dave adicionando o parâmetro `--profile`.

```
aws s3api get-object --bucket DOC-EXAMPLE-BUCKET1 --key HappyFace.jpg Outputfile.jpg --profile UserDaveAccountA
```

Usar o Tools for 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 fazer download do objeto `HappyFace.jpg` e salve-o localmente. Você fornece credenciais ao usuário Dave adicionando o parâmetro `-StoredCredentials`.

```
Read-S3Object -BucketName DOC-EXAMPLE-BUCKET1 -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 [AWS Management Console](#) usando as credenciais da Conta A e faça o seguinte:
 - No console do Amazon S3, remova a política de bucket anexada a [DOC-EXAMPLE-BUCKET1](#). 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 AccountAadmin.
2. Faça login no [AWS Management Console](#) 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 possui

Tópicos

- [Histórico: Permissões entre contas e uso de funções do IAM \(p. 574\)](#)
- [Etapa 0: Preparar-se para a demonstração \(p. 575\)](#)
- [Etapa 1: Fazer as tarefas da Conta A \(p. 577\)](#)
- [Etapa 2: Fazer as tarefas da Conta B \(p. 579\)](#)
- [Etapa 3: Fazer as tarefas da Conta C \(p. 580\)](#)
- [Etapa 4: Limpeza \(p. 581\)](#)
- [Recursos relacionados \(p. 582\)](#)

Neste cenário de exemplo, você é proprietário de um bucket e habilitou outras Contas da AWS para carregar objetos. Se você aplicou a configuração aplicada pelo proprietário do bucket em S3 Object Ownership para o bucket, terá propriedade de todos os objetos do bucket, inclusive de objetos gravados por outra Conta da AWS. Isso resolverá o problema dos objetos que não pertencem a você, o proprietário do bucket. Em seguida, é possível delegar permissão aos usuários em sua própria conta ou para outras Contas da AWS. Suponha que a configuração imposta pelo proprietário do bucket em S3 Object Ownership não esteja habilitada. Ou seja, seu bucket pode ter objetos pertencentes a 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 os objetos criados por outras Contas da AWS. Assim, para que o proprietário do bucket possa conceder permissões sobre objetos que não são de sua propriedade, 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 ([consulte Exemplo 3: O proprietário do bucket concede permissões para objetos que não possui \(p. 568\)](#)), mas não pode delegar permissões a outras Contas da AWS, porque a delegação entre contas não é compatível.

Neste cenário, o proprietário do bucket pode criar uma função do AWS Identity and Access Management (IAM) 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.

Note

Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. É possível usar Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Como resultado, o controle de acesso para seus dados é baseado em políticas, como políticas do IAM, políticas de bucket do S3, políticas de endpoint da Virtual Private Cloud (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations.

A maioria dos casos de uso modernos no Amazon S3 não requer mais o uso de ACLs, e recomendamos desabilitar as ACLs, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com Object Ownership, é possível desabilitar ACLs e confiar em políticas para controle de acesso. Ao desabilitar ACLs, você pode facilmente manter um bucket com objetos carregados por diferentes Contas da AWS. Como proprietário do bucket, você possui todos os objetos do bucket e pode gerenciar o acesso a eles usando políticas. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

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 o acesso entre contas a objetos aos usuários em uma outra Conta da AWS, conta C. Cada função do IAM que você criar terá duas políticas anexadas a ela:

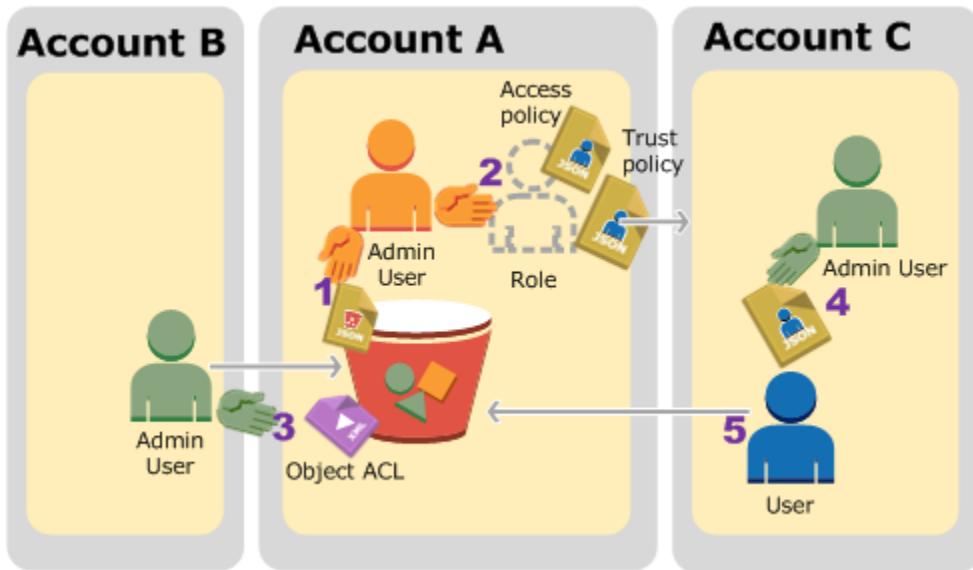
- Uma política de confiança que identifica outra Conta da AWS que pode assumir a função.
- Uma política de acesso que define quais permissões - por exemplo, `s3:GetObject` - são permitidas quando alguém assume a função. Para obter uma lista de permissões que você pode especificar em uma política, consulte [Ações do Amazon S3 \(p. 441\)](#).

A Conta da AWS identificada na política de confiança concede ao respectivo usuário permissão 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 \(p. 554\)](#)), 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
1111-1111-1111	Conta A	AccountAadmin
2222-2222-2222	Conta B	AccountBadmin
3333-3333-3333	Conta C	AccountCadmin

Etapa 0: Preparar-se 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. Você precisa ter três Contas da AWS, e cada conta de ter 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 Crie uma conta da AWS.
 - ii. Siga as instruções da tela.

A AWS notificará você 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 [Criação de 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 [Como 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 AWS Management Console. Para obter mais informações, acesse [Como os usuários fazem login em sua 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 de usuário canônico.

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. Use o ID ou o alias da Conta da AWS, o nome de usuário do IAM, e a senha para fazer login no [console do Amazon S3](#).
 - b. Selecione o nome de um bucket do Amazon S3 para visualizar os detalhes sobre esse bucket.
 - c. Selecione a guia Permissions (Permissões) e selecione Access Control List (Lista de controle de acesso).
 - d. Na seção Access for your Conta da AWS (Acesso à sua Conta da AWS), a coluna Account (Conta) inclui um identificador longo, como c1daexampleaf850ea79cf0430f33d72579fd1611c97f7ded193374c0b163b6. Esse é seu 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 AWS Command Line Interface (CLI) ou o AWS Tools for Windows PowerShell. Certifique-se de ter salvado as credenciais de usuário do administrador deste modo:
 - Se estiver usando a AWS CLI, crie dois perfis, AccountAdmin e AccountBadmin, no arquivo de configuração.
 - Se estiver usando o AWS Tools for Windows PowerShell, armazene as credenciais para a sessão como AccountAdmin e AccountBAdmin.

Para obter instruções, consulte [Configurar as ferramentas para as demonstrações de exemplo \(p. 554\)](#).

Etapa 1: Fazer 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: Fazer login no AWS Management Console

Usando o URL de login de usuário do IAM para a Conta A, primeiro faça login no AWS Management Console como usuário AccountAdmin. Esse usuário criará um bucket e anexará uma política a ele.

Etapa 1.2: Criar um bucket e anexar 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 [Criação de um bucket \(p. 129\)](#).

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:::awsexamplebucket1/*"  
        },  
        {  
            "Sid": "112",  
            "Effect": "Deny",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"  
            },  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::awsexamplebucket1/*"  
        }  
    ]  
}
```

```
"Principal": {
    "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
},
"Action": "s3:PutObject",
"Resource": "arn:aws:s3:::awsexamplebucket1/*",
"Condition": {
    "StringNotEquals": {
        "s3:x-amz-grant-full-control": "id=CanonicalUserId-of-AWSaccountA-
BucketOwner"
    }
}
}
```

Etapa 1.3: Criar uma função do IAM para conceder à Conta C acesso entre contas à Conta A

No console do IAM, crie uma função do IAM (“examplerole”) 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:::awsexamplebucket1/*"
        }
    ]
}
```

- 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. 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 Contas da AWS you own (Fornecer acesso entre suas Contas da AWS).
4. 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.

5. Clique em Next Step (Próxima etapa) para definir as permissões que serão associadas à função.
- 6.

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.

7. Insira `examplerole` para o nome da função e, em seguida, clique em Next Step (Próxima etapa).
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, vá para [AssumeRole](#) na Referência da API do AWS Security Token Service.

```
{  
    "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: Fazer 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 `put-object` da AWS CLI, carregue um objeto para o `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: Fazer as tarefas da Conta C

Nas etapas anteriores, a Conta A já criou uma função, exempleroles, estabelecendo a confiança com a Conta C. Isso permite que os usuários na Conta C accedam 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 exempleroles 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 acceder a Conta A – especificamente, obter objetos em examplebucket.

Etapa 3.1: Criar um usuário na Conta C e delegar a permissão para assumir exempleroles

1. Usando o URL de login de usuário do IAM para a Conta C, primeiro faça login no AWS Management Console como usuário AccountCadmin.
2. No console do IAM, crie um usuário Dave.

Para obter instruções, consulte [Criação de usuários do IAM \(AWS Management Console\)](#) no Guia do usuário do IAM.

3. Anote as credenciais de Dave. Dave precisará dessas credenciais para assumir a função exempleroles.
4. Crie uma política inline para o usuário Dave do IAM para delegar a Dave a permissão sts:AssumeRole na função exempleroles 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 configuração da AWS CLI 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 (exampleroles) e acessar objetos

Agora Dave pode acessar objetos no bucket de propriedade da Conta A, desta forma:

- Dave primeiro assume a exampleroles 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 comando `assume-role` da AWS CLI 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 exampleroles foi definido.

```
aws sts assume-role --role-arn arn:aws:iam::accountA-ID:role/exampleroles --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 de configuração da AWS CLI 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 da AWS CLI a seguir 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 exampleroles 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 exampleroles.

Etapa 4: Limpeza

1. Depois de terminar os testes, você pode seguir uma das etapas a seguir para fazer a limpeza.
 - Faça login no AWS Management Console ([AWS Management Console](#)) usando as credenciais da Conta A e faça o seguinte:

- No console do Amazon S3, remova a política do 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 AccountAdmin.
2. Faça login no AWS Management Console ([AWS Management Console](#)) usando as credenciais da Conta B. No console do IAM, exclua o usuário AccountBadmin.
 3. Faça login no AWS Management Console ([AWS Management Console](#)) usando as credenciais da Conta C. No console do IAM, exclua o usuário AccountCadmin e o usuário Dave.

Recursos relacionados

- [Criação de uma função para delegar permissões a um usuário do IAM](#) no Guia do usuário do IAM.
- [Tutorial do IAM: delegar acesso entre contas da AWS usando funções do IAM](#) no Guia do usuário do IAM.
- [Como trabalhar com políticas](#) no Guia do usuário do IAM.

Usar funções vinculadas a serviços para o Amazon S3 Storage Lens

Para usar o Amazon S3 Storage Lens para coletar e agregar métricas em todas as suas contas no AWS Organizations, primeiro você deve garantir que o S3 Storage Lens tenha acesso confiável habilitado pela conta de gerenciamento em sua organização. O S3 Storage Lens cria uma função vinculada a serviços para permitir que ele obtenha a lista de Contas da AWS pertencentes à sua organização. Essa lista de contas é usada pelo S3 Storage Lens para coletar métricas de recursos do S3 em todas as contas membro quando o painel ou as configurações do S3 Storage Lens são criadas ou atualizadas.

O Amazon S3 Storage Lens usa [funções vinculadas a serviços](#) do AWS Identity and Access Management (IAM). Uma função vinculada a serviço é um tipo exclusivo de função do IAM vinculada diretamente ao S3 Storage Lens. As funções vinculadas a serviços são predefinidas pelo S3 Storage Lens e incluem todas as permissões que o serviço requer para chamar outros produtos da AWS em seu nome.

Uma função vinculada a serviços facilita a configuração do S3 Storage Lens porque você não precisa adicionar as permissões necessárias manualmente. O S3 Storage Lens define as permissões da função vinculada a serviços e, salvo outra definição, somente o S3 Storage Lens pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, e essa política não pode ser anexada a nenhuma outra entidade do IAM.

Você pode excluir essa função vinculada a serviços somente após a exclusão dos recursos relacionados. Isso protege seus recursos do S3 Storage Lens, porque você não consegue remover por engano a permissão para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com funções vinculadas a serviços, consulte [Serviços da AWS que funcionam com o IAM](#) e procure serviços que tenham Yes (Sim) na coluna Service-Linked Role (Função vinculada a serviço). Escolha um Sim com um link para visualizar a documentação sobre funções vinculadas ao serviço para esse serviço.

Permissões de funções vinculadas a serviços para o Amazon S3 Storage Lens

O S3 Storage Lens usa a função vinculada a serviços chamada AWSServiceRoleForS3StorageLens. Ela permite o acesso a serviços e recursos da AWS usados ou gerenciados pelo S3 Storage Lens. Isso permite que o S3 Storage Lens acesse recursos da AWS Organizations em seu nome.

A função vinculada a serviços do S3 Storage Lens confia no seguinte serviço no armazenamento da sua organização:

- `storage-lens.s3.amazonaws.com`

A política de permissões da função permite que o S3 Storage Lens execute as seguintes ações:

- `organizations:DescribeOrganization`
- `organizations>ListAccounts`
- `organizations>ListAWSAccessForOrganization`
- `organizations>ListDelegatedAdministrators`

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua uma função vinculada ao serviço. Para obter mais informações, consulte [Service-linked role permissions](#) (Permissões de função vinculada a serviços) no Guia do usuário do IAM.

Criar uma função vinculada a serviços para o S3 Storage Lens

Você não precisa criar manualmente uma função vinculada a serviço. Quando você conclui uma das seguintes tarefas enquanto faz login no gerenciamento do AWS Organizations ou nas contas de administrador delegado, o S3 Storage Lens cria a função vinculada a serviços para você:

- Crie uma configuração de painel do S3 Storage Lens para sua organização no console do Amazon S3.
- Executar o comando `PUT` de uma configuração do S3 Storage Lens para sua organização usando a API REST, a AWS CLI e os SDKs.

Note

O S3 Storage Lens suportará no máximo cinco administradores delegados por organização.

Se você excluir essa função vinculada a serviços, as ações anteriores a recriarão conforme necessário.

Exemplo de política para função vinculada a serviços do S3 Storage Lens

Example Política de permissões para a função vinculada a serviços do S3 Storage Lens

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AwsOrgsAccess",  
            "Effect": "Allow",  
            "Action": [  
                "organizations:DescribeOrganization",  
                "organizations>ListAccounts",  
                "organizations>ListAWSAccessForOrganization",  
                "organizations>ListDelegatedAdministrators"  
            ],  
            "Resource": [  
                "*"  
            ]  
        }  
    ]  
}
```

[Editar uma função vinculada a serviços para o Amazon S3 Storage Lens](#)

O S3 Storage Lens não permite que você edite a função vinculada a serviços `AWSServiceRoleForS3StorageLens`. Depois que criar uma função vinculada ao serviço, você não poderá alterar o nome da função, pois várias entidades podem fazer referência a ela. No entanto, será possível editar a descrição da função usando o IAM. Para obter mais informações, consulte [Editar uma função vinculada a serviço](#) no Guia do usuário do IAM.

[Excluir uma função vinculada a serviços para o Amazon S3 Storage Lens](#)

Se você não precisa mais usar a função vinculada a serviços, recomendamos que a exclua. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de sua função vinculada ao serviço antes de exclui-la manualmente.

Note

Se o serviço Amazon S3 Storage Lens estiver usando a função quando você tentar excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para excluir `AWSServiceRoleForS3StorageLens`, você deve excluir todas as configurações do S3 Storage Lens de nível da organização presentes em todas as regiões usando o AWS Organizations Management ou as contas de administrador delegadas.

Os recursos são configurações do S3 Storage Lens no nível da organização. Use o S3 Storage Lens para limpar os recursos e, em seguida, use o console do IAM, a CLI, a API REST ou o AWS SDK para excluir a função.

Na API REST, na AWS CLI e nos SDKs, as configurações do S3 Storage Lens podem ser descobertas usando `ListStorageLensConfigurations` em todas as regiões onde sua organização tiver criado configurações do S3 Storage Lens. Use a ação `DeleteStorageLensConfiguration` a fim de excluir essas configurações para que você possa excluir a função.

Note

Para excluir a função vinculada a serviços, você deve excluir todas as configurações do S3 Storage Lens no nível da organização em todas as regiões em que elas existem.

Para excluir recursos do Amazon S3 Storage Lens usados por `AWSServiceRoleForS3StorageLens`

1. Você deve usar `ListStorageLensConfigurations` em todas as regiões em que você tem configurações do S3 Storage Lens para obter uma lista de configurações de nível de organização. Essa lista também pode ser obtida no console do Amazon S3.
2. Essas configurações devem ser excluídas dos endpoints regionais apropriados invocando a chamada de API `DeleteStorageLensConfiguration` ou pelo console do Amazon S3.

Como excluir manualmente a função vinculada ao serviço usando o IAM

Depois que as configurações forem excluídas, o `AWSServiceRoleForS3StorageLens` poderá ser excluído pelo console do IAM ou invocando a API `DeleteServiceLinkedRole` do IAM, a AWS CLI ou o AWS SDK. Para obter mais informações, consulte [Excluir uma função vinculada ao serviço](#) no Guia do usuário do IAM.

[Regiões com suporte para funções vinculadas a serviços do S3 Storage Lens](#)

O S3 Storage Lens é compatível com as funções vinculadas a serviços em todas as Regiões da AWS em que o serviço está disponível. Para obter mais informações, consulte [Regiões e endpoints do Amazon S3](#).

Políticas gerenciadas da AWS para o Amazon S3

Para adicionar permissões a usuários, grupos e funções, é mais fácil usar políticas gerenciadas pela AWS do que gravar políticas por conta própria. É necessário tempo e experiência para [criar políticas gerenciadas pelo cliente do IAM](#) que fornecem à sua equipe apenas as permissões de que precisam. Para começar rapidamente, você pode usar nossas políticas gerenciadas pela AWS. Essas políticas abrangem casos de uso comuns e estão disponíveis na sua Conta da AWS. Para obter mais informações sobre as políticas gerenciadas da AWS, consulte [Políticas gerenciadas da AWS](#) no Guia do usuário do IAM.

Os Serviços da AWS mantêm e atualizam políticas gerenciadas pela AWS. Não é possível alterar as permissões em políticas gerenciadas pela AWS. Os serviços ocasionalmente acrescentam permissões adicionais a uma política gerenciada pela AWS para oferecer suporte a novos recursos. Esse tipo de atualização afeta todas as identidades (usuários, grupos e funções) em que a política está anexada. É mais provável que os serviços atualizem uma política gerenciada pela AWS quando um novo recurso for iniciado ou novas operações se tornarem disponíveis. Os serviços não removem permissões de uma política gerenciada pela AWS, portanto, as atualizações de políticas não suspendem suas permissões existentes.

Além disso, a AWS oferece suporte a políticas gerenciadas para funções de trabalho que abrangem vários serviços. Por exemplo, a política gerenciada pela AWS `viewOnlyAccess` fornece acesso somente de leitura a muitos Serviços da AWS e recursos. Quando um serviço executa um novo recurso, a AWS adiciona permissões somente leitura para novas operações e recursos. Para obter uma lista e descrições das políticas de funções de trabalho, consulte [Políticas gerenciadas pela AWS para funções de trabalho](#) no Guia do usuário do IAM.

Política gerenciada pela AWS: AmazonS3FullAccess

É possível anexar a política `AmazonS3FullAccess` a suas identidades do IAM. Essa política concede permissões que possibilitam acesso total ao Amazon S3.

Para visualizar as permissões para esta política, consulte [AmazonS3FullAccess](#) no AWS Management Console.

Política gerenciada pela AWS: AmazonS3ReadOnlyAccess

É possível anexar a política `AmazonS3ReadOnlyAccess` a suas identidades do IAM. Esta política concede permissões que oferecem acesso somente leitura ao Amazon S3.

Para visualizar as permissões para esta política, consulte [AmazonS3ReadOnlyAccess](#) no AWS Management Console.

Política gerenciada da AWS: AmazonS3ObjectLambdaExecutionRolePolicy

Fornece às funções do AWS Lambda as permissões necessárias para enviar dados para o S3 Object Lambda quando solicitações são feitas para um ponto de acesso do S3 Object Lambda. Também concede permissões do Lambda para gravar nos logs do Amazon CloudWatch.

Para visualizar as permissões para esta política, consulte [AmazonS3ObjectLambdaExecutionRolePolicy](#) no AWS Management Console.

Atualizações do Amazon S3 para políticas gerenciadas pela AWS

Visualize detalhes sobre atualizações em políticas gerenciadas pela AWS para o Amazon S3 desde que esse serviço começou a rastrear essas alterações.

Alteração	Descrição	Data
O Amazon S3 adicionou permissões do Lambda para objetos do S3 a <code>AmazonS3FullAccess</code> e <code>AmazonS3ReadOnlyAccess</code>	O Amazon S3 atualizou as políticas <code>AmazonS3FullAccess</code> e <code>AmazonS3ReadOnlyAccess</code> para incluir do Lambda para objetos do S3.	27 de setembro de 2021
O Amazon S3 adicionou <code>AmazonS3ObjectLambdaExecutionRolePolicy</code>	O Amazon S3 adicionou uma nova política gerenciada pela AWS chamada <code>AmazonS3ObjectLambdaExecutionRolePolicy</code> que fornece permissões de funções Lambda para interagir com o S3 Object Lambda e gravar em logs do CloudWatch.	18 de agosto de 2021
O Amazon S3 começou a monitorar as alterações	O Amazon S3 passou a controlar as alterações para as políticas gerenciadas pela AWS.	18 de agosto de 2021

Gerenciar o acesso com ACLs

As listas de controle de acesso (ACLs) são uma das opções baseadas em recursos (consulte [Visão geral do gerenciamento de acesso \(p. 419\)](#)) que podem ser usadas 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 a outras Contas da AWS. Não é possível conceder permissões a 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 é proprietária do objeto.

Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. É possível usar Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Como resultado, o controle de acesso para seus dados é baseado em políticas, como políticas do IAM, políticas de bucket do S3, políticas de endpoint da Virtual Private Cloud (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations.

A maioria dos casos de uso modernos no Amazon S3 não requer mais o uso de ACLs, e recomendamos desabilitar as ACLs, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com Object Ownership, é possível desabilitar ACLs e confiar em políticas para controle de acesso. Ao desabilitar ACLs, você pode facilmente manter um bucket com objetos carregados por diferentes Contas da AWS. Como proprietário do bucket, você possui todos os objetos do bucket e pode gerenciar o acesso a eles usando políticas. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Important

Se o bucket usar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Solicitações para definir ACLs ou atualizar ACLs falham e retornam o código de erro `AccessControlListNotSupported`. Ainda há suporte para solicitações de leitura de ACLs.

Para obter mais informações sobre ACLs, consulte os tópicos a seguir.

Tópicos

- [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#)
- [Como encontrar o ID de usuário canônico da Conta da AWS \(p. 594\)](#)
- [Configurar ACLs \(p. 596\)](#)

Visão geral da lista de controle de acesso (ACL)

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 grupos ou Contas da AWS o acesso é concedido, bem como 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.

Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. É possível usar Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Como resultado, o controle de acesso para seus dados é baseado em políticas,

como políticas do IAM, políticas de bucket do S3, políticas de endpoint da Virtual Private Cloud (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations.

A maioria dos casos de uso modernos no Amazon S3 não requer mais o uso de ACLs, e recomendamos desabilitar as ACLs, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com Object Ownership, é possível desabilitar ACLs e confiar em políticas para controle de acesso. Ao desabilitar ACLs, você pode facilmente manter um bucket com objetos carregados por diferentes Contas da AWS. Como proprietário do bucket, você possui todos os objetos do bucket e pode gerenciar o acesso a eles usando políticas. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Important

Se o bucket usar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Solicitações para definir ACLs ou atualizar ACLs falham e retornam o código de erro `AccessControlListNotSupported`. Ainda há suporte para solicitações de leitura de ACLs.

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 um ID de usuário canônico da Conta da AWS \(p. 589\)](#). 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.

Tópicos

- [Quem é o favorecido? \(p. 589\)](#)
- [Quais permissões posso conceder? \(p. 591\)](#)
- [Amostra de ACL \(p. 593\)](#)
- [ACL pré-configurada \(p. 594\)](#)

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 da AWS.

Ao conceder direitos de acesso, você especifica cada favorecido como um par de tipo=valor, em que o tipo é um dos seguintes:

- `id`: se o valor especificado é o ID de usuário canônico de uma Conta da AWS
- `uri` — se as permissões estiverem sendo concedidas a um grupo predefinido
- `emailAddress`: se o valor especificado é o endereço de e-mail de uma Conta da AWS

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 (N. da Califórnia)
- Oeste dos EUA (Oregon)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Europa (Irlanda)
- América do Sul (São Paulo)

Para obter uma lista de todas as regiões e endpoints em que o Amazon S3 tem suporte, consulte [Regions and Endpoints](#) (Regiões e endpoints) na Amazon Web Services General Reference (Referência geral da Amazon Web Services).

Example Exemplo: endereço de e-mail

Por exemplo, o cabeçalho `x-amz-grant-read` a seguir concede às Contas da AWS identificadas por endereços de e-mail permissões para ler dados de objetos e os respectivos metadados:

```
x-amz-grant-read: emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"
```

Warning

Ao conceder aos recursos acesso a outras Contas da AWS, esteja ciente de que as Contas da AWS podem delegar as permissões delas a usuários de suas próprias contas. Isso é conhecido como acesso entre contas. Para obter informações sobre como usar o acesso entre contas, consulte [Criar uma função para delegar permissões a um usuário do IAM](#) no Guia do usuário do IAM.

Encontrar um ID de usuário canônico da Conta da AWS

O ID de usuário canônico está associado à Conta da AWS. Este ID é uma longa string de caracteres, como `79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be`. Para obter

informações sobre como encontrar o ID de usuário canônico da sua conta, consulte [Como encontrar o ID de usuário canônico da Conta da AWS \(p. 594\)](#).

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 da AWS individual recebe permissões por meio de uma solicitação de concessão, um registro de concessão é adicionado à ACL com o ID de usuário canônico da conta.

Note

Se você tornar seu bucket público (não recomendado) qualquer usuário não autenticado pode carregar objetos para o bucket. Esses usuários anônimos não têm uma Conta da 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. Para obter mais informações, consulte [Propriedade de bucket e objeto do Amazon S3 \(p. 420\)](#).

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, embora WRITE as permissões não permitam que não proprietários substituam ou excluam objetos existentes, as permissões WRITE ainda permitem que qualquer pessoa armazene objetos em seu bucket, pelo qual você é cobrado. Para obter mais informações sobre essas permissões, consulte a seguinte seção [Quais permissões posso conceder? \(p. 591\)](#).

- 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 [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#)) 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, vá para [Uso do 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.

Para obter mais informações sobre permissões de ACL no console do Amazon S3, consulte [Configurar ACLs \(p. 596\)](#).

Permissões de ACL

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 que o favorecido crie novos objetos no bucket. Para os proprietários de bucket e objeto de objetos existentes, também permite exclusões e substituições desses objetos.	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, conceder acesso de WRITE a um bucket permite que o favorecido crie objetos no bucket. É altamente recomendável que você leia esta seção [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#) 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 [Ações do Amazon S3 \(p. 441\)](#)). 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 uma ACL. Use ACLs principalmente para conceder permissões básicas de leitura/gravação, similares às permissões de sistema de arquivos. Para obter mais informações sobre quando usar uma ACL, consulte [Diretrizes para políticas de acesso \(p. 425\)](#).

Para obter mais informações sobre permissões de ACL no console do Amazon S3, consulte [Configurar ACLs \(p. 596\)](#).

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	s3>ListBucket, s3>ListBucketVersions, e s3>ListBucketMultipartUploads	s3GetObject e s3GetObjectVersion
WRITE	<p>s3PutObject</p> <p>O proprietário do bucket pode criar, substituir e excluir qualquer objeto no bucket, e o proprietário do objeto tem FULL_CONTROL sobre seu objeto.</p> <p>Além disso, quando o favorecido é o proprietário do bucket, conceder a permissão WRITE em uma ACL do bucket permite que a ação s3DeleteObjectVersion seja executada em qualquer versão naquele bucket.</p>	Não aplicável
READ_ACP	s3GetBucketAcl	s3GetObjectAcl e s3GetObjectVersionAcl
WRITE_ACP	s3PutBucketAcl	s3PutObjectAcl e s3PutObjectVersionAcl
FULL_CONTROL	Equivalentes 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.	Equivalentes 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.

Chaves de condição

Ao conceder permissões de política de acesso, você pode usar chaves de condição para restringir o valor da ACL em um objeto usando uma política de bucket. As chaves de contexto abaixo correspondem a ACLs. Você pode usar essas chaves de contexto para obrigar a usar uma ACL específica em uma solicitação:

- s3:x-amz-grant-read: exigir acesso de leitura.
- s3:x-amz-grant-write: exigir acesso de gravação.
- s3:x-amz-grant-read-acp: exigir acesso de leitura à ACL do bucket.
- s3:x-amz-grant-write-acp: exigir acesso de gravação à ACL do bucket.
- s3:x-amz-grant-full-control: exigir controle total.
- s3:x-amz-acl: exigir um [ACL pré-configurada \(p. 594\)](#).

Para obter exemplos de políticas que envolvem cabeçalhos específicos de ACL, consulte [Exemplo 1: Concessão da permissão s3:PutObject com uma condição que exige que o proprietário do bucket obtenha controle total \(p. 448\)](#). Para obter uma lista completa de chaves de condição específicas do Amazon S3, consulte [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#).

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 <code>FULL_CONTROL</code> . Ninguém mais tem direitos de acesso (padrão).
<code>public-read</code>	Bucket e objeto	O proprietário obtém <code>FULL_CONTROL</code> . O grupo <code>AllUsers</code> (consulte Quem é o favorecido? (p. 589)) obtém acesso <code>READ</code> .
<code>public-read-write</code>	Bucket e objeto	O proprietário obtém <code>FULL_CONTROL</code> . 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 <code>FULL_CONTROL</code> . O Amazon EC2 obtém acesso <code>READ</code> a GET um pacote de imagem de máquina da Amazon (AMI) do Amazon S3.
<code>authenticated-read</code>	Bucket e objeto	O proprietário obtém <code>FULL_CONTROL</code> . O grupo <code>AuthenticatedUsers</code> obtém acesso <code>READ</code> .
<code>bucket-owner-read</code>	Objeto	O proprietário do objeto obtém <code>FULL_CONTROL</code> . 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 <code>FULL_CONTROL</code> 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 logs, consulte (Registrar em log as solicitações com registro em log de acesso ao servidor (p. 1041)).

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 encontrar o ID de usuário canônico da Conta da AWS

O ID de usuário canônico é um identificador alfanumérico, como `79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be`, que é uma forma complexa de ID da Conta da AWS. Você pode usar esse ID para identificar uma Conta da AWS ao conceder acesso entre contas a buckets e objetos usando o Amazon S3. Você pode recuperar o ID de usuário canônico da sua Conta da AWS como o usuário raiz ou um usuário do IAM.

Encontre o ID do usuário canônico da Conta da AWS usando o AWS Management Console ou a AWS CLI. O ID de usuário canônico de uma Conta da AWS é específico dessa conta. Você pode recuperar o ID de usuário canônico da sua conta como o usuário root, um usuário federado ou um usuário do IAM.

Pré-requisitos

Se você for um usuário federado ou estiver acessando as informações programaticamente, como pela AWS CLI, deverá ter permissão para listar e exibir um bucket do Amazon S3.

Usar o console do S3 (usuário root ou um usuário do IAM)

Siga estas etapas para encontrar o ID de usuário canônico da sua Conta da AWS quando estiver conectado ao console como usuário raiz ou um usuário do IAM. Para obter mais informações sobre o usuário root e os usuários do IAM, consulte [Visão geral do gerenciamento de identidades da AWS: usuários](#) no Guia do usuário do IAM.

1. Faça login no console como usuário root ou um usuário do IAM.

Para obter mais informações, consulte [Login no AWS Management Console](#) no Guia do usuário do IAM.

2. No canto superior direito da barra de navegação, escolha o nome ou número da conta e, em seguida, Security Credentials (Credenciais de segurança).
3. Encontre o ID canônico da conta:
 - Se você for o usuário root, expanda Account identifiers (Identificadores de conta) e encontre Canonical User ID (ID de usuário canônico).
 - Se você for um usuário do IAM, em Account details (Detalhes da conta), encontre Account canonical user ID (ID de usuário canônico da conta).

Usar o console do S3 (usuário federado)

Siga estas etapas para encontrar o ID de usuário canônico da sua conta quando estiver conectado ao AWS Management Console como usuário federado. Para obter mais informações sobre usuários federados, consulte [Federação de usuários existentes](#) no Guia do usuário do IAM.

Note

Para verificar se você fez login no AWS Management Console como um usuário federado, na página do AWS Management Console, escolha as informações da sua conta e verifique as informações expandidas da conta. Se Federated user (Usuário federado) estiver sendo exibido nas informações da conta, você está conectado como um usuário federado.

1. Faça login no console; como usuário federado.

Para obter mais informações, consulte [Login no AWS Management Console](#) no Guia do usuário do IAM.

2. No console do Amazon S3, escolha um nome de bucket para exibir os detalhes do bucket.
3. Selecione Permissions (Permissões) e role para baixo até a seção Access control list (ACL) [Lista de controle de acesso (ACL)].

Em Bucket owner (your AWS account) [Proprietário do bucket (sua conta da AWS)], você verá o ID de usuário canônico da Conta da AWS.

Uso do AWS CLI

Use o comando [list-buckets](#) da seguinte forma para encontrar o ID de usuário canônico usando a AWS CLI.

```
aws s3api list-buckets --query Owner.ID --output text
```

Configurar ACLs

Esta seção explica como gerenciar permissões de acesso para buckets do S3 e objetos usando listas de controle de acesso (ACLs). Você pode adicionar concessões ao AC do recurso usando o AWS Management Console, a AWS Command Line Interface (CLI), a API REST ou os AWS SDKs.

As permissões do bucket e de objeto são independentes uma da outra. Um objeto não herda as permissões de seu bucket. Por exemplo, se criar um bucket e conceder acesso de gravação a outro usuário, você não poderá acessar os objetos desse usuário, a menos que ele conceda o acesso explicitamente.

Você pode conceder permissões a outros usuários da Conta da AWS ou a grupos predefinidos. O usuário ou o grupo para o qual você concede permissões é chamado de usuário favorecido. Por padrão, o proprietário, que é a Conta da AWS que criou o bucket, tem permissões totais.

Cada permissão concedida a um usuário ou a um grupo adiciona uma entrada na ACL associada ao bucket. A ACL lista concessões, o que identifica o usuário favorecido e a permissão concedida.

Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. É possível usar Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Como resultado, o controle de acesso para seus dados é baseado em políticas, como políticas do IAM, políticas de bucket do S3, políticas de endpoint da Virtual Private Cloud (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations.

A maioria dos casos de uso modernos no Amazon S3 não requer mais o uso de ACLs, e recomendamos desabilitar as ACLs, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com Object Ownership, é possível desabilitar ACLs e confiar em políticas para controle de acesso. Ao desabilitar ACLs, você pode facilmente manter um bucket com objetos carregados por diferentes Contas da AWS. Como proprietário do bucket, você possui todos os objetos do bucket e pode gerenciar o acesso a eles usando políticas. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Important

Se o bucket usar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Solicitações para definir ACLs ou atualizar ACLs falham e retornam o código de erro `AccessControlListNotSupported`. Ainda há suporte para solicitações de leitura de ACLs.

Warning

Recomendamos que você evite conceder acesso de gravação aos grupos Todos (acesso público) ou Usuários autenticados (todos os usuários autenticados da AWS). Para obter mais informações sobre os efeitos da concessão de acesso de gravação a esses grupos, consulte [Grupos predefinidos do Amazon S3 \(p. 590\)](#).

[Usar o console do S3 para definir permissões de ACL para um bucket](#)

O console exibe concessões de acesso combinadas para favorecidos duplicados. Para ver a lista completa de ACLs, use a API REST do Amazon S3, a AWS CLI ou os AWS SDKs.

A tabela a seguir mostra as permissões de ACL que você pode configurar para buckets no console do Amazon S3.

Permissões de ACL do console do Amazon S3 para buckets

Permissão do console	Permissão da ACL	Acesso
Objetos - listar	READ	Permite ao favorecido listar os objetos no bucket
Objetos - gravar	WRITE	Permite que o favorecido crie novos objetos no bucket. Para os proprietários de bucket e objeto de objetos existentes, também permite exclusões e substituições desses objetos.
ACL de bucket: ler	READ_ACP	Permite ao favorecido ler a ACL do bucket
ACL do bucket - gravar	WRITE_ACP	Permite ao favorecido gravar a ACL para o bucket aplicável
Todos (acesso público): Objetos - listar	READ	Concede acesso público de leitura para os objetos no bucket. Quando você concede acesso à lista a Todos (acesso público), qualquer pessoa no mundo pode acessar os objetos no bucket.
Todos (acesso público): ACL do bucket - Ler	READ_ACP	Concede acesso de leitura pública para a ACL de bucket. Quando você concede acesso de leitura a Todos (acesso público), qualquer pessoa no mundo pode acessar a ACL de bucket.

Para obter mais informações sobre permissões de ACL, consulte [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#).

Important

Se o bucket usar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Solicitações para definir ACLs ou atualizar ACLs falham e retornam o código de erro `AccessControlListNotSupported`. Ainda há suporte para solicitações de leitura de ACLs.

Para definir permissões de ACL para um bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket para o qual você deseja definir permissões.
3. Escolha Permissions (Permissões).
4. Em Access control list (Lista de controle de acesso), escolha Edit (Editar).

Você pode editar as seguintes permissões de ACL para o bucket:

Objetos

- List (Listar): permite ao favorecido listar os objetos no bucket.
- Gravação: permite que o favorecido crie novos objetos no bucket. Para os proprietários de bucket e objeto de objetos existentes, também permite exclusões e substituições desses objetos.

No console do S3, você só pode conceder acesso de gravação ao grupo de entrega de log do S3 e ao proprietário do bucket (sua Conta da AWS). É recomendável não conceder acesso de gravação a outros favorecidos. No entanto, se você precisar conceder acesso de gravação, poderá usar a AWS CLI, os AWS SDKs ou a API REST.

Bucket ACL

- Read (Ler): permite ao beneficiário ler a ACL do bucket.
 - Write (Gravar): permite ao beneficiário gravar a ACL para o bucket aplicável.
5. Para alterar as permissões do proprietário do bucket, ao lado de Bucket owner (your Conta da AWS) [Proprietário do bucket (sua Conta da AWS)], desmarque ou selecione uma das seguintes permissões de ACL:
- Objetos: listar ou gravar
 - ACL de bucket: ler ou gravar

O proprietário refere-se ao usuário root da Conta da AWS, e não um usuário do AWS Identity and Access Management (IAM). Para obter mais informações sobre o usuário root, consulte [Usuário root da Conta da AWS](#) no Guia do usuário do IAM.

6. Para conceder ou desfazer permissões para o público em geral (todos na Internet), ao lado de Everyone (public access) (Todos (acesso público)), limpe ou selecione uma das seguintes permissões da ACL:
- Objetos: listar
 - ACL de bucket: ler

Warning

Tenha cuidado ao conceder ao grupo Everyone (Todos) acesso público ao seu bucket do S3. Quando você concede acesso a esse grupo, qualquer pessoa no mundo pode acessar seu bucket. É altamente recomendável que você nunca conceda nenhum tipo de acesso público de gravação ao seu bucket do S3.

7. Para conceder ou cancelar as permissões de qualquer pessoa com uma Conta da AWS, ao lado de Authenticated Users group (anyone with an Conta da AWS) [Grupo de usuários autenticados (qualquer pessoa com uma Conta da AWS)], selecione ou desmarque uma das seguintes permissões de ACL:
- Objetos: listar
 - ACL de bucket: ler
8. Para conceder ou desfazer permissões para o Amazon S3 gravar logs de acesso ao servidor no bucket, em S3 log delivery group (Grupo de entrega de log do S3), limpe ou selecione uma das seguintes permissões ACL:
- Objetos: listar ou gravar
 - ACL de bucket: ler ou gravar

Se um bucket for configurado como o bucket de destino para receber logs de acesso, as permissões do bucket devem permitir ao grupo Log Delivery (Entrega de logs) acesso de gravação ao bucket. Quando você ativa o registro em log do acesso ao servidor em um bucket, o console do Amazon S3 concede acesso de gravação ao grupo Log Delivery (Entrega de logs) para o bucket de destino do qual você opta por receber os logs. Para obter mais informações sobre o registro em log de acesso ao servidor, consulte [Habilitar o log de acesso ao servidor do Amazon S3 \(p. 1043\)](#).

9. Para conceder acesso a outra Conta da AWS, faça o seguinte:
- a. Escolha Add grantee (Adicionar beneficiário).
 - b. Na caixa Grantee (Beneficiário), insira o ID canônico da outra Conta da AWS.
 - c. Selecione uma das seguintes permissões de ACL:

- Objetos: listar ou gravar
- ACL de bucket: ler ou gravar

Warning

Ao conceder aos recursos acesso a outras Contas da AWS, esteja ciente de que as Contas da AWS podem delegar as permissões delas a usuários de suas próprias contas. Isso é conhecido como acesso entre contas. Para obter informações sobre como usar o acesso entre contas, consulte [Criar uma função para delegar permissões a um usuário do IAM](#) no Guia do usuário do IAM.

10. Para remover o acesso a outra Conta da AWS, em Access for other Contas da AWS (Acesso para outras Contas da AWS), selecione Remove (Remover).
11. Para salvar suas alterações, escolha Save changes (Salvar alterações).

[Usar o console do S3 para definir permissões de ACL para um objeto](#)

O console exibe concessões de acesso combinadas para favorecidos duplicados. Para ver a lista completa de ACLs, use a API REST do Amazon S3, a AWS CLI ou os AWS SDKs. A tabela a seguir mostra as permissões de ACL que você pode configurar para objetos no console do Amazon S3.

Permissões de ACL do console do Amazon S3 para objetos

Permissão do console	Permissão da ACL	Acesso
Objeto - Ler	READ	Permite ao favorecido ler os dados do objeto e seus metadados.
ACL do objeto - ler	READ_ACP	Permite ao favorecido ler a ACL do objeto
ACL Objeto - gravar	WRITE_ACP	Permite ao favorecido gravar a ACL para o objeto aplicável

Para obter mais informações sobre permissões de ACL, consulte [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#).

Important

Se o bucket usar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Solicitações para definir ACLs ou atualizar ACLs falham e retornam o código de erro `AccessControlListNotSupported`. Ainda há suporte para solicitações de leitura de ACLs.

Para definir permissões de ACL para um objeto

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket que contém o objeto.
3. Na lista Name (Nome), escolha o nome do objeto para o qual você deseja definir permissões.
4. Escolha Permissions (Permissões).
5. Na lista do controle de acesso (ACL), escolha Edit (Editar).

Você pode editar as seguintes permissões de ACL para o objeto:

Objeto

- Read (Ler): permite ao favorecido ler os dados do objeto e seus metadados.

ACL do objeto

- Read (Ler): permite ao favorecido ler a ACL do objeto.
- Write (Gravar): permite ao favorecido gravar a ACL para o objeto aplicável. No console do S3, você só pode conceder acesso de gravação ao proprietário do bucket (sua Conta da AWS). É recomendável não conceder acesso de gravação a outros favorecidos. No entanto, se você precisar conceder acesso de gravação, poderá usar a AWS CLI, os AWS SDKs ou a API REST.

6. É possível gerenciar as permissões de acesso do objeto ao seguinte:

- a. Acesso para o proprietário do objeto

O proprietário refere-se ao usuário root da Conta da AWS, e não um usuário do AWS Identity and Access Management (IAM). Para obter mais informações sobre o usuário root, consulte [Usuário root da Conta da AWS](#) no Guia do usuário do IAM.

Para alterar as permissões de acesso ao objeto do proprietário, em Access for object owner (Acesso para o proprietário do objeto), selecione Your AWS Account (owner) [Sua conta da AWS (proprietário)].

Marque as caixas de seleção das permissões que você deseja alterar e escolha Save (Salvar).

- b. Acesso para outras Contas da AWS

Para conceder permissões a um usuário da AWS de uma Conta da AWS diferente, em Access for other Contas da AWS (Acesso para outras Contas da AWS), selecione Add account (Adicionar conta). No campo Enter an ID (Inserir um ID), digite o ID canônico do usuário da AWS ao qual você deseja conceder permissões para o objeto. Para obter informações sobre como encontrar um ID canônico, consulte [Identificadores de sua Conta da AWS](#) na Referência geral da Amazon Web Services. Você pode adicionar até 99 usuários.

Marque as caixas de seleção das permissões que você deseja conceder ao usuário e escolha Save (Salvar). Para exibir informações sobre as permissões, escolha os ícones da Ajuda.

- c. Acesso público

Para conceder acesso ao seu objeto para o público geral (todos no mundo), em Public access (Acesso público), escolha Everyone (Todos). Conceder permissões de acesso público significa que todos no mundo podem acessar o objeto.

Marque as caixas de seleção das permissões que você deseja conceder e escolha Save (Salvar).

Warning

- Tenha cuidado ao conceder ao grupo Everyone (Todos) acesso anônimo aos seus objetos do Amazon S3. Quando você concede acesso a esse grupo, qualquer pessoa no mundo pode acessar seu objeto. Se você precisar conceder acesso a todos, é altamente recomendável que só conceda permissões para Read objects (Ler objetos).
- Recomendamos fortemente que você não conceda ao grupo Everyone (Todos) permissões de gravação no objeto. Isso permite que qualquer pessoa substitua as permissões da ACL para o objeto.

Uso de AWS SDKs

Esta seção fornece exemplos de como configurar concessões na lista de controle de acesso (ACL) em buckets e objetos.

Important

Se o bucket usar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Solicitações para definir ACLs ou atualizar ACLs falham e retornam o código de erro `AccessControlListNotSupported`. Ainda há suporte para solicitações de leitura de ACLs.

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 \(p. 594\)](#)), 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()`.

Example Crie um bucket e especifique uma ACL pré-configurada que conceda permissão ao grupo de entrega de logs do S3

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.

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

import java.io.IOException;
import java.util.ArrayList;

public class CreateBucketWithACL {

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

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

            // Create a bucket with a canned ACL. This ACL will be replaced by the
            setBucketAcl()
            // calls below. It is included here for demonstration purposes.
            CreateBucketRequest createBucketRequest = new
            CreateBucketRequest(bucketName, clientRegion.getName())
                .withCannedAcl(CannedAccessControlList.LogDeliveryWrite);
            s3Client.createBucket(createBucketRequest);

            // Create a collection of grants to add to the bucket.
            ArrayList<Grant> grantCollection = new ArrayList<Grant>();

            // Grant the account owner full control.
            Grant grant1 = new Grant(new
            CanonicalGrantee(s3Client.gets3AccountOwner().getId()), Permission.FullControl);

        } catch (AmazonServiceException | SdkClientException e) {
            System.out.println("Caught an error creating the bucket: " + e.getMessage());
        }
    }
}
```

```
grantCollection.add(grant1);

// Grant the LogDelivery group permission to write to the bucket.
Grant grant2 = new Grant(GroupGrantee.LogDelivery, Permission.Write);
grantCollection.add(grant2);

// Save grants by replacing all current ACL grants with the two we just
created.
AccessControlList bucketAcl = new AccessControlList();
bucketAcl.grantAllPermissions(grantCollection.toArray(new Grant[0]));
s3Client.setBucketAcl(bucketName, bucketAcl);

// Retrieve the bucket's ACL, add another grant, and then save the new ACL.
AccessControlList newBucketAcl = s3Client.getBucketAcl(bucketName);
Grant grant3 = new Grant(new
EmailAddressGrantee(userEmailForReadPermission), Permission.Read);
newBucketAcl.grantAllPermissions(grant3);
s3Client.setBucketAcl(bucketName, newBucketAcl);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Example Atualizar ACL em um objeto existente

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? \(p. 591\)](#)) para o usuário identificado por endereço de e-mail
- Salva a ACL no objeto

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.AccessControlList;
import com.amazonaws.services.s3.model.CanonicalGrantee;
import com.amazonaws.services.s3.model.EmailAddressGrantee;
import com.amazonaws.services.s3.model.Permission;

import java.io.IOException;

public class ModifyACLExistingObject {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ***";
        String emailGrantee = "*** user@example.com ***";
```

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

    // Get the existing object ACL that we want to modify.
    AccessControlList acl = s3Client.getObjectAcl(bucketName, keyName);

    // Clear the existing list of grants.
    acl.getGrantsAsList().clear();

    // Grant a sample set of permissions, using the existing ACL owner for Full
    Control permissions.
    acl.grantPermission(new CanonicalGrantee(acl.getOwner().getId()),
Permission.FullControl);
    acl.grantPermission(new EmailAddressGrantee(emailGrantee),
Permission.WriteAcp);

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

.NET

Example Crie um bucket e especifique uma ACL pré-configurada que conceda permissão ao grupo de entrega de logs do S3

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 como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
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());
            }
        }
    }
}
```

Example Atualizar 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 como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class ManagingObjectACLTTest
```

```
{  
    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());
        }
    }
}
```

Uso dos REST API

As APIs do Amazon S3 permitem a definição de uma ACL quando você cria um bucket ou um objeto. O Amazon S3 também fornece API para definir uma ACL em um bucket ou 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 informações sobre o suporte à API REST para gerenciar ACLs, consulte as seções a seguir na Referência da API do Amazon Simple Storage Service:

- [GET Bucket acl](#)
- [acl de PUT Bucket](#)
- [GET Object acl](#)
- [Put Object ACL \(ACL de objeto PUT\)](#)
- [Objeto PUT](#)
- [Bucket PUT](#)
- [Objeto PUT - Copiar](#)
- [Iniciar multipart upload](#)

Important

Se o bucket usar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Solicitações para definir ACLs ou atualizar ACLs falham e retornam o código de erro `AccessControlListNotSupported`. Ainda há suporte para solicitações de leitura de ACLs.

[Lista de controle de acesso \(ACL - Access Control List\) - Cabeçalhos específicos de solicitação](#)

Você pode usar cabeçalhos para conceder permissões baseadas em lista de controle de acesso (ACL). Por padrão, todos os objetos são privados. Somente o proprietário tem controle total de acesso. Ao adicionar um novo objeto, você pode conceder permissões a Contas da AWS individuais ou a grupos predefinidos pelo Amazon S3. Depois essas permissões são adicionadas à lista de controle de acesso (ACL) no objeto. Para obter mais informações, consulte [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#).

Com esta operação, você pode conceder permissões de acesso usando um destes dois métodos:

- **Canned ACL (`x-amz-acl`) (ACL pré-configurada)** — O Amazon S3 oferece suporte a um conjunto de ACLs predefinidas, conhecidas como ACLs pré-configuradas. Cada ACL pré-configurada tem um conjunto predefinido de concessões e permissões. Para obter mais informações, consulte [ACL pré-configurada \(p. 594\)](#).

- Access Permissions (Permissões de acesso): para conceder explicitamente permissões de acesso a grupos ou Contas da AWS específicos, use os seguintes cabeçalhos. Cada cabeçalho mapeia para permissões específicas compatíveis com o Amazon S3 em uma ACL. Para obter mais informações, consulte [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#). No cabeçalho, você especifica uma lista de favorecidos que obtêm a permissão específica.
 - x-amz-grant-read
 - x-amz-grant-write
 - x-amz-grant-read-acp
 - x-amz-grant-write-acp
 - x-amz-grant-full-control

Uso do AWS CLI

Para obter mais informações sobre como gerenciar ACLs usando a AWS CLI, consulte [put-bucket-acl](#) na Referência de comandos da AWS CLI.

Important

Se o bucket usar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Solicitações para definir ACLs ou atualizar ACLs falham e retornam o código de erro `AccessControlListNotSupported`. Ainda há suporte para solicitações de leitura de ACLs.

Usar o compartilhamento de recursos de origem cruzada (CORS)

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 o suporte do CORS, você pode criar aplicações web no lado do cliente com o Amazon S3 e permitir seletivamente o acesso de origem cruzada aos seus recursos do Amazon S3.

Esta seção fornece uma visão geral do CORS. Os subtópicos descrevem como você pode ativar o CORS usando o console do Amazon S3 ou, programaticamente, usando a API REST do Amazon S3 e os AWS SDKs.

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 usando o Amazon S3 \(p. 1188\)](#). 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.us-east-1.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 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 de uma 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.

Para obter mais informações sobre como usar o CORS, consulte os tópicos a seguir.

Tópicos

- [Configuração de CORS \(p. 608\)](#)
- [Configurar o compartilhamento de recursos de origem cruzada \(CORS\) \(p. 612\)](#)
- [Solução de problemas do CORS \(p. 618\)](#)

Configuração de CORS

Para configurar seu bucket para permitir solicitações de origem cruzada, crie uma configuração CORS. A configuração CORS é um documento com regras que identificam as origens que você permitirá que acessem seu bucket, as operações (métodos HTTP) compatíveis com cada origem e outras informações específicas da operação. Você pode adicionar até 100 regras à configuração. Você pode adicionar a configuração CORS como o sub-recurso `cors` ao bucket.

Se você estiver configurando o CORS no console do S3, use o JSON para criar uma configuração CORS. O novo console do S3 oferece suporte somente a configurações JSON CORS.

Para obter mais informações sobre a configuração de CORS e os elementos nele, consulte os tópicos a seguir. Para obter instruções sobre como adicionar uma configuração de CORS, consulte [Configurar o compartilhamento de recursos de origem cruzada \(CORS\) \(p. 612\)](#).

Important

No novo console do S3, a configuração CORS deve ser JSON.

Tópicos

- [Exemplo 1 \(p. 609\)](#)

- [Exemplo 2 \(p. 610\)](#)
- [Elemento AllowedMethod \(p. 611\)](#)
- [Elemento AllowedOrigin \(p. 611\)](#)
- [Elemento AllowedHeader \(p. 611\)](#)
- [Elemento ExposeHeader \(p. 612\)](#)
- [Elemento MaxAgeSeconds \(p. 612\)](#)

Exemplo 1

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 [Configurar um site estático usando um domínio personalizado registrado no Route 53 \(p. 105\)](#).

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.

JSON

```
[  
  {  
    "AllowedHeaders": [  
      "*"  
    ],  
    "AllowedMethods": [  
      "PUT",  
      "POST",  
      "DELETE"  
    ],  
    "AllowedOrigins": [  
      "http://www.example1.com"  
    ],  
    "ExposeHeaders": []  
  },  
  {  
    "AllowedHeaders": [  
      "*"  
    ],  
    "AllowedMethods": [  
      "PUT",  
      "POST",  
      "DELETE"  
    ],  
    "AllowedOrigins": [  
      "http://www.example2.com"  
    ],  
    "ExposeHeaders": []  
  },  
  {  
    "AllowedHeaders": [],  
    "AllowedMethods": [  
      "GET"  
    ],  
    "AllowedOrigins": ["*"],  
    "ExposeHeaders": []  
  }]
```

```
    "AllowedMethods": [
        "GET"
    ],
    "AllowedOrigins": [
        "*"
    ],
    "ExposeHeaders": []
}
]
```

XML

```
<corsConfiguration>
<corsRule>
    <allowedOrigin>http://www.example1.com</allowedOrigin>

    <allowedMethod>PUT</allowedMethod>
    <allowedMethod>POST</allowedMethod>
    <allowedMethod>DELETE</allowedMethod>

    <allowedHeader>*</allowedHeader>
</corsRule>
<corsRule>
    <allowedOrigin>http://www.example2.com</allowedOrigin>

    <allowedMethod>PUT</allowedMethod>
    <allowedMethod>POST</allowedMethod>
    <allowedMethod>DELETE</allowedMethod>

    <allowedHeader>*</allowedHeader>
</corsRule>
<corsRule>
    <allowedOrigin>*</allowedOrigin>
    <allowedMethod>GET</allowedMethod>
</corsRule>
</corsConfiguration>
```

Exemplo 2

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>.

JSON

```
[
    {
        "allowedHeaders": [
            "*"
        ],
        "allowedMethods": [
            "PUT",
            "POST",
            "DELETE"
        ],
        "allowedOrigins": [
            "http://www.example.com"
        ],
        "exposeHeaders": [
            "x-amz-server-side-encryption",
            "x-amz-request-id",
            "x-amz-id-2"
        ]
    }
]
```

```
        ],
        "MaxAgeSeconds": 3000
    }
]
```

XML

```
<corsConfiguration>
  <corsRule>
    <allowedOrigin>http://www.example.com</allowedOrigin>
    <allowedMethod>PUT</allowedMethod>
    <allowedMethod>POST</allowedMethod>
    <allowedMethod>DELETE</allowedMethod>
    <allowedHeader>*</allowedHeader>
    <maxAgeSeconds>3000</maxAgeSeconds>
    <exposeHeader>x-amz-server-side-encryption</
      exposeHeader>
      <exposeHeader>x-amz-request-id</
        exposeHeader>
        <exposeHeader>x-amz-id-2</exposeHeader>
    </corsRule>
  </corsConfiguration>
```

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 coincidir com uma entrada correspondente na regra. O Amazon S3 enviará somente os cabeçalhos permitidos que foram solicitados em uma resposta. Para obter uma lista de exemplos de cabeçalhos que podem ser usados em solicitações para o Amazon S3, acesse [Cabeçalhos de solicitação comuns](#) no guia de Referência da API do Amazon Simple Storage Service.

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 obter uma lista de cabeçalhos de resposta comuns do Amazon S3, acesse [Cabeçalhos de resposta comuns](#) no guia de Referência da API do Amazon Simple Storage Service .

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.

Configurar o compartilhamento de recursos de origem cruzada (CORS)

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 o suporte do CORS, você pode criar aplicações web no lado do cliente com o Amazon S3 e permitir seletivamente o acesso de origem cruzada aos seus recursos do Amazon S3.

Esta seção mostra como habilitar o CORS usando o console do Amazon S3, a API REST do Amazon S3 e os AWS SDKs. Para configurar seu bucket para permitir solicitações de origem cruzada, adicione uma configuração de CORS ao bucket. Uma configuração CORS é um documento que define regras que identificam as origens que você permitirá que acessem seu bucket, as operações (métodos HTTP) compatíveis para cada origem e outras informações específicas da operação. No console do S3, a configuração CORS deve ser um documento JSON.

Para obter exemplos de configurações de CORS em JSON e XML, consulte [Configuração de CORS \(p. 608\)](#).

Uso do console do S3

Esta seção explica como usar o console do Amazon S3 para adicionar uma configuração de compartilhamento de recursos de origem cruzada (CORS) para um bucket do S3.

Quando você permitir o CORS no bucket, as listas de controle de acesso (ACLs) e outras políticas de permissão de acesso continuarão sendo aplicadas.

Important

No novo console do S3, a configuração CORS deve ser JSON. Para obter exemplos de configurações de CORS em JSON e XML, consulte [Configuração de CORS \(p. 608\)](#).

Para adicionar uma configuração CORS a um bucket do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket para o qual você deseja criar uma política de bucket.

3. Escolha Permissions (Permissões).
4. Na seção Cross-origin resource sharing (CORS) (Compartilhamento de recursos de origem cruzada (CORS)) escolha Edit (Editar).
5. Na caixa de texto CORS configuration editor (Editor de configuração CORS), digite ou copie e cole uma nova configuração CORS, ou edite uma configuração existente.

A configuração de CORS é um arquivo JSON. O texto que você digita no editor deve ser um JSON válido. Para obter mais informações, consulte [Configuração de CORS \(p. 608\)](#).

6. Selecione Save changes.

Note

O Amazon S3 exibe o Nome de recurso da Amazon (ARN) para o bucket próximo ao título CORS configuration editor (Editor de configuração CORS). Para obter mais informações sobre ARNs, [Nomes de recursos da Amazon \(ARNs\)](#) e [Namespaces de serviços da AWS](#) na Referência geral da Amazon Web Services.

Uso de AWS SDKs

Você pode usar o AWS SDK para gerenciar o Cross-Origin Resource Sharing (CORS – Compartilhamento de recursos entre origens) para um bucket. Para obter mais informações sobre CORS, consulte [Usar o compartilhamento de recursos de origem cruzada \(CORS\) \(p. 607\)](#).

Veja os seguintes exemplos:

- 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

Java

Example

Example

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

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

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

public class CORS {

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

```
Regions clientRegion = Regions.DEFAULT_REGION;
String bucketName = "*** Bucket name ***";

// Create two CORS rules.
List<CORSRule.AllowedMethods> rule1AM = new
ArrayList<CORSRule.AllowedMethods>();
rule1AM.add(CORSRule.AllowedMethods.PUT);
rule1AM.add(CORSRule.AllowedMethods.POST);
rule1AM.add(CORSRule.AllowedMethods.DELETE);
CORSRule rule1 = new CORSRule().withId("CORSRule1").withAllowedMethods(rule1AM)
.withAllowedOrigins(Arrays.asList("http://*.example.com"));

List<CORSRule.AllowedMethods> rule2AM = new
ArrayList<CORSRule.AllowedMethods>();
rule2AM.add(CORSRule.AllowedMethods.GET);
CORSRule rule2 = new CORSRule().withId("CORSRule2").withAllowedMethods(rule2AM)
.withAllowedOrigins(Arrays.asList("*")).withMaxAgeSeconds(3000)
.withExposedHeaders(Arrays.asList("x-amz-server-side-encryption"));

List<CORSRule> rules = new ArrayList<CORSRule>();
rules.add(rule1);
rules.add(rule2);

// Add the rules to a new CORS configuration.
BucketCrossOriginConfiguration configuration = new
BucketCrossOriginConfiguration();
configuration.setRules(rules);

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

    // Add the configuration to the bucket.
    s3Client.setBucketCrossOriginConfiguration(bucketName, configuration);

    // Retrieve and display the configuration.
    configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
    printCORSConfiguration(configuration);

    // Add another new rule.
    List<CORSRule.AllowedMethods> rule3AM = new
ArrayList<CORSRule.AllowedMethods>();
    rule3AM.add(CORSRule.AllowedMethods.HEAD);
    CORSRule rule3 = new
CORSRule().withId("CORSRule3").withAllowedMethods(rule3AM)
.withAllowedOrigins(Arrays.asList("http://www.example.com"));

    rules = configuration.getRules();
    rules.add(rule3);
    configuration.setRules(rules);
    s3Client.setBucketCrossOriginConfiguration(bucketName, configuration);

    // Verify that the new rule was added by checking the number of rules in
    // the configuration.
    configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
    System.out.println("Expected # of rules = 3, found " +
configuration.getRules().size());

    // Delete the configuration.
    s3Client.deleteBucketCrossOriginConfiguration(bucketName);
    System.out.println("Removed CORS configuration.");

    // Retrieve and display the configuration to verify that it was
    // successfully deleted.
```

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

private static void printCORSConfiguration(BucketCrossOriginConfiguration
configuration) {
    if (configuration == null) {
        System.out.println("Configuration is null.");
    } else {
        System.out.println("Configuration has " + configuration.getRules().size() +
" rules\n");

        for (CORSRule rule : configuration.getRules()) {
            System.out.println("Rule ID: " + rule.getId());
            System.out.println("MaxAgeSeconds: " + rule.getMaxAgeSeconds());
            System.out.println("AllowedMethod: " + rule.getAllowedMethods());
            System.out.println("AllowedOrigins: " + rule.getAllowedOrigins());
            System.out.println("AllowedHeaders: " + rule.getAllowedHeaders());
            System.out.println("ExposeHeader: " + rule.getExposedHeaders());
            System.out.println();
        }
    }
}
```

.NET

Example

Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

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

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            CORSConfigTestAsync().Wait();
        }
        private static async Task CORSConfigTestAsync()
```

```
{  
    try  
    {  
        // Create a new configuration request and add two rules  
        CORSConfiguration configuration = new CORSConfiguration  
        {  
            Rules = new System.Collections.Generic.List<CORSRule>  
            {  
                new CORSRule  
                {  
                    Id = "CORSRule1",  
                    AllowedMethods = new List<string> {"PUT", "POST",  
"DELETE"},  
                    AllowedOrigins = new List<string> {"http://*.example.com"}  
                },  
                new CORSRule  
                {  
                    Id = "CORSRule2",  
                    AllowedMethods = new List<string> {"GET"},  
                    AllowedOrigins = new List<string> {"*"},  
                    MaxAgeSeconds = 3000,  
                    ExposeHeaders = new List<string> {"x-amz-server-side-  
encryption"}  
                }  
            };  
  
            // Add the configuration to the bucket.  
            await PutCORSConfigurationAsync(configuration);  
  
            // Retrieve an existing configuration.  
            configuration = await RetrieveCORSConfigurationAsync();  
  
            // Add a new rule.  
            configuration.Rules.Add(new CORSRule  
            {  
                Id = "CORSRule3",  
                AllowedMethods = new List<string> { "HEAD" },  
                AllowedOrigins = new List<string> { "http://www.example.com" }  
            });  
  
            // Add the configuration to the bucket.  
            await PutCORSConfigurationAsync(configuration);  
  
            // Verify that there are now three rules.  
            configuration = await RetrieveCORSConfigurationAsync();  
            Console.WriteLine();  
            Console.WriteLine("Expected # of 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()));
        }
    }
}
```

```
}
```

Uso dos REST API

Para definir uma configuração de CORS no bucket, use o AWS Management Console. Se o seu aplicativo exigir, você também pode enviar solicitações REST diretamente. As seções a seguir na Referência da API do Amazon Simple Storage Service descrevem as ações da API REST relacionadas à configuração CORS:

- [PutBucketCors](#)
- [GetBucketCors](#)
- [DeleteBucketCors](#)
- 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.

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 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 dos 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.

Bloquear o acesso público ao armazenamento do Amazon S3

O recurso Bloqueio de acesso público do Amazon S3 fornece configurações para pontos de acesso, buckets e contas para ajudar você a gerenciar o acesso público aos recursos do Amazon S3. Por padrão, novos buckets, pontos de acesso e objetos não permitem acesso público. No entanto, os usuários podem

modificar políticas de bucket, políticas de ponto de acesso ou permissões de objeto para permitir acesso público. As configurações do Bloqueio de acesso público do S3 substituem essas políticas e permissões, de maneira que seja possível limitar o acesso público a esses recursos.

Com o Bloqueio de acesso público do S3, os administradores de conta e os proprietários de bucket podem configurar facilmente os controles centralizados impostos para limitar o acesso público aos recursos do Amazon S3, independentemente de como os recursos são criados.

Para obter instruções sobre como configurar o acesso público a blocos, consulte [Configurar o bloqueio de acesso público \(p. 625\)](#).

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 Bloqueio de acesso público aplicada. Se a solicitação foi feita por meio de um ponto de acesso, o Amazon S3 também verificará se há configurações do Bloqueio de acesso público para o ponto de acesso. Caso haja uma configuração do Bloqueio de acesso público que proiba o acesso solicitado, o Amazon S3 rejeitará a solicitação.

O Bloqueio de acesso público do Amazon S3 fornece quatro configurações. Essas configurações são independentes e podem ser usadas em qualquer combinação. Cada configuração pode ser aplicada a um ponto de acesso, a um bucket ou a uma Conta da AWS inteira. Se as configurações do Bloqueio de acesso público para o ponto de acesso, o bucket ou a conta forem diferentes, o Amazon S3 aplicará a combinação mais restritiva das configurações de ponto de acesso, bucket e conta.

Quando o Amazon S3 avalia se uma operação é proibida por uma configuração do Bloqueio de acesso público, ela rejeita qualquer solicitação que viole uma configuração de ponto de acesso, bucket ou conta.

Warning

O acesso público aos buckets e objetos é concedido através de listas de controle de acesso (ACLs), políticas de ponto de acesso, políticas de bucket ou todos. Para garantir que todos os seus pontos de acesso, buckets e objetos do Amazon S3 tenham o acesso público bloqueado, recomendamos ativar as quatro configurações de bloqueio de acesso público na sua conta.

Estas configurações bloqueiam o acesso público a todos os buckets e pontos de acesso atuais e futuros.

Antes de aplicar estas configurações, verifique se seus aplicativos funcionarão corretamente sem acesso público. Se você precisa de um certo nível de acesso público aos seus buckets ou objetos, por exemplo, para hospedar um site estático como descrito em [Hospedagem de um site estático usando o Amazon S3 \(p. 1188\)](#), é possível personalizar as configurações individualmente conforme cada caso de uso de armazenamento.

Note

- Você pode habilitar as configurações de bloqueio de acesso público somente para pontos de acesso, buckets e Contas da AWS. O Amazon S3 não oferece suporte a configurações de bloqueio de acesso público por objeto.
- Quando você aplica configurações de bloqueio de acesso público 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 bloqueio de acesso público \(p. 620\)](#)
- [Executar operações de bloqueio de acesso público em um ponto de acesso \(p. 622\)](#)
- [O significado de "público" \(p. 622\)](#)
- [Usar o Analisador de acesso para S3 para revisar buckets públicos \(p. 624\)](#)
- [Permissões \(p. 624\)](#)

- [Configurar o bloqueio de acesso público \(p. 625\)](#)
- [Configurar o bloqueio de acesso público para sua conta \(p. 625\)](#)
- [Configurar o bloqueio de acesso público para seus buckets do S3 \(p. 627\)](#)

Configurações do bloqueio de acesso público

O Bloqueio de acesso público do S3 fornece quatro configurações. É possível aplicar essas configurações em qualquer combinação a pontos de acesso individuais, buckets ou Contas da AWS inteiras. Caso você aplique uma configuração a uma conta, ela se aplica a todos os buckets e pontos de acesso de propriedade dessa conta. Da mesma forma, se você aplicar uma configuração a um bucket, ela se aplicará a todos os pontos de acesso associados a esse bucket.

A tabela a seguir contém as configurações disponíveis.

Nome	Descrição
<code>BlockPublicAcls</code>	<p>A definição dessa opção como <code>TRUE</code> causa o seguinte comportamento:</p> <ul style="list-style-type: none">• As chamadas <code>PUT Bucket acl</code> e <code>PUT Object</code> falharão se a Access Control List (ACL – Lista de controle de acesso) especificada for pública.• As chamadas <code>PUT Object</code> falharão se a solicitação incluir uma ACL pública.• Se essa configuração for aplicada a uma conta, as chamadas <code>PUT Bucket</code> falharão se a solicitação incluir uma ACL pública. <p>Quando essa configuração for definida como <code>TRUE</code>, as operações especificadas falharão (quer por meio da API REST, da AWS CLI ou dos AWS SDKs). 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.</p> <p>Note</p> <p>Os pontos de acesso não têm ACLs associadas a eles. Se você aplicar essa configuração a um ponto de acesso, ela atuará como uma passagem para o bucket subjacente. Se um ponto de acesso tiver essa configuração ativada, as solicitações feitas por meio do ponto de acesso se comportam como se o bucket subjacente tivesse essa configuração ativada, independentemente de o bucket realmente ter essa configuração ativada.</p>
<code>IgnorePublicAcls</code>	<p>A definição dessa opção como <code>TRUE</code> 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 <code>PUT Object</code> que incluem uma ACL pública (ao contrário de <code>BlockPublicAcls</code>, que rejeita chamadas <code>PUT Object</code> que incluem uma ACL pública). A habilitação dessa configuração não afeta a persistência de ACLs existentes nem evita a definição de novas ACLs públicas.</p> <p>Note</p> <p>Os pontos de acesso não têm ACLs associadas a eles. Se você aplicar essa configuração a um ponto de acesso, ela atuará como uma passagem para o bucket subjacente. Se um ponto de acesso tiver essa configuração ativada, as solicitações feitas por meio do ponto</p>

Nome	Descrição
	<p>de acesso se comportam como se o bucket subjacente tivesse essa configuração ativada, independentemente de o bucket realmente ter essa configuração ativada.</p>
BlockPublicPolicy	<p>Definir esta opção como TRUE para um bucket faz com que o Amazon S3 rejeite chamadas para a política PUT de bucket, se a política de bucket especificada permitir acesso público, e rejeite chamadas para a política PUT de ponto de acesso para todos os pontos de acesso do bucket, se a política especificada permitir acesso público. Definir esta opção como TRUE para um ponto de acesso faz com que o Amazon S3 rejeite chamadas para a política PUT de ponto de acesso e a política PUT de bucket que são feitas por meio do ponto de acesso, se a política especificada (para o ponto de acesso ou para o bucket subjacente) for pública.</p> <p>Essa configuração permite que os usuários gerenciem políticas de ponto de acesso e bucket sem permitir que compartilhem publicamente o bucket ou os objetos contidos. Habilitar essa configuração não afeta as políticas de ponto de acesso ou de bucket existentes.</p> <p>Important</p> <p>Para usar essa configuração de maneira efetiva, aplique-a no nível da conta. Uma política de bucket pode permitir que os usuários alterem as configurações do Bloqueio de acesso público de um bucket. Portanto, os usuários com permissão para alterar a política de bucket podem inserir uma política que os permita desabilitar as configurações do Bloqueio de acesso público 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.</p>
RestrictPublicBucketAccess	<p>A definição dessa opção como TRUE restringe o acesso a um ponto de acesso ou a um bucket com uma política pública apenas a principais de serviços da AWS e a usuários autorizados dentro da conta do proprietário do bucket. Essa definição bloqueia todo o acesso entre contas ao ponto de acesso ou ao bucket (exceto por principais de serviços da AWS), ao mesmo tempo que continua permitindo que usuários dentro da conta gerenciem o ponto de acesso ou o bucket.</p> <p>A ativação dessa configuração não afeta políticas de ponto de acesso ou bucket existentes, exceto se o Amazon S3 bloquear os acessos público e entre contas derivados de qualquer política pública de ponto de acesso ou bucket, inclusive delegação não pública a contas específicas.</p>

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.
- As configurações do Bloqueio de acesso público não alteram as políticas ou ACLs existentes. Portanto, a remoção de uma configuração do Bloqueio de acesso público disponibiliza novamente um bucket ou um objeto com uma política pública ou uma ACL.

Executar operações de bloqueio de acesso público em um ponto de acesso

Para realizar operações do Block Public Access em um ponto de acesso, use o serviço s3control da AWS CLI.

Important

Observe que não é possível alterar as configurações do Bloqueio de acesso público de um ponto de acesso após ele ser criado. Assim, a única maneira de especificar configurações do Bloqueio de acesso público para um ponto de acesso é incluí-las ao criar o ponto de acesso.

O significado de "público"

Buckets

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 \(p. 590\)](#).

Políticas

Ao avaliar uma política de bucket, o Amazon S3 começa presumindo que a política é pública. Em seguida, ele avalia a política para determinar se ela se qualifica como não pública. Para ser considerada não pública, uma política de bucket deve conceder acesso apenas a valores fixos (valores que não contenham um curinga nem uma variável de política do AWS Identity and Access Management) de um ou mais dos seguintes itens:

- 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.
- Um principal, um usuário, uma função ou um principal do produto da AWS (por exemplo, aws:PrincipalOrgID)
- 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:*
- s3:DataAccessPointArn

Note

Quando usado em uma política de bucket, esse valor pode conter um curinga para o nome do ponto de acesso sem tornar a política pública, desde que o ID da conta seja corrigido. Por exemplo, permitir acesso a arn:aws:s3:us-west-2:123456789012:accesspoint/* permitiria o acesso a qualquer ponto de acesso associado à conta 123456789012 na região us-west-2, sem tornar pública a política de bucket. Observe que esse comportamento é diferente para políticas de ponto de acesso. Para obter mais informações, consulte [Pontos de acesso \(p. 624\)](#).

- s3:DataAccessPointAccount

Nessas regras, as políticas de exemplo a seguir 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-*" } }
}
```

É possível tornar essas políticas 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 o seguinte.

```
{
"Principal": "*",
"Resource": "*",
>Action": "s3:PutObject",
"Effect": "Allow",
"Condition": { "StringEquals": { "aws:SourceVpc": "vpc-91237329" } }
}
```

Para obter mais informações sobre políticas de bucket, consulte [Políticas de bucket e políticas de usuário \(p. 437\)](#).

Exemplo

Este exemplo mostra como o Amazon S3 avalia uma política de bucket 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. No entanto, se você adicionar uma instrução pública à política, o `RestrictPublicBuckets` será ativado no bucket. Isso permite que somente principais de serviço da AWS e usuários autorizados da conta do proprietário do bucket acessem o 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 implementada e `RestrictPublicBuckets` ativado, o Amazon S3 permite o acesso somente pelo CloudTrail. Embora a instrução 2 não seja pública, o Amazon S3 desabilita o acesso de "Account-2". Isso porque a instrução 3 renderiza toda a política pública. Assim, `RestrictPublicBuckets` se aplica. Dessa forma, o Amazon 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.

Pontos de acesso

O Amazon S3 avalia as configurações do Bloqueio de acesso público de maneira um pouco diferente para os pontos de acesso em comparação com os buckets. As regras que o Amazon S3 aplica para determinar quando uma política de ponto de acesso é pública são geralmente as mesmas para pontos de acesso e para buckets, exceto nas seguintes situações:

- Um ponto de acesso que tenha uma origem de rede da VPC é sempre considerado não público, independentemente do conteúdo da política de ponto de acesso.
- Uma política de ponto de acesso que concede acesso a um conjunto de pontos de acesso usando `s3:DataAccessPointArn` é considerada pública. Observe que esse comportamento é diferente para políticas de bucket. Por exemplo, uma política de bucket que concede acesso a valores de `s3:DataAccessPointArn` que correspondem a `arn:aws:s3:us-west-2:123456789012:accesspoint/*` não é considerada pública. No entanto, a mesma instrução em uma política de ponto de acesso tornaria o ponto de acesso público.

Usar o Analisador de acesso para S3 para revisar buckets públicos

Você pode usar o Analisador de acesso para S3 para revisar buckets com ACLs de bucket, políticas de bucket ou políticas de ponto de acesso que concedem acesso público. O Access Analyzer for S3 alerta você sobre buckets que são configurados para permitir o acesso a qualquer pessoa na Internet ou a outras Contas da AWS, incluindo Contas da AWS fora da organização. Para cada bucket público ou compartilhado, você recebe descobertas que relatam a origem e o nível de acesso público ou compartilhado.

Com o conhecimento apresentado nas descobertas, você pode tomar medidas corretivas imediatas e precisas. No Analisador de acesso para S3, você pode bloquear todo o acesso público a um bucket com um único clique. Você também pode detalhar as configurações de permissão no nível do bucket para definir níveis granulares de acesso. Para casos de uso específicos e verificados que exigem acesso público ou compartilhado, você pode confirmar e registrar sua intenção de que o bucket permaneça público ou compartilhado arquivando as descobertas do bucket.

Em eventos raros, o Analisador de acesso para S3 pode relatar nenhuma descoberta para um bucket que uma análise de bloqueio de acesso público do Amazon S3 relata como público. Isso acontece porque o bloqueio de acesso público do Amazon S3 analisa as políticas de ações atuais e quaisquer ações potenciais que possam ser adicionadas no futuro, fazendo com que um bucket se torne público. Por outro lado, o Analisador de acesso para S3 analisa somente as ações atuais especificadas para o serviço do Amazon S3 na avaliação do status do acesso.

Para obter mais informações sobre o Access Analyzer for S3, consulte [Revisar o acesso de bucket usando o Access Analyzer for S3 \(p. 629\)](#).

Permissões

Para usar os recursos do Bloqueio de acesso público do Amazon S3, você deve ter as permissões a seguir.

Operação	Permissões obrigatórias
Status da política de bucket GET	s3:GetBucketPolicyStatus
Configurações do Bloqueio de acesso público do bucket GET	s3:GetBucketPublicAccessBlock
Configurações do Bloqueio de acesso público do bucket PUT	s3:PutBucketPublicAccessBlock
Configurações do Bloqueio de acesso público do bucket DELETE	s3:PutBucketPublicAccessBlock
Configurações do Bloqueio de acesso público da conta GET	s3:GetAccountPublicAccessBlock
Configurações do Bloqueio de acesso público da conta PUT	s3:PutAccountPublicAccessBlock
Configurações do Bloqueio de acesso público da conta DELETE	s3:PutAccountPublicAccessBlock
Configurações do Bloqueio de acesso público do ponto de acesso PUT	s3>CreateAccessPoint

Note

As operações DELETE exigem as mesmas permissões das operações PUT. Não há permissões separadas para as operações DELETE.

Configurar o bloqueio de acesso público

Para obter mais informações sobre como configurar o bloqueio de acesso público para sua Conta da AWS e seus buckets do Amazon S3, consulte os tópicos a seguir.

- [Configurar o bloqueio de acesso público para sua conta \(p. 625\)](#)
- [Configurar o bloqueio de acesso público para seus buckets do S3 \(p. 627\)](#)

Configurar o bloqueio de acesso público para sua conta

O Bloqueio de acesso público do Amazon S3 fornece configurações para pontos de acesso, buckets e contas para ajudar você a gerenciar o acesso público aos recursos do Amazon S3. Por padrão, novos buckets, pontos de acesso e objetos não permitem acesso público.

Para obter mais informações, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

Você pode usar o console do S3, a AWS CLI, os AWS SDKs e a API REST para definir as configurações de bloqueio de acesso público para todos os buckets da sua conta. Para obter mais informações, consulte as seções abaixo.

Para definir as configurações do bloqueio de acesso público para os buckets, consulte [Configurar o bloqueio de acesso público para seus buckets do S3 \(p. 627\)](#). Para obter mais informações sobre pontos de acesso, consulte [Executar operações de bloqueio de acesso público em um ponto de acesso \(p. 622\)](#).

Uso do console do S3

O bloqueio de acesso público do Amazon S3 evita a aplicação de todas as configurações que permitem acesso público a dados dentro de buckets do S3. Esta seção descreve como editar as configurações de bloqueio de acesso público para todos os buckets do S3 na sua Conta da AWS. Para obter mais informações sobre como bloquear o acesso público, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

Para editar configurações de bloqueio de acesso público para todos os buckets do S3 em uma Conta da AWS

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Escolha Block Public Access settings for this account (Configurações Bloquear acesso público para esta conta).
3. Escolha Edit (Editar) para alterar as configurações de acesso público de bloqueio para todos os buckets na Conta da AWS.
4. Escolha as configurações que deseja alterar e selecione Save (Salvar).
5. Quando a confirmação é solicitada, digite **confirm**. Em seguida, escolha Confirm (Confirmar) para salvar as alterações.

Uso do AWS CLI

Você pode usar o bloqueio de acesso público do Amazon S3 por meio da AWS CLI. Para obter mais informações sobre como configurar e usar a AWS CLI, consulte [O que é a AWS Command Line Interface?](#)

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)

Para obter informações e exemplos adicionais, consulte [put-public-access-block](#) na Referência da AWS CLI.

Uso de AWS SDKs

Java

Os exemplos a seguir mostram como usar o bloqueio de acesso público do Amazon S3 com o AWS SDK for Java para colocar uma configuração de bloqueio de acesso público em uma conta do Amazon S3. Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Usar a AWS SDK for Java \(p. 1266\)](#).

```
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.

Other SDKs

Para obter informações sobre como usar os outros AWS SDKs, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers \(p. 1258\)](#).

Uso dos REST API

Para obter informações sobre o uso do Bloqueio de acesso público do Amazon S3 por meio das APIs REST, consulte os tópicos a seguir na Referência da API do Amazon Simple Storage Service.

- Operações no nível da conta
 - [PUT PublicAccessBlock](#)
 - [GET PublicAccessBlock](#)
 - [DELETE PublicAccessBlock](#)

Configurar o bloqueio de acesso público para seus buckets do S3

O Bloqueio de acesso público do Amazon S3 fornece configurações para pontos de acesso, buckets e contas para ajudar você a gerenciar o acesso público aos recursos do Amazon S3. Por padrão, novos buckets, pontos de acesso e objetos não permitem acesso público.

Para obter mais informações, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

Você pode usar o console do S3, a AWS CLI, os AWS SDKs e a API REST para definir as configurações de bloqueio de acesso público para o bucket. Para obter mais informações, consulte as seções abaixo.

Para definir as configurações de bloqueio de acesso público para todos os buckets da sua conta, consulte [Configurar o bloqueio de acesso público para sua conta \(p. 625\)](#). Para obter informações sobre como configurar o bloqueio de acesso público para pontos de acesso, consulte [Executar operações de bloqueio de acesso público em um ponto de acesso \(p. 622\)](#).

Uso do console do S3

O bloqueio de acesso público do Amazon S3 evita a aplicação de todas as configurações que permitem acesso público a dados dentro de buckets do S3. Esta seção descreve como editar configurações de acesso público de bloqueio para um ou mais buckets do S3. Para obter informações sobre como bloquear o acesso público usando a AWS CLI, os AWS SDKs e as APIs REST do Amazon S3, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

Você pode ver se seu bucket está acessível publicamente na lista Buckets. Na coluna Access (Acessar), o Amazon S3 rotula as permissões para um bucket da seguinte forma:

- Público – Todos têm acesso a um ou mais dos seguintes: listagem de objetos, gravação de objetos, permissões de leitura e gravação.

- Objetos podem ser públicos – O bucket não é público, mas qualquer pessoa com as permissões apropriadas pode conceder acesso público a objetos.
- Buckets e objetos não públicos – O bucket e os objetos não têm acesso público.
- Somente usuários autorizados desta conta: O acesso é isolado a usuários e funções do IAM nesta conta e nas entidades principais do serviço da AWS porque há uma política que concede acesso público.

Também é possível filtrar pesquisas de bucket por tipo de acesso. Escolha um tipo de acesso na lista suspensa próxima da barra Search for buckets (Procurar buckets).

Se você vir um **Error** ao listar seus buckets e as respectivas configurações de acesso público, talvez você não tenha as permissões necessárias. Confira se você tem as seguintes permissões adicionadas à política de usuário ou perfil:

```
s3:GetAccountPublicAccessBlock
s3:GetBucketPublicAccessBlock
s3:GetBucketPolicyStatus
s3:GetBucketLocation
s3:GetBucketAcl
s3>ListAccessPoints
s3>ListAllMyBuckets
```

Em alguns casos raros, as solicitações também podem falhar devido a uma interrupção da Região da AWS.

Para editar as configurações de bloqueio de acesso público do Amazon S3 para um bucket do S3

Siga essas etapas caso você precise alterar as configurações de acesso público para um único bucket do S3.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Bucket name (Nome do bucket), escolha o nome do bucket que você deseja.
3. Escolha Permissions (Permissões).
4. Escolha Edit (Editar) para alterar as configurações de acesso público do bucket. Para obter mais informações sobre as quatro configurações do Bloqueio de acesso público do Amazon S3, consulte [Configurações do bloqueio de acesso público \(p. 620\)](#).
5. Escolha as configurações que você deseja alterar e Save (Salvar).
6. Quando a confirmação é solicitada, digite **confirm**. Em seguida, escolha Confirm (Confirmar) para salvar as alterações.

É possível alterar as configurações de bloqueio de acesso público do Amazon S3 ao criar um bucket. Para obter mais informações, consulte [Criação de um bucket \(p. 129\)](#).

Uso do AWS CLI

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

Para obter mais informações e exemplos, consulte [put-public-access-block](#) na Referência da AWS CLI.

Uso de AWS SDKs

Java

```
AmazonS3 client = AmazonS3ClientBuilder.standard()
    .withCredentials(<credentials>)
    .build();

client.setPublicAccessBlock(new SetPublicAccessBlockRequest()
    .withBucketName(<bucket-name>)
    .withPublicAccessBlockConfiguration(new PublicAccessBlockConfiguration()
        .withBlockPublicAcls(<value>)
        .withIgnorePublicAccls(<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.

Other SDKs

Para obter informações sobre como usar os outros AWS SDKs, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers \(p. 1258\)](#).

Uso dos REST API

Para obter informações sobre o uso do Bloqueio de acesso público do Amazon S3 por meio das APIs REST, consulte os tópicos a seguir na Referência da API do Amazon Simple Storage Service.

- Operações no nível do bucket
 - [PUT PublicAccessBlock](#)
 - [GET PublicAccessBlock](#)
 - [DELETE PublicAccessBlock](#)
 - [GET BucketPolicyStatus](#)

Revisar o acesso de bucket usando o Access Analyzer for S3

O Access Analyzer para S3 alerta você sobre buckets do S3 que são configurados para permitir o acesso a qualquer pessoa na Internet ou a outras Contas da AWS, incluindo Contas da AWS fora da organização. Para cada bucket público ou compartilhado, você recebe descobertas sobre a origem e o nível de acesso público ou compartilhado. Por exemplo, o analisador de acesso para S3 pode mostrar que um bucket tem acesso de leitura ou gravação fornecido por meio de uma lista de controle de acesso (ACL) de bucket, uma política de bucket, uma política de ponto de acesso multiregional ou uma política de ponto de acesso. Com esse conhecimento, você pode tomar medidas corretivas imediatas e precisas para restaurar o acesso ao bucket da forma desejada.

Ao revisar um bucket em risco no Analisador de acesso para S3, você pode bloquear todo o acesso público ao bucket com um único clique. Recomendamos que você bloquee todo o acesso aos buckets, a menos que exija acesso público para dar suporte a um caso de uso específico. Antes de bloquear todo o acesso público, certifique-se de que os aplicativos continuarão funcionando corretamente sem acesso

público. Para obter mais informações, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

Você também pode detalhar as configurações de permissão no nível do bucket para definir níveis granulares de acesso. Para casos de uso específicos e verificados que exigem acesso público, como hospedagem estática de sites, downloads públicos ou compartilhamento entre contas, você pode reconhecer e registrar sua intenção de que o bucket permaneça público ou compartilhado arquivando as descobertas do bucket. Você pode acessar novamente e modificar essas configurações de bucket a qualquer momento. Você também pode fazer download das descobertas como um relatório CSV para fins de auditoria.

O Analisador de acesso para S3 está disponível sem custo adicional no console do Amazon S3. O Access Analyzer for S3 usa tecnologia do AWS Identity and Access Management (IAM) Access Analyzer. Para usar o Analisador de acesso para S3 no console do Amazon S3, visite o console do IAM e habilite o IAM Analisador de acesso por região.

Para obter mais informações sobre o Analisador de acesso do IAM, consulte [O que é o Analisador de acesso?](#) no Guia do usuário do IAM. Para obter mais informações sobre o Analisador de acesso para S3, consulte as seções a seguir.

Important

- O Analisador de acesso para S3 requer um analisador de nível de conta. Para usar o Analisador de acesso para S3, acesse Analisador de acesso do IAM e crie um analisador que tenha uma conta como a zona de confiança. Para obter mais informações, consulte [Habilitar o Analisador de acesso](#) no Guia do usuário do IAM.
- Quando uma política de bucket ou a ACL do bucket é adicionada ou modificada, o Analisador de acesso gera e atualiza descobertas com base na alteração dentro de 30 minutos. As descobertas relacionadas às configurações do Bloqueio de acesso público na conta podem não ser geradas ou atualizadas por até 6 horas após a alteração das configurações. As descobertas relacionadas a pontos de acesso multirregionais podem não ser geradas ou atualizadas até seis horas após o ponto de acesso multirregional ser criado, excluído ou você alterar sua política.

Tópicos

- [Quais informações o Analisador de acesso para S3 fornece? \(p. 630\)](#)
- [Habilitar o Analisador de acesso para S3 \(p. 631\)](#)
- [Bloquear todo o acesso público \(p. 631\)](#)
- [Revisar e alterar o acesso ao bucket \(p. 632\)](#)
- [Arquivar descobertas de bucket \(p. 633\)](#)
- [Ativar uma descoberta de bucket arquivada \(p. 633\)](#)
- [Visualizar detalhes de descobertas \(p. 633\)](#)
- [Fazer download de um relatório do Analisador de acesso para S3 \(p. 634\)](#)

Quais informações o Analisador de acesso para S3 fornece?

O Access Analyzer para S3 fornece descobertas para buckets que podem ser acessados fora da sua Conta da AWS. Os buckets listados em Buckets with public access (Buckets com acesso público) podem ser acessados por qualquer pessoa na internet. Se o Analisador de acesso para S3 identificar buckets públicos, você também verá um aviso na parte superior da página que mostra o número de buckets públicos na região. Os buckets listados em Buckets with access from other Contas da AWS — including third-party Contas da AWS (Buckets com acesso de outras Contas da AWS, inclusive de terceiro) são compartilhados condicionalmente com outras Contas da AWS, inclusive com contas fora de sua organização.

Para cada bucket, o Analisador de acesso para S3 fornece as seguintes informações:

- Bucket name
- Descoberto pelo Analisador de acesso – Quando o Analisador de acesso para S3 descobriu o acesso a bucket público ou compartilhado.
- Compartilhado por: como o bucket é compartilhado, por meio de uma política de bucket, uma ACL de bucket ou uma política de ponto de acesso multirregional ou uma política de ponto de acesso. Pontos de acesso multirregionais são refletidos em pontos de acesso. Um bucket pode ser compartilhado por meio de políticas e ACLs. Se você quiser descobrir e analisar a origem do acesso ao bucket, poderá usar as informações nesta coluna como um ponto de partida para tomar medidas corretivas imediatas e precisas.
- Status – O status da descoberta do bucket. O Analisador de acesso para S3 exibe descobertas para todos os buckets públicos e compartilhados.
 - Ativa – a descoberta não foi analisada.
 - Arquivada – a descoberta foi analisada e confirmada conforme pretendido.
 - All (Todas): todas as descobertas de buckets que são públicos ou compartilhados com outras Contas da AWS, incluindo Contas da AWS fora da sua organização.
- Nível de acesso — permissões de acesso concedidas para o bucket:
 - Lista – lista de recursos.
 - Leitura – ler, mas não editar o conteúdo e os atributos do recurso.
 - Gravação – criar, excluir ou modificar recursos.
 - Permissões – conceder ou modificar permissões de recursos.
 - Atribuição de tags – atualizar as tags associadas ao recurso.

Habilitar o Analisador de acesso para S3

Para usar o Analisador de acesso para S3, você deve concluir as seguintes etapas de pré-requisito.

1. Conceda as permissões necessárias aos usuários.

Para obter mais informações, consulte [Permissões necessárias para usar o Analisador de acesso](#) no Guia do usuário do IAM.

2. Acesse o IAM para criar um analisador de nível de conta para cada região onde você deseja usar o Analisador de acesso.

O Analisador de acesso para S3 requer um analisador de nível de conta. Para usar o Analisador de acesso para S3, você deve criar um analisador que tenha uma conta como a zona de confiança. Para obter mais informações, consulte [Habilitar o Analisador de acesso](#) no Guia do usuário do IAM.

Bloquear todo o acesso público

Se você quiser bloquear todo o acesso a um bucket com um único clique, use o botão Block all public access (Bloquear todos os acessos públicos) no Analisador de acesso para S3. Quando você bloquear todo o acesso público a um bucket, nenhum acesso público será concedido. Recomendamos que você bloquee todo o acesso público aos buckets, a menos que exija acesso público para dar suporte a um caso de uso específico e verificado. Antes de bloquear todo o acesso público, certifique-se de que os aplicativos continuarão funcionando corretamente sem acesso público.

Se não quiser bloquear todo o acesso público ao bucket, você pode editar as configurações de bloqueio de acesso público no console do Amazon S3 para configurar níveis granulares de acesso aos buckets. Para obter mais informações, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

Em eventos raros, o Analisador de acesso para S3 pode relatar nenhuma descoberta para um bucket que uma análise de bloqueio de acesso público do Amazon S3 relata como público. Isso acontece porque o bloqueio de acesso público do Amazon S3 analisa as políticas de ações atuais e quaisquer ações

potenciais que possam ser adicionadas no futuro, fazendo com que um bucket se torne público. Por outro lado, o Analisador de acesso para S3 analisa somente as ações atuais especificadas para o serviço do Amazon S3 na avaliação do status do acesso.

Para bloquear todo o acesso público a um bucket usando o Analisador de acesso para S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, em Dashboards (Painéis), escolha Access Analyzer for S3.
3. No Analisador de acesso para S3, escolha um bucket.
4. Selecione Block all public access (Bloquear todo o acesso público).
5. Para confirmar sua intenção de bloquear todo o acesso público ao bucket, em Block all public access (bucket settings) (Bloquear todo o acesso público [configurações de bucket]), digite **confirm**.

O Amazon S3 bloqueia todo o acesso público ao bucket. O status da descoberta do bucket é atualizado para resolved (resolvido) e o bucket desaparece da listagem do Analisador de acesso para S3. Se você quiser revisar os buckets resolvidos, abra o IAM Analisador de acesso no console do IAM.

Revisar e alterar o acesso ao bucket

Se você não pretendia conceder acesso às contas públicas ou outras Contas da AWS, incluindo contas fora da organização, você pode modificar a ACL de bucket, a política de bucket ou a política de ponto de acesso multirregional para remover o acesso ao bucket. A coluna Shared through (Compartilhado por) mostra todas as origens de acesso ao bucket: política de bucket, ACL de bucket e/ou política de ponto de acesso. Pontos de acesso multirregionais são refletidos em pontos de acesso.

Para revisar e alterar uma política de bucket, uma ACL de bucket, um ponto de acesso multirregional ou uma política de ponto de acesso

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Access Analyzer for S3 (Analísisador de acesso para S3).
3. Para ver se o acesso público ou o acesso compartilhado é concedido por meio de uma política de bucket, uma ACL de bucket ou uma política de ponto de acesso multirregional, verifique a coluna Shared through (Compartilhado por).
4. Em Buckets, escolha o nome do bucket com a política de bucket, a ACL de bucket ou a política de ponto de acesso multirregional que deseja alterar ou revisar.
5. Se você quiser alterar ou exibir uma ACL de bucket:
 - a. Escolha Permissions (Permissões).
 - b. Escolha Access Control List.
 - c. Revise a ACL de bucket e faça as alterações conforme necessário.

Para obter mais informações, consulte [Configurar ACLs \(p. 596\)](#).
6. Se você quiser alterar ou revisar uma política de bucket:
 - a. Escolha Permissions (Permissões).
 - b. Escolha Bucket Policy.
 - c. Revise ou altere a política de bucket conforme necessário.

Para obter mais informações, consulte [Adicionar uma política de bucket usando o console do Amazon S3 \(p. 516\)](#).
7. Se você quiser alterar ou exibir uma política de ponto de acesso multirregional:
 - a. Escolha Multi-Region Access Point (Ponto de acesso multirregional).

- b. Escolha o nome do ponto de acesso multirregional.
- c. Revise ou altere a política de pontos de acesso multirregionais, conforme necessário.

Para obter mais informações, consulte [Permissões do ponto de acesso multirregional \(p. 348\)](#).

8. Se você quiser revisar ou alterar uma política de ponto de acesso:

- a. Escolha access points (pontos de acesso).
- b. Escolha o nome do ponto de acesso.
- c. Revise ou altere o acesso conforme necessário.

Para obter mais informações, consulte [Como usar pontos de acesso do Amazon S3 com o console do Amazon S3 \(p. 331\)](#).

Se você editar ou remover uma ACL de bucket, uma política de bucket ou uma política de ponto de acesso para remover o acesso público ou compartilhado, o status das descobertas do bucket será atualizado para resolved (resolvido). As descobertas de bucket resolvidas desaparecem da lista do Analisador de acesso para S3, mas você pode visualizá-las no Analisador de acesso do IAM.

Arquivar descobertas de bucket

Se um bucket conceder acesso a contas públicas ou outras Contas da AWS, inclusive contas fora da organização, para comportar um caso de uso específico (por exemplo, um site estático, downloads públicos ou compartilhamento entre contas), você poderá arquivar a descoberta do bucket. Ao arquivar descobertas de bucket, você reconhece e registra sua intenção de que o bucket permaneça público ou compartilhado. As descobertas de bucket arquivadas permanecem na listagem do Analisador de acesso para S3 para que você sempre saiba quais buckets são públicos ou compartilhados.

Como arquivar descobertas de bucket no Analisador de acesso para S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Access Analyzer for S3 (Analisador de acesso para S3).
3. No Analisador de acesso para S3, escolha um bucket ativo.
4. Para confirmar sua intenção de que esse bucket seja acessado pelas contas públicas ou outras Contas da AWS, inclusive contas fora da organização, escolha Archive (Arquivar).
5. Digite **confirm** e escolha Archive (Arquivar).

Ativar uma descoberta de bucket arquivada

Depois de arquivar descobertas, você sempre poderá revisitá-las e alterar o status novamente para ativo, indicando que o bucket requer outra análise.

Como ativar uma descoberta de bucket arquivado no Analisador de acesso para S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Access Analyzer for S3 (Analisador de acesso para S3).
3. Escolha as descobertas do bucket arquivadas.
4. Escolha Mark as active (Marcar como ativo).

Visualizar detalhes de descobertas

Se você precisar ver mais informações sobre um bucket, poderá abrir os detalhes de descobertas de bucket no Analisador de acesso do IAM no console do IAM.

Para exibir detalhes de busca no Analisador de acesso para S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Access Analyzer for S3 (Analisador de acesso para S3).
3. No Analisador de acesso para S3, escolha um bucket.
4. Escolha View details (Exibir detalhes).

Os detalhes de descoberta são abertos no Analisador de acesso do IAM no console do IAM.

Fazer download de um relatório do Analisador de acesso para S3

Você pode fazer download das descobertas de bucket como um relatório CSV que pode ser usado para fins de auditoria. O relatório inclui as mesmas informações que você vê no Analisador de acesso para S3 no console do Amazon S3.

Como fazer download de um relatório

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Access Analyzer for S3 (Analisador de acesso para S3).
3. No filtro Região, escolha a região.

Atualizações do Analisador de acesso para S3 mostram buckets para a região escolhida.
4. Escolha Download report (Fazer download do relatório).

Um relatório CSV é gerado e salvo no computador.

Verificar a propriedade do bucket com a condição de proprietário do bucket

A condição de proprietário do bucket do Amazon S3 garante que os buckets que você usa em suas operações do S3 pertençam às Contas da AWS que você espera.

A maioria das operações do S3 lê ou grava em buckets específicos do S3. Essas operações incluem fazer upload, cópia e download de objetos, recuperar ou modificar as configurações de bucket e recuperar ou modificar as configurações de objeto. Ao executar essas operações, você especifica o bucket que deseja usar, incluindo seu nome com a solicitação. Por exemplo, para recuperar um objeto do S3, você faz uma solicitação que especifica o nome de um bucket e a chave de objeto a ser recuperada desse bucket.

Como o Amazon S3 identifica buckets com base em seus nomes, uma aplicação que usa um nome de bucket incorreto em uma solicitação pode executar operações inadvertidamente em um bucket diferente do esperado. Para ajudar a evitar interações não intencionais de bucket em situações como essa, você pode usar a condição do proprietário do bucket. A condição de proprietário do bucket permite que você verifique se o bucket de destino pertence à Conta da AWS esperada, fornecendo uma camada adicional de garantia de que suas operações do S3 estão tendo os efeitos que você pretende.

Tópicos

- [Quando usar a condição do proprietário do bucket \(p. 635\)](#)
- [Verificar um proprietário do bucket \(p. 635\)](#)
- [Exemplos \(p. 636\)](#)
- [Restrições e limitações \(p. 638\)](#)

Quando usar a condição do proprietário do bucket

Recomendamos usar a condição de proprietário do bucket sempre que você executar uma operação compatível do S3 e conhecer o ID da conta do proprietário do bucket esperado. A condição do proprietário do bucket está disponível para todas as operações de objeto do S3 e para a maioria das operações de bucket do S3. Para obter uma lista de operações do S3 que não oferecem suporte à condição de proprietário do bucket, consulte [Restrições e limitações \(p. 638\)](#).

Para ver o benefício de usar a condição de proprietário do bucket, considere o seguinte cenário envolvendo a cliente da AWS, Bea:

1. Bea desenvolve uma aplicação que usa o Amazon S3. Durante o desenvolvimento, Bea usa sua Conta da AWS somente para testes para criar um bucket chamado `bea-data-test` e configura sua aplicação para fazer solicitações a `bea-data-test`.
2. Bea implanta sua aplicação, mas esquece de reconfigurá-la para usar um bucket em sua Conta da AWS de produção.
3. Na produção, a aplicação de Bea faz solicitações para `bea-data-test`, que são bem-sucedidas. Isso resulta na gravação de dados de produção no bucket na conta de teste da Bea.

Bea pode ajudar a evitar situações como esta usando a condição de proprietário do bucket. Com a condição de proprietário do bucket, Bea pode incluir o ID da Conta da AWS do proprietário do bucket esperado em suas solicitações. O Amazon S3, então, verifica o ID da conta do proprietário do bucket antes de processar cada solicitação. Se o proprietário real do bucket não corresponder ao proprietário do bucket esperado, a solicitação falhará.

Se Bea usar a condição de proprietário do bucket, o cenário descrito anteriormente não resultará na gravação inadvertida da aplicação de Bea em um bucket de teste. Em vez disso, as solicitações que a aplicação dela fizer na etapa 3 falharão e apresentarão uma mensagem de erro de `Access Denied`. Ao usar a condição de proprietário do bucket, Bea ajuda a eliminar o risco de interagir acidentalmente com buckets na Conta da AWS errada.

Verificar um proprietário do bucket

Para usar a condição de proprietário do bucket, você inclui um parâmetro com sua solicitação que especifica o proprietário do bucket esperado. A maioria das operações do S3 envolve apenas um único bucket e requer apenas esse parâmetro único para usar a condição de proprietário do bucket. Para operações `CopyObject`, esse primeiro parâmetro especifica o proprietário esperado do bucket de destino e você inclui um segundo parâmetro para especificar o proprietário esperado do bucket de origem.

Quando você faz uma solicitação que inclui um parâmetro de condição do proprietário do bucket, o S3 verifica o ID da conta do proprietário do bucket em relação ao parâmetro especificado antes de processar a solicitação. Se o parâmetro corresponder ao ID da conta do proprietário do bucket, o S3 processará a solicitação. Se o parâmetro não corresponder ao ID da conta do proprietário do bucket, a solicitação falhará e apresentará uma mensagem de erro de `Access Denied`.

Você pode usar a condição de proprietário do bucket com a AWS Command Line Interface (AWS CLI), AWS SDKs e APIs REST do Amazon S3. Ao usar a condição de proprietário do bucket com a AWS CLI e as APIs REST do Amazon S3, utilize os seguintes nomes de parâmetros.

Método de acesso	Parâmetro para operações sem cópia	Copiar parâmetro de origem da operação	Copiar parâmetro de destino da operação
AWS CLI	<code>--expected-bucket-owner</code>	<code>--expected-source-bucket-owner</code>	<code>--expected-bucket-owner</code>

Método de acesso	Parâmetro para operações sem cópia	Copiar parâmetro de origem da operação	Copiar parâmetro de destino da operação
APIs REST do Amazon S3	x-amz-expected-bucket-ownerCabeçalho	x-amz-source-expected-bucket-ownerCabeçalho	x-amz-expected-bucket-ownerCabeçalho

Os nomes de parâmetros necessários para usar a condição de proprietário do bucket com os AWS SDKs variam dependendo da linguagem. Para determinar os parâmetros necessários, consulte a documentação do SDK para o idioma desejado. Você pode encontrar a documentação do SDK em [Ferramentas para criar na AWS](#).

Exemplos

Os exemplos a seguir mostram como você pode implementar a condição de proprietário do bucket no Amazon S3 usando a AWS CLI ou o AWS SDK for Java 2.x.

Example

Exemplo: fazer upload de um objeto

O exemplo a seguir carrega um objeto para o bucket do S3 `DOC-EXAMPLE-BUCKET1` usando a condição de proprietário do bucket para garantir que `DOC-EXAMPLE-BUCKET1` seja de propriedade da Conta da AWS 111122223333.

AWS CLI

```
aws s3api put-object \
    --bucket DOC-EXAMPLE-BUCKET1 --key exampleobject --
body example_file.txt \
    --expected-bucket-owner 111122223333
```

AWS SDK for Java 2.x

```
public void putObjectExample() {
    S3Client s3Client = S3Client.create();
    PutObjectRequest request = PutObjectRequest.builder()
        .bucket("DOC-EXAMPLE-BUCKET1")
        .key("exampleobject")
        .expectedBucketOwner("111122223333")
        .build();
    Path path = Paths.get("example_file.txt");
    s3Client.putObject(request, path);
}
```

Example

Exemplo: copiar um objeto

O exemplo a seguir copia o objeto `object1` do bucket do S3 `DOC-EXAMPLE-BUCKET1` para o bucket do S3 `DOC-EXAMPLE-BUCKET2`. Ele usa a condição de proprietário do bucket para garantir que os buckets sejam de propriedade das contas esperadas de acordo com a tabela a seguir.

Bucket	Proprietário esperado
<code>DOC-EXAMPLE-BUCKET1</code>	111122223333

Bucket	Proprietário esperado
DOC-EXAMPLE-BUCKET2	444455556666

AWS CLI

```
aws s3api copy-object --copy-source DOC-EXAMPLE-BUCKET1/object1 \
                      --bucket DOC-EXAMPLE-BUCKET2 --key object1copy \
                      --expected-source-bucket-owner 111122223333 --expected-
                      bucket-owner 444455556666
```

AWS SDK for Java 2.x

```
public void copyObjectExample() {
    S3Client s3Client = S3Client.create();
    CopyObjectRequest request = CopyObjectRequest.builder()
        .copySource("DOC-EXAMPLE-BUCKET1/object1")
        .destinationBucket("DOC-EXAMPLE-BUCKET2")
        .destinationKey("object1copy")
        .expectedSourceBucketOwner("111122223333")
        .expectedBucketOwner("444455556666")
        .build();
    s3Client.copyObject(request);
}
```

Example

Exemplo: recuperar uma política de bucket

O exemplo a seguir recupera a política de acesso para o bucket do S3 **DOC-EXAMPLE-BUCKET1** usando a condição de proprietário do bucket para garantir que **DOC-EXAMPLE-BUCKET1** seja de propriedade da Conta da AWS **111122223333**.

AWS CLI

```
aws s3api get-bucket-policy --bucket DOC-EXAMPLE-BUCKET1 --expected-bucket-
                                owner 111122223333
```

AWS SDK for Java 2.x

```
public void getBucketPolicyExample() {
    S3Client s3Client = S3Client.create();
    GetBucketPolicyRequest request = GetBucketPolicyRequest.builder()
        .bucket("DOC-EXAMPLE-BUCKET1")
        .expectedBucketOwner("111122223333")
        .build();
    try {
        GetBucketPolicyResponse response = s3Client.getBucketPolicy(request);
    }
    catch (S3Exception e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
}
```

Restrições e limitações

A condição de proprietário do bucket do Amazon S3 tem as seguintes restrições e limitações:

- O valor do parâmetro de condição do proprietário do bucket deve ser um ID de Conta da AWS (string alfanumérica de 12 dígitos). Não há compatibilidade com os principais de serviço.
- A condição do proprietário do bucket não está disponível para [CreateBucket](#), [ListBuckets](#) ou qualquer uma das operações incluídas no [AWS S3 Control](#). O Amazon S3 ignora todos os parâmetros de condição do proprietário do bucket incluídos nas solicitações para essas operações.
- A condição do proprietário do bucket verifica apenas se a conta especificada no parâmetro de verificação é proprietária do bucket. A condição do proprietário do bucket não verifica a configuração do bucket. Também não garante que a configuração do bucket atenda a quaisquer condições específicas ou corresponda a qualquer estado passado.

Controlar a propriedade de objetos e desabilitar ACLs para seu bucket

O S3 Object Ownership é uma configuração no nível do bucket do Amazon S3 que você pode usar para desabilitar as [listas de controle de acesso \(ACLs\)](#) (p. 587) e assumir a propriedade de cada objeto em seu bucket, simplificando o gerenciamento de acesso para dados armazenados no Amazon S3. Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio de ACLs. Você pode usar o Object Ownership para alterar esse comportamento padrão. Em Object Ownership, as ACLs são desabilitadas e você, como proprietário do bucket, automaticamente tem a propriedade de todos os objetos de seu bucket. Como resultado, o controle de acesso para seus dados é baseado em políticas, como políticas do IAM, políticas de bucket do S3, políticas de endpoint da Virtual Private Cloud (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations.

A maioria dos casos de uso modernos no Amazon S3 não requer mais o uso de ACLs, e recomendamos desabilitar as ACLs, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com Object Ownership, é possível desabilitar ACLs e confiar em políticas para controle de acesso. Ao desabilitar ACLs, você pode facilmente manter um bucket com objetos carregados por diferentes Contas da AWS. Como proprietário do bucket, você possui todos os objetos do bucket e pode gerenciar o acesso a eles usando políticas.

Há três configurações em Object Ownership que podem ser usadas para controlar a propriedade de objetos carregados no bucket e para desabilitar ou habilitar ACLs:

ACLs desabilitadas

- Bucket owner enforced (Aplicada pelo proprietário do bucket) (recomendado): as ACLs são desabilitadas e o proprietário do bucket automaticamente tem a propriedade e o controle total sobre todos os objetos do bucket. As ACLs não afetam mais as permissões nos dados no bucket do S3. O bucket usa políticas para definir o controle de acesso.

ACLs habilitadas

- Bucket owner preferred (Preferencial do proprietário do bucket): o proprietário do bucket tem a propriedade e o controle total sobre novos objetos que outras contas gravam no bucket com a ACL `bucket-owner-full-control` pré-configurada.
- Object writer (Gravador de objetos) (padrão): a Conta da AWS que carrega um objeto é a proprietária do objeto, tem controle total sobre ele e pode conceder acesso a outros usuários por meio de ACLs.

Na maioria dos casos de uso modernos do S3, recomendamos desabilitar as ACLs escolhendo a configuração aplicada pelo proprietário do bucket e usar sua política de bucket para compartilhar dados com usuários fora da conta, conforme necessário. Esse método simplifica o gerenciamento e a auditoria de permissões. É possível desabilitar as ACLs em buckets recém-criados e já existentes. No caso de um bucket existente que já contenha objetos, depois que você desabilitar ACLs, as ACLs de objeto e bucket não farão mais parte de uma avaliação de acesso, e o acesso será concedido ou negado com base em políticas. Para buckets existentes, você pode habilitar novamente as ACLs a qualquer momento após desabilitá-las, e suas ACLs de objeto e bucket preexistentes serão restauradas.

Antes de desabilitar as ACLs, recomendamos revisar sua política de bucket para garantir que ela abrange todas as maneiras pelas quais você pretende conceder acesso ao bucket fora da conta. É necessário redefinir a ACL do bucket para o padrão (controle total para o proprietário do bucket). Depois que você desabilitar as ACLs, seu bucket aceitará somente solicitações PUT que não especificam ACL ou solicitações PUT com ACLs de controle total do proprietário do bucket, como a ACL `bucket-owner-full-control` predefinida ou formas equivalentes dessa ACL expressa em XML. Aplicações existentes compatíveis com ACLs de controle total do proprietário do bucket não são afetadas. Solicitações PUT que contêm outras ACLs (por exemplo, concessões personalizadas para determinadas Contas da AWS) falham e retornam um erro 400 com o código de erro `AccessControlListNotSupported`.

Em contrapartida, um bucket com a configuração preferencial do proprietário do bucket continua aceitando e respeitando as ACLs de bucket e objeto. Com essa configuração, novos objetos que são escritos com a ACL `bucket-owner-full-control` pré-configurada são automaticamente de propriedade do proprietário do bucket, e não do gravador de objeto. Todos os outros comportamentos da ACL permanecem em vigor. Para exigir que todas as operações PUT do Amazon S3 incluam a ACL `bucket-owner-full-control` pré-configurada, [adicone uma política de bucket \(p. 661\)](#) que permita somente o carregamento de objetos que usam essa ACL.

Configurações do Object Ownership

Esta tabela mostra como cada configuração do Object Ownership afeta ACLs, objetos, propriedade de objetos e carregamentos de objetos.

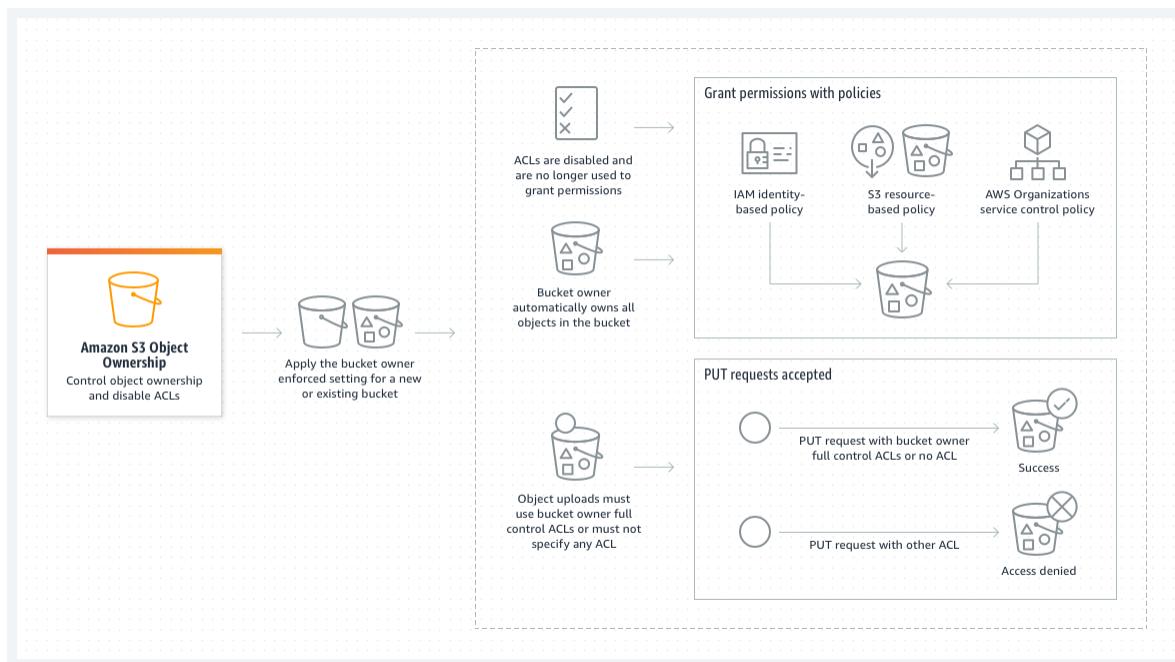
Configuração	Aplica-se a	Efeito na propriedade de objetos	Efeito nas ACLs	Carregamentos aceitos
Aplicada pelo proprietário do bucket (recomendado)	Objetos novos e existentes	O proprietário do bucket possui todos os objetos.	As ACLs estão desabilitadas e não afetam mais as permissões de acesso ao bucket. As solicitações para definir ou atualizar ACLs falham. Porém, há suporte para solicitações de leitura de ACLs. O proprietário do bucket tem total propriedade e controle. O gravador de objetos não tem mais total propriedade e controle.	Carregamentos com ACLs de controle total do proprietário do bucket ou carregamentos que não especificam uma ACL

Configuração	Aplica-se a	Efeito na propriedade de objetos	Efeito nas ACLs	Carregamentos aceitos
Preferencial do proprietário do bucket	Novos objetos	<p>Se um carregamento de objeto incluir a ACL <code>bucket-owner-full-control</code> pré-configurada, o proprietário do bucket terá o objeto.</p> <p>Os objetos carregados com outras ACLs são de propriedade da conta de gravação.</p>	<p>As ACLs podem ser atualizadas e podem conceder permissões.</p> <p>Se um carregamento de objeto incluir a ACL <code>bucket-owner-full-control</code> pré-configurada, o proprietário do bucket terá acesso de controle total, e o gravador de objetos não terá mais acesso de controle total.</p>	Todos os carregamentos
Gravador de objetos (padrão)	Novos objetos	O gravador de objetos é o proprietário do objeto.	<p>As ACLs podem ser atualizadas e podem conceder permissões.</p> <p>O gravador de objetos tem total acesso de controle.</p>	Todos os carregamentos

Alterações introduzidas ao desabilitar ACLs

Ao adotar a configuração aplicada pelo proprietário do bucket em Object Ownership para desabilitar ACLs, você automaticamente é o proprietário e assume controle total sobre todos os objetos do bucket sem realizar nenhuma outra ação. Após aplicar essa configuração, você verá três alterações:

- Todas as ACLs de bucket e as ACLs de objeto estão desabilitadas, dando a você acesso total, como o proprietário do bucket. Ao executar uma solicitação de ACL de leitura em seu bucket ou objeto, você verá que o acesso total é dado somente ao proprietário do bucket.
- Você, como proprietário do bucket, automaticamente tem a propriedade e o controle total sobre todos os objetos de seu bucket.
- As ACLs não afetam mais as permissões de acesso ao bucket. Como resultado, o controle de acesso a seus dados é baseado em políticas, como políticas do IAM, políticas de bucket do S3, políticas de endpoint da VPC e SCPs do Organizations.



Se você usar o versionamento do S3, o proprietário do bucket terá propriedade e controle total sobre todas as versões de objeto do bucket. Adotar a configuração aplicada pelo proprietário do bucket não adiciona uma nova versão do objeto.

Será possível carregar novos objetos no bucket somente se eles usarem ACLs de controle total do proprietário do bucket ou não especificarem uma ACL. Os carregamentos de objetos falharão se especificarem qualquer outra ACL. Para obter mais informações, consulte [Solução de problemas \(p. 662\)](#).

Como a operação PutObject do exemplo a seguir usando a AWS Command Line Interface (AWS CLI) contém a ACL `bucket-owner-full-control` pré-configurada, o objeto pode ser carregado em um bucket com ACLs desabilitadas.

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key key-name --body path-to-file --acl bucket-owner-full-control
```

Como a operação PutObject não especifica uma ACL, ela também funciona em um bucket com ACLs desabilitadas.

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key key-name --body path-to-file
```

Note

Se outras Contas da AWS precisarem de acesso a objetos após o carregamento, será necessário conceder outras permissões a essas contas por meio de políticas de bucket. Para obter mais informações, consulte [Demonstrações de exemplo: gerenciar o acesso aos recursos do Amazon S3 \(p. 553\)](#).

Habilitar ACLs novamente

É possível habilitar novamente as ACLs alterando da configuração aplicada pelo proprietário do bucket para outra configuração do Object Ownership a qualquer momento. Se você usou ACLs de objeto para gerenciamento de permissões antes de adotar a configuração aplicada pelo proprietário do bucket e não migrou essas permissões de ACL de objeto para sua política de bucket, essas permissões serão

restauradas quando as ACLs forem habilitadas novamente. Além disso, os objetos gravados no bucket enquanto a configuração aplicada pelo proprietário do bucket foi aplicada ainda pertencem ao proprietário do bucket.

Por exemplo, ao mudar da configuração aplicada pelo proprietário do bucket para o gravador de objetos, como proprietário do bucket, você não terá mais propriedade e controle total sobre objetos que anteriormente pertenciam a outras Contas da AWS. Em vez disso, as contas de carregamento voltam a ser proprietárias desses objetos. Objetos pertencentes a outras contas usam ACLs para permissões. Portanto, você não pode usar políticas para conceder permissões a esses objetos. No entanto, como proprietário do bucket, você ainda possui os objetos que foram gravados no bucket enquanto a configuração aplicada pelo proprietário do bucket foi aplicada. Esses objetos não pertencem ao gravador de objetos, mesmo que você habilite as ACLs novamente.

Pré-requisitos para desabilitar ACLs

Antes de desabilitar as ACLs de um bucket existente, cumpra os pré-requisitos a seguir.

Revise as ACLs de bucket e de objeto e migre permissões de ACLs

Quando você desabilita ACLs, as permissões concedidas por ACLs de bucket e de objeto não afetam mais o acesso. Antes de desabilitar as ACLs, revise suas ACLs de objeto e de bucket. Se as ACLs de bucket concederem permissões de leitura ou gravação a outras pessoas fora da conta, será necessário migrar essas permissões para a política de bucket antes de adotar a configuração aplicada pelo proprietário do bucket. Se você não migrar ACLs de bucket que concedem acesso de leitura ou gravação fora de sua conta, a solicitação para aplicar a configuração aplicada pelo proprietário do bucket falhará e retornará o código de erro [InvalidBucketAclWithObjectOwnership \(p. 662\)](#).

Por exemplo, se você quiser desabilitar as ACLs de um bucket que recebe logs de acesso ao servidor, deverá migrar as permissões de ACL do bucket para o grupo de entrega de log do S3 para a entidade principal do serviço de log em uma política de bucket. Para obter mais informações, consulte [Conceder acesso ao grupo de entrega de logs do S3 para log de acesso ao servidor \(p. 649\)](#).

Para que o gravador de objetos mantenha o controle total do objeto que ele carrega, o gravador de objetos é a melhor configuração do Object Ownership para seu caso de uso. Se você quiser controlar o acesso no nível de objeto individual, preferencial do proprietário do bucket é a melhor opção. Esses casos de uso são incomuns.

Para revisar ACLs e migrar permissões de ACL para políticas de bucket, consulte [Pré-requisitos para desabilitar ACLs \(p. 645\)](#).

Revisar e atualizar políticas de bucket que usam chaves de condição relacionadas à ACL

Depois de adotar a configuração aplicada pelo proprietário do bucket para desabilitar ACLs, será possível carregar novos objetos no bucket somente se a solicitação usar ACLs de controle total do proprietário do bucket ou não especificar uma ACL. Antes de desabilitar as ACLs, revise a política de bucket para chaves de condição relacionadas à ACL.

Se a política de bucket usar uma chave de condição relacionada à ACL para exigir a ACL `bucket-owner-full-control` pré-configurada (por exemplo, `s3:x-amz-acl`), não será necessário atualizar sua política de bucket. A política de bucket a seguir usa `s3:x-amz-acl` para exigir a ACL `bucket-owner-full-control` pré-configurada para solicitações `PutObject` do S3. Essa política ainda requer que o gravador de objetos especifique a ACL `bucket-owner-full-control` pré-configurada. No entanto, buckets com ACLs desabilitadas ainda aceitam essa ACL, então as solicitações continuam sendo bem-sucedidas sem necessidade de alterações no lado do cliente.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Only allow writes to my bucket with bucket owner full control",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": [  
                    "arn:aws:iam::111122223333:user/ExampleUser"  
                ]  
            },  
            "Action": [  
                "s3:PutObject"  
            ],  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",  
            "Condition": {  
                "StringEquals": {  
                    "s3:x-amz-acl": "bucket-owner-full-control"  
                }  
            }  
        }  
    ]  
}
```

Porém, se a política de bucket usar uma chave de condição relacionada à ACL que requer uma ACL diferente, será necessário remover essa chave de condição. Este exemplo de política de bucket requer a ACL public-read pré-configurada para solicitações PutObject do S3 e devem ser atualizadas antes de desabilitar as ACLs.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Only allow writes to my bucket with public read access",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": [  
                    "arn:aws:iam::111122223333:user/ExampleUser"  
                ]  
            },  
            "Action": [  
                "s3:PutObject"  
            ],  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",  
            "Condition": {  
                "StringEquals": {  
                    "s3:x-amz-acl": "public-read"  
                }  
            }  
        }  
    ]  
}
```

Permissões do Object Ownership

Para aplicar, atualizar ou excluir uma configuração do Object Ownership de um bucket, é necessária a permissão s3:PutBucketOwnershipControls. Para retornar a configuração do Object Ownership para um bucket, é necessário ter a permissão s3:GetBucketOwnershipControls. Para obter mais informações, consulte [Configurar Object Ownership ao criar um bucket \(p. 653\)](#) e [Visualizar a configuração do Object Ownership de um bucket do Amazon S3 \(p. 659\)](#).

Desabilitar ACLs para todos os novos buckets

Você pode exigir que todos os novos buckets sejam criados com ACLs desabilitadas usando políticas do IAM ou do Organizations. Você pode usar a chave de condição `s3:x-amz-object-ownership` em uma política do IAM ou do Organizations para exigir a configuração aplicada pelo proprietário do bucket para Object Ownership em todos os buckets recém-criados. Ao exigir a configuração aplicada pelo proprietário do bucket, você garante que as ACLs estejam desabilitadas para todos os novos buckets de sua conta ou organização. Para obter mais informações, consulte [Desabilitar ACLs para todos os novos buckets \(aplicada pelo proprietário do bucket\)](#) (p. 660).

Replicação e Object Ownership

Quando você usa a replicação do S3 e os buckets de origem e destino pertencem a diferentes Contas da AWS, é possível desabilitar ACLs (com a configuração aplicada pelo proprietário do bucket em Object Ownership) para alterar a propriedade da réplica para a Conta da AWS que possui o bucket de destino. Essa configuração imita o comportamento de substituição do proprietário existente sem a necessidade da permissão `s3:ObjectOwnerOverrideToBucketOwner`. Todos os objetos que são replicados para o bucket de destino com a configuração aplicada pelo proprietário do bucket pertencem ao proprietário do bucket de destino. Para obter mais informações sobre a opção de substituição do proprietário para configurações de replicação, consulte [Alterar o proprietário da réplica](#) (p. 864).

Ao usar a configuração padrão do gravador de objetos em Object Ownership ou aplicar a configuração preferencial do proprietário do bucket para o bucket de destino, você poderá usar a opção de substituição do proprietário da replicação do Amazon S3 para transferir a propriedade de objetos replicados ao proprietário do bucket de destino.

Configurar o Object Ownership

É possível aplicar uma configuração do Object Ownership usando o console do S3, a AWS CLI, AWS SDKs, a API REST do Amazon S3 ou o AWS CloudFormation. A seguinte API REST e comandos da AWS CLI são compatíveis com o Object Ownership:

API REST	AWS CLI	Descrição
PutBucketOwnershipControls	<code>put-bucket-ownership-controls</code>	Cria ou modifica a configuração do Object Ownership para um bucket do S3 existente.
CreateBucket	<code>create-bucket</code>	Cria um bucket usando o cabeçalho de solicitação <code>x-amz-object-ownership</code> para especificar a configuração do Object Ownership.
GetBucketOwnershipControls	<code>get-bucket-ownership-controls</code>	Recupera a configuração do Object Ownership de um bucket do Amazon S3.
DeleteBucketOwnershipControls	<code>delete-bucket-ownership-controls</code>	Exclui a configuração do Object Ownership de um bucket do Amazon S3.

Para obter mais informações sobre como aplicar e trabalhar com configurações de Object Ownership, consulte os tópicos a seguir.

Tópicos

- [Pré-requisitos para desabilitar ACLs \(p. 645\)](#)
- [Configurar Object Ownership ao criar um bucket \(p. 653\)](#)
- [Configurar Object Ownership em um bucket existente \(p. 656\)](#)
- [Visualizar a configuração do Object Ownership de um bucket do Amazon S3 \(p. 659\)](#)
- [Desabilitar ACLs de todos os novos buckets e aplicar Object Ownership \(p. 660\)](#)
- [Solução de problemas \(p. 662\)](#)

Pré-requisitos para desabilitar ACLs

Se a ACL de bucket conceder acesso fora de sua Conta da AWS, antes de desabilitar as ACLs, é necessário migrar suas permissões de ACL de bucket para a política de bucket e redefinir a ACL de bucket para a ACL privada padrão. Se você não migrar essas ACLs de bucket, a solicitação para adotar a configuração aplicada pelo proprietário do bucket para desabilitar ACLs falhará e retornará o código de erro [InvalidBucketAclWithObjectOwnership \(p. 662\)](#). Também recomendamos revisar suas permissões de ACL de objeto e migrá-las para a política de bucket. Para obter mais informações sobre outros pré-requisitos sugeridos, consulte [Pré-requisitos para desabilitar ACLs \(p. 642\)](#).

Cada uma das ACLs de objeto e de bucket existentes tem um equivalente em uma política do IAM. Os exemplos de política de bucket a seguir mostram como as permissões READ e WRITE para ACLs de objeto e de bucket são mapeadas para permissões do IAM. Para obter mais informações sobre como cada ACL é convertida em permissões do IAM, consulte [Mapeamento das permissões da ACL e das permissões da política de acesso \(p. 591\)](#).

Para revisar e migrar permissões de ACL para políticas de bucket, consulte os tópicos a seguir.

Tópicos

- [Exemplos de políticas de bucket \(p. 645\)](#)
- [Usar o console do S3 para revisar e migrar permissões de ACL \(p. 647\)](#)
- [Usar a AWS CLI para revisar e migrar permissões de ACL \(p. 647\)](#)
- [Demonstrações de exemplo \(p. 649\)](#)

Exemplos de políticas de bucket

Esses exemplos de políticas de bucket mostram como migrar permissões de ACL de bucket e de objeto READ e WRITE para uma Conta da AWS de terceiros para uma política de bucket. As ACLs READ_ACP e WRITE_ACP são menos relevantes para políticas porque concedem permissões relacionadas a ACLs (`s3:GetBucketAcl`, `s3:GetObjectAcl`, `s3:PutBucketAcl` e `s3:PutObjectAcl`).

Example – ACL READ para um bucket

Se o bucket tiver uma ACL READ que conceda à Conta da AWS 111122223333 permissão para listar o conteúdo do bucket, é possível gravar uma política de bucket que conceda as permissões `s3>ListBucket`, `s3>ListBucketVersions`, `s3>ListBucketMultipartUploads` para o bucket.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Permission to list the objects in a bucket",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": [  
                    "arn:aws:iam::111122223333:root"  
                ]  
            }  
        }  
    ]  
}
```

```
        },
        "Action": [
            "s3>ListBucket",
            "s3>ListBucketVersions",
            "s3>ListBucketMultipartUploads"
        ],
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
    }
}
```

Example – ACLs READ para cada objeto de um bucket

Se cada objeto em seu bucket tiver uma ACL READ que conceda acesso à Conta da AWS 111122223333, você pode gravar uma política de bucket que conceda permissões s3:GetObject e s3:GetObjectVersion a essa conta para cada objeto do bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Read permission for every object in a bucket",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root"
                ]
            },
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        }
    ]
}
```

Esse elemento de recurso de exemplo concede acesso a um objeto específico.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/OBJECT-KEY"
```

Example – ACL de gravação que concede permissões para gravar objetos em um bucket

Se o bucket tiver uma ACL de gravação que conceda à Conta da AWS 111122223333 permissão para gravar objetos no bucket, é possível gravar uma política de bucket que conceda a permissão s3:PutObject para o bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Permission to write objects to a bucket",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root"
                ]
            },
            "Action": [
                "s3:PutObject"
            ],

```

```
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    }
}
```

Usar o console do S3 para revisar e migrar permissões de ACL

Para revisar permissões de ACL

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Para exibir permissões de ACL de bucket, na lista Buckets, escolha o nome do bucket.
3. Escolha Permissions (Permissões).
4. Em Access control list (ACL) (Lista de controle de acesso [ACL]), revise suas permissões de ACL de bucket.
5. Escolha a guia Objetos.
6. Na lista Objects (Objetos), escolha o nome do objeto.
7. Escolha Permissions (Permissões).
8. Em Access control list (ACL) (Lista de controle de acesso [ACL]), revise suas permissões de ACL de objeto.

Para migrar permissões de ACL e atualizar a ACL de bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets (Buckets), escolha o nome do bucket.
3. Na guia Permissions (Permissões), escolha Bucket policy (Política de bucket), Edit (Editar).
4. Na caixa Policy (Política), adicione ou atualize a política de bucket.

Para obter exemplos de políticas de bucket, consulte [Exemplos de políticas de bucket \(p. 645\)](#) e [Demonstrações de exemplo \(p. 649\)](#).

5. Selecione Save changes.
6. [Atualize sua ACL de bucket \(p. 596\)](#) para remover concessões de ACL para outros grupos ou Contas da AWS.
7. [Aplique a configuração aplicada pelo proprietário do bucket \(p. 656\)](#) em Object Ownership.

Usar a AWS CLI para revisar e migrar permissões de ACL

1. Para devolver a ACL de bucket ao bucket, use o comando `get_bucket_acl` da AWS CLI:

```
aws s3api get-bucket-acl --bucket DOC-EXAMPLE-BUCKET
```

Por exemplo, essa ACL de bucket concede acesso WRITE e READ a uma conta de terceiros. Nessa ACL, a conta de terceiros é identificada pelo [ID de usuário canônico \(p. 594\)](#). Para adotar a configuração aplicada pelo proprietário do bucket e desabilitar ACLs, é necessário migrar essas permissões para a conta de terceiros para uma política de bucket.

```
{
  "Owner": {
    "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
    "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
```

```
        },
        "Grants": [
            {
                "Grantee": {
                    "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
                    "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
                    "Type": "CanonicalUser"
                },
                "Permission": "FULL_CONTROL"
            },
            {
                "Grantee": {
                    "DisplayName": "THIRD-PARTY-EXAMPLE-ACCOUNT",
                    "ID": "72806de9d1ae8b171cca9e2494a8d1335dfced4ThirdPartyAccountCanonicalUserID",
                    "Type": "CanonicalUser"
                },
                "Permission": "READ"
            },
            {
                "Grantee": {
                    "DisplayName": "THIRD-PARTY-EXAMPLE-ACCOUNT",
                    "ID": "72806de9d1ae8b171cca9e2494a8d1335dfced4ThirdPartyAccountCanonicalUserID",
                    "Type": "CanonicalUser"
                },
                "Permission": "WRITE"
            }
        ]
    }
```

Para ver outros exemplos de ACL, consulte [Demonstrações de exemplo \(p. 649\)](#).

2. Migrre suas permissões de ACL de bucket para uma política de bucket:

Este exemplo de política de bucket concede as permissões s3:PutObject e s3>ListBucket para uma conta de terceiros. Na política de bucket, a conta de terceiros é identificada pelo ID da Conta da AWS ([111122223333](#)).

```
aws s3api put-bucket-policy --bucket DOC-EXAMPLE-BUCKET --policy file://policy.json

policy.json:
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PolicyForCrossAccountAllowUpload",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root"
                ]
            },
            "Action": [
                "s3:PutObject",
                "s3>ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ]
        }
    ]
}
```

Para obter mais exemplos de políticas de bucket, consulte [Exemplos de políticas de bucket \(p. 645\)](#) e [Demonstrações de exemplo \(p. 649\)](#).

3. Para retornar a ACL de um objeto específico, use o comando [get-object-acl](#) da AWS CLI.

```
aws s3api get-object-acl --bucket DOC-EXAMPLE-BUCKET --key EXAMPLE-OBJECT-KEY
```

4. Se necessário, migre as permissões de ACL de objeto para a política de bucket.

Esse elemento de recurso de exemplo concede acesso a um objeto específico em uma política de bucket.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/EXAMPLE-OBJECT-KEY"
```

5. Redefina a ACL de bucket para a ACL padrão.

```
aws s3api put-bucket-acl --bucket DOC-EXAMPLE-BUCKET --acl private
```

6. [Aplique a configuração aplicada pelo proprietário do bucket \(p. 656\)](#) em Object Ownership.

Demonstrações de exemplo

Os exemplos a seguir mostram como migrar permissões de ACL para políticas de bucket para casos de uso específicos.

Tópicos

- [Conceder acesso ao grupo de entrega de logs do S3 para log de acesso ao servidor \(p. 649\)](#)
- [Conceder acesso público de leitura aos objetos do bucket \(p. 650\)](#)
- [Conceder acesso do Amazon ElastiCache for Redis ao bucket do Amazon S3 \(p. 652\)](#)

Conceder acesso ao grupo de entrega de logs do S3 para log de acesso ao servidor

Para adotar a configuração aplicada pelo proprietário do bucket para desabilitar ACLs para um bucket de destino de log de acesso ao servidor, será necessário migrar permissões de ACL de bucket para o grupo de entrega de log do S3 para a entidade principal do serviço de log (`logging.s3.amazonaws.com`) em uma política de bucket. Para obter mais informações sobre permissões de entrega de logs, consulte [Permissões para entrega de logs \(p. 1044\)](#).

Essa ACL de bucket concede acesso `WRITE` e `READ_ACP` ao grupo de entrega de logs do S3:

```
{  
    "Owner": {  
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",  
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"  
    },  
    "Grants": [  
        {  
            "Grantee": {  
                "Type": "CanonicalUser",  
                "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",  
                "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"  
            },  
            "Permission": "FULL_CONTROL"  
        },  
        {  
            "Grantee": {  
                "Type": "Group",  
                "DisplayName": "AWS CloudWatch Logs",  
                "ID": "S3LogDeliveryGroup" // This is the canonical user ID for the CloudWatch Logs group.  
            },  
            "Permission": "READ_ACP"  
        }  
    ]  
}
```

```
{  
    "Grantee": {  
        "Type": "Group",  
        "URI": "http://acs.amazonaws.com/groups/s3/LogDelivery"  
    },  
    "Permission": "WRITE"  
},  
{  
    "Grantee": {  
        "Type": "Group",  
        "URI": "http://acs.amazonaws.com/groups/s3/LogDelivery"  
    },  
    "Permission": "READ_ACP"  
}  
]  
}
```

Para migrar permissões de ACL de bucket para o grupo de entrega de log do S3 para a entidade principal do serviço de log em uma política de bucket

1. Adicione a política de bucket a seguir ao bucket de destino, substituindo os valores de exemplo.

```
aws s3api put-bucket-policy --bucket DOC-EXAMPLE-BUCKET --policy file://policy.json  
  
policy.json:  
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "S3ServerAccessLogsPolicy",  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "logging.s3.amazonaws.com"  
            },  
            "Action": [  
                "s3:PutObject"  
            ],  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/EXAMPLE-LOGGING-PREFIX*",  
            "Condition": {  
                "ArnLike": {  
                    "aws:SourceArn": "arn:aws:s3:::SOURCE-BUCKET-NAME"  
                },  
                "StringEquals": {  
                    "aws:SourceAccount": "SOURCE-AWS-ACCOUNT-ID"  
                }  
            }  
        }  
    ]  
}
```

2. Redefina a ACL de bucket de destino para a ACL padrão.

```
aws s3api put-bucket-acl --bucket DOC-EXAMPLE-BUCKET --acl private
```

3. [Aplique a configuração aplicada pelo proprietário do bucket \(p. 656\)](#) em Object Ownership para o bucket de destino.

Conceder acesso público de leitura aos objetos do bucket

Se as ACLs de objeto concederem acesso público de leitura a todos os objetos no bucket, será possível migrar essas permissões de ACL para uma política de bucket.

Essa ACL de objeto concede acesso público de leitura a um objeto de um bucket:

```
{  
    "Owner": {  
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",  
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"  
    },  
    "Grants": [  
        {  
            "Grantee": {  
                "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",  
                "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",  
                "Type": "CanonicalUser"  
            },  
            "Permission": "FULL_CONTROL"  
        },  
        {  
            "Grantee": {  
                "Type": "Group",  
                "URI": "http://acs.amazonaws.com/groups/global/AllUsers"  
            },  
            "Permission": "READ"  
        }  
    ]  
}
```

Para migrar permissões públicas de ACL de leitura para uma política de bucket

1. Para conceder acesso público de leitura a todos os objetos do bucket, adicione a política de bucket a seguir, substituindo os valores de exemplo.

```
aws s3api put-bucket-policy --bucket DOC-EXAMPLE-BUCKET --policy file://policy.json  
  
policy.json:  
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "PublicReadGetObject",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"  
            ]  
        }  
    ]  
}
```

Para conceder acesso público a um objeto específico de uma política de bucket, use o seguinte formato para o elemento Resource.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/OBJECT-KEY"
```

Para conceder acesso público a todos os objetos com um prefixo específico, use o seguinte formato para o elemento Resource.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/PREFIX/*"
```

- Aplique a configuração aplicada pelo proprietário do bucket (p. 656) em Object Ownership.

Conceder acesso do Amazon ElastiCache for Redis ao bucket do Amazon S3

Você pode [exportar o backup do ElastiCache for Redis](#) para um bucket do S3, o que lhe dá acesso ao backup de fora do ElastiCache. Para exportar o backup para um bucket do S3, é necessário conceder permissões do ElastiCache para copiar um snapshot para o bucket. Se você concedeu permissões ao ElastiCache em uma ACL de bucket, deverá migrar essas permissões para uma política de bucket antes de adotar a configuração aplicada pelo proprietário do bucket para desabilitar ACLs. Para obter mais informações, consulte [Conceder acesso do ElastiCache ao bucket do Amazon S3](#) no Guia do usuário do Amazon ElastiCache.

O exemplo a seguir mostra as permissões de ACL de bucket que concedem permissões ao ElastiCache.

```
{  
    "Owner": {  
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",  
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"  
    },  
    "Grants": [  
        {  
            "Grantee": {  
                "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",  
                "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",  
                "Type": "CanonicalUser"  
            },  
            "Permission": "FULL_CONTROL"  
        },  
        {  
            "Grantee": {  
                "DisplayName": "aws-scs-s3-readonly",  
                "ID": "540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353",  
                "Type": "CanonicalUser"  
            },  
            "Permission": "READ"  
        },  
        {  
            "Grantee": {  
                "DisplayName": "aws-scs-s3-readonly",  
                "ID": "540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353",  
                "Type": "CanonicalUser"  
            },  
            "Permission": "WRITE"  
        },  
        {  
            "Grantee": {  
                "DisplayName": "aws-scs-s3-readonly",  
                "ID": "540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353",  
                "Type": "CanonicalUser"  
            },  
            "Permission": "READ_ACP"  
        }  
    ]  
}
```

Para migrar permissões de ACL de bucket ao ElastiCache for Redis para uma política de bucket

- Adicione a política de bucket a seguir ao bucket, substituindo os valores de exemplo.

```
aws s3api put-bucket-policy --bucket DOC-EXAMPLE-BUCKET --policy file://policy.json
```

```
policy.json:  
  "Id": "Policy15397346",  
  "Statement": [  
    {  
      "Sid": "Stmt15399483",  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "Region.elasticache-snapshot.amazonaws.com"  
      },  
      "Action": [  
        "s3:PutObject",  
        "s3:GetObject",  
        "s3>ListBucket",  
        "s3:GetBucketAcl",  
        "s3>ListMultipartUploadParts",  
        "s3>ListBucketMultipartUploads"  
      ],  
      "Resource": [  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"  
      ]  
    }  
  ]  
}
```

2. Redefina a ACL de bucket para a ACL padrão:

```
aws s3api put-bucket-acl --bucket DOC-EXAMPLE-BUCKET --acl private
```

3. [Aplique a configuração aplicada pelo proprietário do bucket \(p. 656\)](#) em Object Ownership.

Configurar Object Ownership ao criar um bucket

Ao criar um bucket, você pode configurar o S3 Object Ownership. Para definir o Object Ownership para um bucket existente, consulte [Configurar Object Ownership em um bucket existente \(p. 656\)](#).

O S3 Object Ownership é uma configuração no nível do bucket do Amazon S3 que você pode usar para desabilitar as [listas de controle de acesso \(ACLs\) \(p. 587\)](#) e assumir a propriedade de cada objeto em seu bucket, simplificando o gerenciamento de acesso para dados armazenados no Amazon S3. Recomendamos desabilitar as ACLs, a menos que você precise controlar o acesso no nível de objeto individual.

Há três configurações em Object Ownership que podem ser usadas para controlar a propriedade de objetos carregados no bucket e para desabilitar ou habilitar ACLs:

ACLs desabilitadas

- Bucket owner enforced (Aplicada pelo proprietário do bucket) (recomendado): as ACLs são desabilitadas e o proprietário do bucket automaticamente tem a propriedade e o controle total sobre todos os objetos do bucket. As ACLs não afetam mais as permissões nos dados no bucket do S3. O bucket usa políticas para definir o controle de acesso.

ACLs habilitadas

- Bucket owner preferred (Preferencial do proprietário do bucket): o proprietário do bucket tem a propriedade e o controle total sobre novos objetos que outras contas gravam no bucket com a ACL `bucket-owner-full-control` pré-configurada.
- Object writer (Gravador de objetos) (padrão): a Conta da AWS que carrega um objeto é a proprietária do objeto, tem controle total sobre ele e pode conceder acesso a outros usuários por meio de ACLs.

Permissões: para criar um bucket e selecionar uma configuração para Object Ownership, é necessário ter as permissões `s3:CreateBucket` e `s3:PutBucketOwnershipControls`. Para obter mais informações sobre as permissões do Amazon S3, consulte [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#).

Uso do console do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Selecione Create bucket (Criar bucket).

O assistente Create bucket (Criar bucket) é aberto.

3. Em Bucket name (Nome do bucket), insira um nome compatível com o DNS para seu bucket.

O nome do bucket deve:

- Seja exclusivo em todo o Amazon S3.
- Ter entre 3 e 63 caracteres.
- Não contém caracteres maiúsculos.
- Começar com uma letra minúscula ou um número.

Depois de criado o bucket, não é possível mudar o nome. Para obter informações sobre nomeação de buckets, consulte [Regras de nomeação de bucket \(p. 127\)](#).

Important

Evite incluir informações sigilosas, como números de conta, no nome do bucket. O nome do bucket é visível nos URLs que apontam para os objetos no bucket.

4. Em Region (Região), escolha a Região da AWS onde deseja que o bucket resida.

Escolha uma região próxima de você para minimizar a latência e os custos e atender aos requisitos regulatórios. Os objetos armazenados em uma região nunca saem dessa região, a menos que você os transfira para outra região. Para obter uma lista das Regiões da AWS do Amazon S3, consulte [Endpoints de serviço da AWS](#) na Referência geral da Amazon Web Services.

5. Em Object Ownership, para desabilitar ou habilitar ACLs e controlar a propriedade de objetos carregados em seu bucket, escolha uma das seguintes configurações:

ACLs desabilitadas

- Bucket owner enforced (Aplicada pelo proprietário do bucket) (recomendado): as ACLs são desabilitadas e o proprietário do bucket automaticamente tem a propriedade e o controle total sobre todos os objetos do bucket. As ACLs não afetam mais as permissões nos dados no bucket do S3. O bucket usa políticas para definir o controle de acesso.

Para exigir que todos os novos buckets sejam criados com ACLs desabilitadas usando IAM ou políticas do AWS Organizations, consulte [Desabilitar ACLs para todos os novos buckets \(aplicada pelo proprietário do bucket\) \(p. 660\)](#).

ACLs habilitadas

- Bucket owner preferred (Preferencial do proprietário do bucket): o proprietário do bucket tem a propriedade e o controle total sobre novos objetos que outras contas gravam no bucket com a ACL `bucket-owner-full-control` pré-configurada.

Se você aplicar a configuração preferencial do proprietário do bucket para exigir que todos os carregamentos do Amazon S3 incluam a ACL `bucket-owner-full-control` pré-configurada,

será possível [adicionar uma política de bucket \(p. 661\)](#) que só permita carregamentos de objetos que usam essa ACL.

- Object writer (Gravador de objetos): a Conta da AWS que carrega um objeto é a proprietária do objeto, tem controle total sobre ele e pode conceder acesso a outros usuários por meio de ACLs.

Note

Para aplicar a configuração Bucket owner enforced (Aplicada pelo proprietário do bucket) ou Bucket owner preferred (Preferencial do proprietário do bucket), é necessário ter as seguintes permissões: `s3:CreateBucket` e `s3:PutBucketOwnershipControls`.

6. Em Bucket settings for Block Public Access (Configurações de bucket para o Bloqueio de acesso público), escolha as configurações de bloqueio de acesso público que deseja aplicar ao bucket.

Recomendamos que você mantenha todas as configurações ativadas, a menos que saiba que precisa desativar uma ou mais delas para seu caso de uso, como para hospedar um site público. As configurações de bloqueio de acesso público que você habilitar para o bucket também serão ativadas para todos os pontos de acesso criados no bucket. Para obter mais informações sobre como bloquear o acesso público, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

7. (Opcional) Se você quiser habilitar o Bloqueio de Objetos do S3, faça o seguinte:

- a. Selecione Advanced settings (Configurações avançadas).

Important

Você só pode habilitar o bloqueio de objetos do S3 para um bucket ao criá-lo. Se você habilitar o bloqueio de objeto para o bucket, não poderá desabilitá-lo mais tarde. A ativação do bloqueio de objetos também permite o versionamento para o bucket. Após habilitar o bloqueio de objeto para o bucket, é necessário definir as configurações padrão de retenção e de retenção legal do bloqueio de objeto para proteger novos objetos de serem excluídos ou substituídos. Para obter mais informações, consulte [Configurar o bloqueio de objetos do S3 usando o console \(p. 727\)](#).

- b. Se quiser ativar o bloqueio de objetos, escolha **Enable**, leia o aviso que aparece e confirme-o.

Para obter mais informações sobre o recurso bloqueio de objeto do S3, consulte [Usar o bloqueio de objetos do S3 \(p. 722\)](#).

Note

Para criar um bucket habilitado para bloqueio de objetos, você deve ter as seguintes permissões: `S3:CreateBucket`, `S3:PutBucketVersioning` e `S3:PutBucketObjectLockConfiguration`.

8. Selecione Create bucket (Criar bucket).

Uso do AWS CLI

Para definir Object Ownership ao criar um novo bucket, use o comando `create-bucket` da AWS CLI com o parâmetro `--object-ownership`.

Este exemplo aplica a configuração aplicada pelo proprietário do bucket para um novo bucket usando a AWS CLI:

```
aws s3api create-bucket --bucket DOC-EXAMPLE-BUCKET --region us-east-1 --object-ownership BucketOwnerEnforced
```

Usar o AWS SDK para Java

Este exemplo define a configuração aplicada pelo proprietário do bucket para um novo bucket usando a AWS SDK for Java:

```
// Build the ObjectOwnership for CreateBucket
CreateBucketRequest createBucketRequest = CreateBucketRequest.builder()
    .bucket(bucketName)
    .objectOwnership(ObjectOwnership.BucketOwnerEnforced)
    .build()

// Send the request to S3
s3client.createBucket(createBucketRequest);
```

Usar o AWS CloudFormation

Para usar o recurso `AWS::S3::Bucket` do AWS CloudFormation para definir Object Ownership ao criar um novo bucket, consulte [OwnershipControls em AWS# S3# Bucket](#) no Guia do usuário do AWS CloudFormation.

Uso dos REST API

Para adotar a configuração aplicada pelo proprietário do bucket para S3 Object Ownership, use a operação de API `CreateBucket` com o cabeçalho da solicitação `x-amz-object-ownership` definido como `BucketOwnerEnforced`. Para obter informações e exemplo, consulte [CreateBucket](#) na Referência da API do Amazon Simple Storage Service.

Próximas etapas: depois de adotar as configurações preferenciais aplicadas pelo proprietário do bucket ou do proprietário do bucket para Object Ownership, você poderá seguir estas etapas:

- [Bucket owner enforced \(p. 660\)](#) (Aplicada pelo proprietário do bucket): exija que todos os novos buckets sejam criados com ACLs desabilitadas usando uma política do IAM ou do Organizations.
- [Bucket owner preferred \(p. 661\)](#) (Preferencial do proprietário do bucket): adicione uma política de bucket do S3 para exigir a `ACL bucket-owner-full-control` pré-configurada para todos os carregamentos de objetos para o bucket.

Configurar Object Ownership em um bucket existente

Você pode configurar S3 Object Ownership em um bucket do S3 existente. Para aplicar o Object Ownership ao criar um bucket, consulte [Configurar Object Ownership ao criar um bucket \(p. 653\)](#).

O S3 Object Ownership é uma configuração no nível do bucket do Amazon S3 que você pode usar para desabilitar as [listas de controle de acesso \(ACLs\) \(p. 587\)](#) e assumir a propriedade de cada objeto em seu bucket, simplificando o gerenciamento de acesso para dados armazenados no Amazon S3. Recomendamos desabilitar as ACLs, a menos que você precise controlar o acesso no nível de objeto individual.

Há três configurações em Object Ownership que podem ser usadas para controlar a propriedade de objetos carregados no bucket e para desabilitar ou habilitar ACLs:

ACLs desabilitadas

- Bucket owner enforced (Aplicada pelo proprietário do bucket) (recomendado): as ACLs são desabilitadas e o proprietário do bucket automaticamente tem a propriedade e o controle total sobre todos os objetos

do bucket. As ACLs não afetam mais as permissões nos dados no bucket do S3. O bucket usa políticas para definir o controle de acesso.

ACLs habilitadas

- Bucket owner preferred (Preferencial do proprietário do bucket): o proprietário do bucket tem a propriedade e o controle total sobre novos objetos que outras contas gravam no bucket com a ACL `bucket-owner-full-control` pré-configurada.
- Object writer (Gravador de objetos) (padrão): a Conta da AWS que carrega um objeto é a proprietária do objeto, tem controle total sobre ele e pode conceder acesso a outros usuários por meio de ACLs.

Pré-requisitos: antes de adotar a configuração aplicada pelo proprietário do bucket para desabilitar ACLs, você deve migrar permissões de ACL de bucket para políticas de bucket e redefinir suas ACLs de bucket para a ACL privada padrão. Também recomendamos migrar permissões de ACL de objeto para políticas de bucket e editar políticas de bucket que exigem ACLs que não sejam ACLs de controle total do proprietário do bucket. Para obter mais informações, consulte [Pré-requisitos para desabilitar ACLs \(p. 645\)](#).

Permissions (Permissões): para usar essa operação, é necessário ter a permissão `s3:PutBucketOwnershipControls`. Para obter mais informações sobre as permissões do Amazon S3, consulte [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#).

Uso do console do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket para o qual você deseja aplicar uma configuração S3 Object Ownership.
3. Escolha a guia Permissions (Permissões).
4. Em Object Ownership, escolha Edit (Editar).
5. Em Object Ownership, para desabilitar ou habilitar ACLs e controlar a propriedade de objetos carregados em seu bucket, escolha uma das seguintes configurações:

ACLs desabilitadas

- Bucket owner enforced (Aplicada pelo proprietário do bucket) (recomendado): as ACLs são desabilitadas e o proprietário do bucket automaticamente tem a propriedade e o controle total sobre todos os objetos do bucket. As ACLs não afetam mais as permissões nos dados no bucket do S3. O bucket usa políticas para definir o controle de acesso.

Para exigir que todos os novos buckets sejam criados com ACLs desabilitadas usando IAM ou políticas do AWS Organizations, consulte [Desabilitar ACLs para todos os novos buckets \(aplicada pelo proprietário do bucket\) \(p. 660\)](#).

ACLs habilitadas

- Bucket owner preferred (Preferencial do proprietário do bucket): o proprietário do bucket tem a propriedade e o controle total sobre novos objetos que outras contas gravam no bucket com a ACL `bucket-owner-full-control` pré-configurada.

Se você aplicar a configuração preferencial do proprietário do bucket para exigir que todos os carregamentos do Amazon S3 incluam a ACL `bucket-owner-full-control` pré-configurada, será possível [adicionar uma política de bucket \(p. 661\)](#) que só permita carregamentos de objetos que usam essa ACL.

- Object writer (Gravador de objetos): a Conta da AWS que carrega um objeto é a proprietária do objeto, tem controle total sobre ele e pode conceder acesso a outros usuários por meio de ACLs.
6. Escolha Save (Salvar).

Uso do AWS CLI

Para aplicar uma configuração do Object Ownership a um bucket existente, use o comando `put-bucket-ownership-controls` com o parâmetro `--ownership-controls`.

Este exemplo aplica a configuração aplicada pelo proprietário do bucket para um bucket existente usando a AWS CLI:

```
aws s3api put-bucket-ownership-controls --bucket DOC-EXAMPLE-BUCKET --ownership-controls  
Rules=[{ObjectOwnership=BucketOwnerEnforced}]
```

Usar o AWS SDK para Java

Este exemplo aplica a configuração `BucketOwnerEnforced` aplicada pelo proprietário do bucket para um bucket existente usando AWS SDK for Java:

```
// Build the ObjectOwnership for BucketOwnerEnforced  
OwnershipControlsRule rule = OwnershipControlsRule.builder()  
    .objectOwnership(ObjectOwnership.BucketOwnerEnforced)  
    .build();  
  
OwnershipControls ownershipControls = OwnershipControls.builder()  
    .rules(rule)  
    .build()  
  
// Build the PutBucketOwnershipControlsRequest  
PutBucketOwnershipControlsRequest putBucketOwnershipControlsRequest =  
    PutBucketOwnershipControlsRequest.builder()  
        .bucket(BUCKET_NAME)  
        .ownershipControls(ownershipControls)  
        .build();  
  
// Send the request to S3  
s3client.putBucketOwnershipControls(putBucketOwnershipControlsRequest);
```

Usar o AWS CloudFormation

Para usar o AWS CloudFormation para aplicar uma configuração do Object Ownership a um bucket existente, consulte [AWS::S3::Bucket OwnershipControls](#) no Guia do usuário do AWS CloudFormation.

Uso dos REST API

Para usar a API REST para aplicar uma configuração do Object Ownership a um bucket do S3 existente, use `PutBucketOwnershipControls`. Para obter mais informações, consulte [PutBucketOwnershipControls](#) na Referência da API do Amazon Simple Storage Service.

Próximas etapas: depois de adotar as configurações preferenciais aplicadas pelo proprietário do bucket ou do proprietário do bucket para Object Ownership, você poderá seguir estas etapas:

- [Bucket owner enforced \(p. 660\)](#) (Aplicada pelo proprietário do bucket): exija que todos os novos buckets sejam criados com ACLs desabilitadas usando uma política do IAM ou do Organizations.
- [Bucket owner preferred \(p. 661\)](#) (Preferencial do proprietário do bucket): adicione uma política de bucket do S3 para exigir a ACL `bucket-owner-full-control` pré-configurada para todos os carregamentos de objetos para o bucket.

Visualizar a configuração do Object Ownership de um bucket do Amazon S3

O S3 Object Ownership é uma configuração no nível do bucket do Amazon S3 que você pode usar para desabilitar as [listas de controle de acesso \(ACLs\) \(p. 587\)](#) e assumir a propriedade de cada objeto em seu bucket, simplificando o gerenciamento de acesso para dados armazenados no Amazon S3. Recomendamos desabilitar as ACLs, a menos que você precise controlar o acesso no nível de objeto individual.

Há três configurações em Object Ownership que podem ser usadas para controlar a propriedade de objetos carregados no bucket e para desabilitar ou habilitar ACLs:

ACLs desabilitadas

- Bucket owner enforced (Aplicada pelo proprietário do bucket) (recomendado): as ACLs são desabilitadas e o proprietário do bucket automaticamente tem a propriedade e o controle total sobre todos os objetos do bucket. As ACLs não afetam mais as permissões nos dados no bucket do S3. O bucket usa políticas para definir o controle de acesso.

ACLs habilitadas

- Bucket owner preferred (Preferencial do proprietário do bucket): o proprietário do bucket tem a propriedade e o controle total sobre novos objetos que outras contas gravam no bucket com a ACL `bucket-owner-full-control` pré-configurada.
- Object writer (Gravador de objetos) (padrão): a Conta da AWS que carrega um objeto é a proprietária do objeto, tem controle total sobre ele e pode conceder acesso a outros usuários por meio de ACLs.

É possível ver as configurações do S3 Object Ownership de um bucket do Amazon S3. Para definir o Object Ownership para um novo bucket, consulte [Configurar Object Ownership ao criar um bucket \(p. 653\)](#). Para definir o Object Ownership para um bucket existente, consulte [Configurar Object Ownership em um bucket existente \(p. 656\)](#).

Permissões (Permissões): para usar essa operação, é necessário ter a permissão `s3:GetBucketOwnershipControls`. Para obter mais informações sobre as permissões do Amazon S3, consulte [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#).

Uso do console do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket para o qual você deseja aplicar uma configuração do Object Ownership.
3. Escolha a guia Permissions (Permissões).
4. Em Object Ownership, você pode ver as configurações de Object Ownership do bucket.

Uso do AWS CLI

Para recuperar a configuração S3 Object Ownership para um bucket do S3, use o comando `get-bucket-ownership-controls` da AWS CLI.

```
aws s3api get-bucket-ownership-controls --bucket DOC-EXAMPLE-BUCKET
```

Uso dos REST API

Para recuperar a configuração de Object Ownership para um bucket do S3, use a operação de API [GetBucketOwnershipControls](#). Para obter mais informações, consulte [GetBucketOwnershipControls](#).

Desabilitar ACLs de todos os novos buckets e aplicar Object Ownership

Recomendamos desabilitar ACLs nos buckets do Amazon S3. Faça isso aplicando a configuração aplicada pelo proprietário do bucket para S3 Object Ownership. Quando você aplica essa configuração, as ACLs são desabilitadas, e você automaticamente tem a propriedade e o controle total sobre todos os objetos de seu bucket. Você pode exigir que todos os novos buckets sejam criados com ACLs desabilitadas usando políticas do AWS Identity and Access Management (IAM) ou políticas de controle de serviço (SCPs) do AWS Organizations, conforme descrito na próxima seção.

Para aplicar a propriedade do objeto para novos objetos sem desabilitar ACLs, aplique a configuração preferencial do proprietário do bucket. Quando você aplica essa configuração, recomendamos enfaticamente atualizar a política de bucket para exigir a ACL `bucket-owner-full-control` pré-configurada para todas as solicitações PUT para seu bucket. Os clientes também devem ser atualizados para enviar a ACL `bucket-owner-full-control` pré-configurada de outras contas para seu bucket.

Tópicos

- [Desabilitar ACLs para todos os novos buckets \(aplicada pelo proprietário do bucket\)](#) (p. 660)
- [Exigir a ACL pré-configurada bucket-owner-full-control para operações PUT do Amazon S3 \(preferencial do proprietário do bucket\)](#) (p. 661)

Desabilitar ACLs para todos os novos buckets (aplicada pelo proprietário do bucket)

O exemplo a seguir de política do IAM nega a permissão `s3:CreateBucket` para um usuário ou uma função do IAM específica, a menos que a configuração aplicada pelo proprietário do bucket seja aplicada a Object Ownership. O par de chave-valor no bloco `Condition` especifica a chave `s3:x-amz-object-ownership` e a configuração `BucketOwnerEnforced` como valor. Em outras palavras, o usuário do IAM pode criar buckets somente se definir a configuração aplicada pelo proprietário do bucket para Object Ownership e desabilitar ACLs. Também é possível usar essa política como um SCP de limite para sua organização da AWS.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "RequireBucketOwnerFullControl",  
            "Action": "s3:CreateBucket",  
            "Effect": "Deny",  
            "Resource": "*",  
            "Condition": {  
                "StringNotEquals": {  
                    "s3:x-amz-object-ownership": "BucketOwnerEnforced"  
                }  
            }  
        }  
    ]  
}
```

Exigir a ACL pré-configurada bucket-owner-full-control para operações PUT do Amazon S3 (preferencial do proprietário do bucket)

Com a configuração preferencial do proprietário do bucket para Object Ownership, como proprietário do bucket, você tem a propriedade e o controle total sobre novos objetos que outras contas gravam em seu bucket com a ACL `bucket-owner-full-control` pré-configurada. Porém, se outras contas gravarem objetos em seu bucket sem a ACL `bucket-owner-full-control` pré-configurada, o gravador de objetos manterá o acesso de controle total. Como proprietário do bucket, você pode implementar uma política de bucket que permita gravações somente se especificarem a ACL `bucket-owner-full-control` pré-configurada.

Note

Caso tenha ACLs desabilitadas com a configuração aplicada pelo proprietário do bucket, como proprietário do bucket, você automaticamente terá propriedade e controle total sobre todos os objetos de seu bucket. Não é necessário usar esta seção para atualizar sua política de bucket a fim de aplicar a propriedade do objeto para o proprietário do bucket.

A política de bucket a seguir especifica que a conta `111122223333` pode fazer upload de objetos `DOC-EXAMPLE-BUCKET` somente quando a ACL do objeto estiver definida como `bucket-owner-full-control`. Substitua `111122223333` por uma conta real e `DOC-EXAMPLE-BUCKET` pelo nome de seu bucket.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Only allow writes to my bucket with bucket owner full control",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": [  
                    "arn:aws:iam::111122223333:user/ExampleUser"  
                ]  
            },  
            "Action": [  
                "s3:PutObject"  
            ],  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",  
            "Condition": {  
                "StringEquals": {  
                    "s3:x-amz-acl": "bucket-owner-full-control"  
                }  
            }  
        }  
    ]  
}
```

Veja a seguir um exemplo de operação de cópia que inclui a ACL padrão `bucket-owner-full-control` usando a AWS Command Line Interface (AWS CLI).

```
aws s3 cp file.txt s3://DOC-EXAMPLE-BUCKET --acl bucket-owner-full-control
```

Depois que a política de bucket entrar em vigor, se o cliente não incluir a ACL padrão `bucket-owner-full-control`, a operação falhará, e quem realizar o carregamento receberá o seguinte erro:

An error occurred (AccessDenied) when calling the PutObject operation: Access Denied (Ocorreu um erro (AccessDenied) ao chamar a operação PutObject: acesso negado.

Note

Se os clientes precisarem de acesso a objetos após o upload, será necessário conceder permissões adicionais à conta de upload. Para obter informações sobre como conceder a contas acesso aos seus recursos, consulte [Demonstrações de exemplo: gerenciar o acesso aos recursos do Amazon S3 \(p. 553\)](#).

Solução de problemas

Ao adotar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership, as listas de controle de acesso (ACLs) serão desabilitadas e, como proprietário do bucket, você assumirá a propriedade de todos os objetos de seu bucket. As ACLs não afetam mais as permissões dos objetos do bucket. Você pode usar políticas para conceder permissões. Todas as solicitações PUT do S3 devem especificar ACLs de controle total do proprietário do bucket ou não especificar uma ACL, senão elas falharão. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Se uma ACL inválida for especificada ou se as permissões de ACL de bucket concederem acesso fora de sua Conta da AWS, você poderá ver as respostas de erro a seguir.

AccessControlListNotSupported

Após adotar a configuração aplicada pelo proprietário do bucket em Object Ownership, as ACLs são desabilitadas. Solicitações para definir ACLs ou atualizar ACLs falham com um erro 400 e retornam o código de erro AccessControlListNotSupported. Ainda há suporte para solicitações de leitura de ACLs. As solicitações de leitura para ACLs sempre retornam uma resposta que mostra controle total para o proprietário do bucket. Nas operações PUT, é necessário especificar ACLs de controle total do proprietário do bucket ou não especificar uma ACL, senão as operações PUT falharão.

O exemplo a seguir de operação put-object usando a AWS CLI inclui a ACL public-read pré-configurada.

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key object-key-name --body doc-example-body --acl public-read
```

Se o bucket usar a configuração aplicada pelo proprietário do bucket para desabilitar ACLs, essa operação falhará, e a pessoa que fizer o carregamento receberá esta mensagem de erro:

An error occurred (AccessControlListNotSupported) when calling the PutObject operation: The bucket does not allow ACLs (Ocorreu um erro [AccessControlListNotSupported] ao chamar a operação PutObject: o bucket não permite ACLs)

InvalidBucketAclWithObjectOwnership

Para adotar a configuração aplicada pelo proprietário do bucket para desabilitar ACLs, a ACL do bucket deverá dar controle total somente ao proprietário do bucket. Sua ACL de bucket não pode dar acesso a uma Conta da AWS externa ou a qualquer outro grupo. Por exemplo, se a solicitação CreateBucket definir a configuração aplicada pelo proprietário do bucket e especificar uma ACL de bucket que fornece acesso a uma Conta da AWS externa, sua solicitação apresentará o erro 400 e retornará o código de erro InvalidBucketAclWithObjectOwnership. Da mesma forma, se a solicitação PutBucketOwnershipControls definir a configuração aplicada pelo proprietário do bucket em um bucket que tem uma ACL de bucket que concede permissões a outros, a solicitação falhará.

Example : ACL de bucket existente concede acesso público de leitura

Por exemplo, se uma ACL de bucket existente conceder acesso público de leitura, não será possível adotar a configuração aplicada pelo proprietário do bucket para Object Ownership até migrar essas

permissões de ACL para uma política de bucket e redefinir sua ACL de bucket para a ACL privada padrão. Para obter mais informações, consulte [Pré-requisitos para desabilitar ACLs \(p. 645\)](#).

Este exemplo de ACL de bucket concede acesso público de leitura:

```
{  
    "Owner": {  
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"  
    },  
    "Grants": [  
        {  
            "Grantee": {  
                "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",  
                "Type": "CanonicalUser"  
            },  
            "Permission": "FULL_CONTROL"  
        },  
        {  
            "Grantee": {  
                "Type": "Group",  
                "URI": "http://acs.amazonaws.com/groups/global/AllUsers"  
            },  
            "Permission": "READ"  
        }  
    ]  
}
```

Este exemplo de operação `put-bucket-ownership-controls` da AWS CLI adota a configuração aplicada pelo proprietário do bucket para Object Ownership:

```
aws s3api put-bucket-ownership-controls --bucket DOC-EXAMPLE-BUCKET --ownership-controls  
Rules=[{ObjectOwnership=BucketOwnerEnforced}]
```

Como a ACL de bucket concede acesso público de leitura, a solicitação falha e retorna o seguinte código de erro:

An error occurred (`InvalidBucketAclWithObjectOwnership`) when calling the `PutBucketOwnershipControls` operation: Bucket cannot have ACLs set with ObjectOwnership's BucketOwnerEnforced setting (Ocorreu um erro [`InvalidBucketAclWithObjectOwnership`] ao chamar a operação `PutBucketOwnershipControls`: o bucket não pode ter ACLs definidas com a configuração `BucketOwnerEnforced` do `ObjectOwnership`)

Registrar em log e monitorar no Amazon S3

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e a performance do Amazon S3 e das soluções da AWS. É necessário coletar dados de monitoramento de todas as partes de sua solução da AWS para depurar uma falha de vários pontos com mais facilidade, caso ocorra. A AWS fornece várias ferramentas para monitorar seus recursos do Amazon S3 e responder a possíveis incidentes.

Para obter mais informações, consulte [Monitorar o Amazon S3 \(p. 1020\)](#).

Alarmes do Amazon CloudWatch

Com o uso de alarmes do Amazon CloudWatch, você observa uma única métrica durante um período especificado. Se a métrica exceder determinado limite, uma notificação será enviada para um tópico do Amazon SNS ou para uma política do AWS Auto Scaling. Os alarmes do CloudWatch não invocam ações só porque estão em um determinado estado. O estado deve ter sido alterado e mantido por uma quantidade especificada de períodos. Para obter mais informações, consulte [Monitoramento de métricas com o Amazon CloudWatch \(p. 1067\)](#).

Logs do AWS CloudTrail

O CloudTrail fornece um registro de ações executadas por um usuário, uma função ou um produto da AWS no Amazon S3. Usando as informações coletadas pelo CloudTrail, é possível determinar 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 obter mais informações, consulte [Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail \(p. 1023\)](#).

Logs de acesso do Amazon S3

Os logs de acesso ao servidor fornecem registros detalhados das solicitações 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. Para obter mais informações, consulte [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#).

AWS Trusted Advisor

O Trusted Advisor conta com as melhores práticas aprendidas com o atendimento a centenas de milhões de clientes da AWS. O Trusted Advisor inspeciona seu ambiente da AWS e faz recomendações quando há oportunidades para economizar dinheiro, melhorar a performance e a disponibilidade do sistema e ajuda a corrigir falhas de segurança. Todos os clientes da AWS têm acesso a cinco verificações do Trusted Advisor. Os clientes com um plano de suporte Business ou Enterprise podem ver todas as verificações do Trusted Advisor.

A Trusted Advisor tem as seguintes verificações relacionadas ao Amazon S3:

- Configuração de registro em log de buckets do Amazon S3.
- Verificações de segurança para buckets do Amazon S3 que têm permissões de acesso livre.
- Verificações de tolerância a falhas para buckets do Amazon S3 que não têm versionamento habilitado ou têm versionamento suspenso.

Para obter mais informações, consulte [AWS Trusted Advisor](#) no [Guia do usuário do AWS Support](#).

As melhores práticas de segurança a seguir também abordam registros em logs e monitoramento:

- [Identify and audit all your Amazon S3 buckets](#)
- [Implement monitoring using Amazon Web Services monitoring tools](#)
- [Enable AWS Config](#)
- [Enable Amazon S3 server access logging](#)
- [Use CloudTrail](#)

- Monitor Amazon Web Services security advisories

Validação de conformidade para o Amazon S3

Auditores terceiros avaliam a segurança e a conformidade do Amazon S3 como parte de vários programas de conformidade da AWS, incluindo o seguinte:

- Controles do Sistema e da Organização (CSO)
- Padrão de segurança de dados do setor de cartão de pagamento (PCI DSS – Payment Card Industry Data Security Standard)
- Federal Risk and Authorization Management Program (FedRAMP)
- Health Insurance Portability and Accountability Act (HIPAA)

A AWS fornece uma lista atualizada com frequência de serviços da AWS no escopo de programas de conformidade específicos em [Serviços da AWS no escopo do programa de conformidade](#).

Os relatórios de auditoria de terceiros estão disponíveis para fazer download por meio do AWS Artifact. Para obter mais informações, consulte [Download de relatórios no AWS Artifact](#).

Para obter mais informações sobre programas de conformidade da AWS, consulte [Programas de conformidade da AWS](#).

Sua responsabilidade de conformidade ao usar o Amazon S3 é determinada pela confidencialidade de seus dados, pelas metas de conformidade da sua organização e pelas regulamentações e leis aplicáveis. Caso o uso do Amazon S3 esteja sujeito à conformidade com padrões como HIPAA, PCI ou FedRAMP, a AWS fornecerá os recursos para ajudar:

- [Guias de início rápido de segurança e compatibilidade](#) que discutem as considerações de arquitetura e fornecem etapas para a implantação de ambientes de linha de base focados em conformidade e segurança na AWS.
- O whitepaper [Architecting for HIPAA Security and Compliance](#) descreve como as empresas usam a AWS para ajudá-las a atender aos requisitos da HIPAA.
- [Recursos de conformidade da AWS](#) fornecem vários guias e manuais que podem se aplicar ao seu setor e local.
- [AWS Config](#) O pode ser usado para avaliar até que ponto suas configurações de recursos atendem adequadamente a práticas internas e a diretrizes e regulamentações do setor.
- O [Security Hub da AWS](#) fornece uma visão abrangente do seu estado de segurança na AWS que ajuda você a verificar a conformidade com os padrões e as melhores práticas do setor de segurança.
- O [Usar o bloqueio de objetos do S3 \(p. 722\)](#) pode ajudar a atender os requisitos de reguladores de serviços financeiros (como o SEC, FINRA e CFTC) que exigem armazenamento de dados WORM (Write Once, Read Many) para determinados tipos de informações gravadas.
- O [Inventário do Amazon S3 \(p. 786\)](#) pode ajudar você a auditar e gerar relatórios sobre o status da replicação e da criptografia de seus objetos para suas necessidades comerciais, de conformidade e regulatórias.

Resiliência no Amazon S3

A infraestrutura global da AWS é construída em torno de regiões e zonas de disponibilidade. As Regiões da AWS oferecem várias zonas de disponibilidade separadas e isoladas fisicamente que são conectadas com baixa latência, alto throughput e redes altamente redundantes. As zonas de disponibilidade oferecem a você uma forma eficiente para criar e operar aplicativos e bancos de dados. Elas são mais altamente disponíveis, tolerantes a falhas e escaláveis que infraestruturas com único data center ou infraestruturas com vários datacenters tradicionais. Se for necessário replicar seus dados especificamente em distâncias geográficas maiores, você poderá usar o [Replicação de objetos \(p. 801\)](#), que permite a cópia automática e assíncrona de objetos entre buckets de diferentes Regiões da AWS.

Cada Região da AWS tem várias zonas de disponibilidade. Você pode implantar suas aplicações em diversas zonas de disponibilidade na mesma região para tolerância de falha e baixa latência. As zonas de disponibilidade são conectadas umas às outras com redes de fibra ótica rápidas e privadas, que permitem o desenvolvimento de aplicações submetidas a failover automaticamente entre as zonas de disponibilidade, sem interrupções.

Para obter mais informações sobre Regiões da AWS e zonas de disponibilidade, consulte [Infraestrutura global da AWS](#).

Além da infraestrutura global da AWS, o Amazon S3 oferece vários recursos para oferecer suporte às suas necessidades de resiliência e backup de dados.

Configuração do ciclo de vida

Uma configuração de ciclo de vida é um conjunto de regras que definem as ações aplicadas pelo Amazon S3 a um grupo de objetos. 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. Para obter mais informações, consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

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. Para obter mais informações, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

Bloqueio de objetos do S3

Você pode usar o bloqueio de objetos do S3 para armazenar objetos usando um modelo gravar uma vez, ler muitas (WORM, write once read many). Ao usar o bloqueio de objetos do S3, você pode evitar que um objeto seja excluído ou substituído por um período fixo ou indefinidamente. O bloqueio de objetos do S3 permite atender aos requisitos regulamentares que exigem armazenamento WORM, ou simplesmente adicionar uma camada extra de proteção contra alterações e exclusões de objetos. Para obter mais informações, consulte [Usar o bloqueio de objetos do S3 \(p. 722\)](#).

Classes de armazenamento

O Amazon S3 oferece uma variedade de classes de armazenamento para escolher, de acordo com os requisitos da workload. As classes de armazenamento S3 Standard – IA e S3 One Zone – IA foram desenvolvidas para dados que você acessa cerca de uma vez por mês e precisam de acesso de milissegundos. A classe de armazenamento S3 Glacier Instant Retrieval foi projetada para dados de arquivo de longa duração com acesso de milissegundos que você acessa cerca de uma vez por trimestre. Para dados de arquivo que não necessitam de acesso imediato, como backups, use as classes de armazenamento S3 Glacier Flexieval ou S3 Glacier Deep Archive. Para obter mais informações, consulte [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#).

As melhores práticas de segurança a seguir também abordam resiliência:

- Enable versioning
- Consider Amazon S3 cross-region replication
- Identify and audit all your Amazon S3 buckets

Criptografia de backups do Amazon S3

Se estiver armazenando backups usando o Amazon S3, a criptografia dos backups dependerá da configuração desses buckets. O 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. A criptografia padrão oferece suporte a chaves armazenadas no AWS KMS (SSE-KMS). Para obter mais informações, consulte [Definir o comportamento padrão da criptografia para os buckets do Amazon S3 \(p. 142\)](#).

Para obter mais informações sobre versionamento e bloqueio de objetos, consulte os seguintes tópicos:
[Usando o versionamento em buckets do S3 \(p. 678\)](#) [Usar o bloqueio de objetos do S3 \(p. 722\)](#)

Segurança da infraestrutura no Amazon S3

Como um serviço gerenciado, o Amazon S3 é protegido pelo procedimento de segurança de redes global da AWS que são descritos no pilar de segurança do [AWS Well-Architected Framework](#).

O acesso ao Amazon S3 pela rede acontece por meio de APIs publicadas pela AWS. Os clientes devem ter suporte ao Transport Layer Security (TLS) 1.0. Recomendamos TLS 1.2. Os clientes também devem ter suporte a pacotes de criptografia com sigilo de encaminhamento perfeito (PFS) como Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Diffie-Hellman Encaminhamento (ECDHE). Além disso, as solicitações devem ser assinadas usando AWS Signature V4 ou AWS Signature V2, necessitando de credenciais válidas para o fornecimento.

Essas APIs podem ser chamadas de qualquer local da rede. No entanto, o Amazon S3 não oferece suporte a políticas de acesso com base em recursos, que podem incluir restrições com base no endereço IP de origem. Você também pode usar políticas de bucket do Amazon S3 para controlar o acesso a buckets a partir de endpoints específicos da virtual private cloud (VPC) ou de VPCs específicas. Efetivamente, isso isola o acesso à rede para um determinado bucket do Amazon S3 somente da VPC específica dentro da rede da AWS. Para obter mais informações, consulte [Controlar o acesso a partir de VPC endpoints com políticas de bucket \(p. 517\)](#).

As melhores práticas de segurança a seguir também abordam a segurança de infraestrutura no Amazon S3:

- Consider VPC endpoints for Amazon S3 access
- Identify and audit all your Amazon S3 buckets

Análise de configuração e vulnerabilidade no Amazon S3

A AWS se encarrega das tarefas básicas de segurança, como aplicação de patches a bancos de dados e sistemas operacionais (SOs) convidados, configuração de firewalls e recuperação de desastres. Esses procedimentos foram revisados e certificados por terceiros certificados. Para obter mais detalhes, consulte os seguintes recursos da :

- [Validação de conformidade para o Amazon S3 \(p. 666\)](#)
- [Modelo de responsabilidade compartilhada](#)
- [Amazon Web Services: visão geral dos processos de segurança](#)

As seguintes melhores práticas de segurança também abordam a análise de configuração e vulnerabilidade no Amazon S3:

- [Identify and audit all your Amazon S3 buckets](#)
- [Enable AWS Config](#)

Melhores práticas de segurança para o Amazon S3

O Amazon S3 fornece uma série de recursos de segurança a serem considerados no desenvolvimento e na implementação das suas próprias políticas de segurança. As melhores práticas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas práticas recomendadas podem não ser adequadas ou suficientes no seu ambiente, trate-as como considerações úteis em vez de requisitos.

Tópicos

- [Melhores práticas de segurança preventivas do Amazon S3 \(p. 672\)](#)
- [Melhores práticas de auditoria e monitoramento do Amazon S3 \(p. 675\)](#)

Melhores práticas de segurança preventivas do Amazon S3

As seguintes práticas recomendadas do Amazon S3 podem ajudar a evitar incidentes de segurança.

Garantir que seus buckets do Amazon S3 utilizem as políticas corretas e não sejam acessados publicamente

A não ser que seja absolutamente necessário que alguém na internet possa ler ou escrever no seu bucket do S3, o bucket do S3 não deve ser público. Você pode seguir algumas das etapas abaixo:

- Use o bloqueio de acesso público do Amazon S3. Com o Bloqueio de acesso público do Amazon S3, os administradores de conta e os proprietários de bucket podem configurar facilmente os controles centralizados impostos para limitar o acesso público aos recursos do Amazon S3, independentemente de como os recursos são criados. Para obter mais informações, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).
- Identifique as políticas de bucket do Amazon S3 que permitem uma identidade curinga, como entidade principal “*” (que efetivamente significa “qualquer pessoa”), ou que permitem uma ação curinga “*” (que efetivamente permite que o usuário realize qualquer ação no bucket do Amazon S3).
- Da mesma forma, observe as listas de controle de acesso (ACLs) do bucket do Amazon S3 que fornecem acesso de leitura, gravação, ou total a “Todos” ou a “Qualquer usuário autenticado da AWS”.
- Use a API `ListBuckets` para verificar todos os buckets do Amazon S3. Depois, use `GetBucketAcl`, `GetBucketWebsite` e `GetBucketPolicy` para determinar se o bucket tem controles de acesso e configuração dentro da conformidade.
- Use o [AWS Trusted Advisor](#) para inspecionar sua implementação do Amazon S3.
- Considere implementar os controles de detecção contínuos usando as regras `s3-bucket-public-read-prohibited` e `s3-bucket-public-write-prohibited` gerenciadas pelo AWS Config Rules.

Para obter mais informações, consulte [Identity and Access Management no Amazon S3 \(p. 419\)](#).

Implementar o acesso de privilégio mínimo

Ao conceder permissões, você decide quem receberá quais permissões para quais recursos do Amazon S3. Você habilita ações específicas que quer permitir nesses recursos. Portanto, você deve conceder somente as permissões necessárias para executar uma tarefa. A implementação do privilégio de acesso mínimo é fundamental para reduzir o risco de segurança e o impacto que pode resultar de erros ou usuários mal-intencionados.

As ferramentas a seguir estão disponíveis para implementar o privilégio de acesso mínimo:

- [Ações do Amazon S3 \(p. 441\)](#) e [Limites de permissões para entidades do IAM](#)
- [Políticas de bucket e políticas de usuário \(p. 437\)](#)
- [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#)
- [Políticas de controle de serviço](#)

Para obter orientação sobre o que considerar ao escolher um ou mais mecanismos precedentes, consulte [Diretrizes para políticas de acesso \(p. 425\)](#).

Usar funções do IAM para aplicações e serviços da AWS que exigem acesso ao Amazon S3

Para que aplicações no Amazon EC2 ou outros produtos da AWS acessem recursos do Amazon S3, eles devem incluir credenciais válidas da AWS em suas solicitações de API da AWS. Você não deve armazenar credenciais da AWS diretamente na aplicação ou na instância do Amazon EC2. Essas são credenciais de longo prazo que não são automaticamente alternadas e podem ter um impacto comercial significativo se forem comprometidas.

Em vez disso, você deve usar uma função do IAM para gerenciar credenciais temporárias para aplicações ou serviços que precisam de acesso ao Amazon S3. Quando você usa uma função, não é necessário distribuir credenciais de longo prazo (como um nome de usuário e uma senha ou chaves de acesso) para uma instância do Amazon EC2 ou produto da AWS como o AWS Lambda. Em vez disso, a função fornece permissões temporárias que os aplicativos podem usar ao fazer chamadas para outros recursos da AWS.

Para obter mais informações, consulte os seguintes tópicos no Manual do usuário do IAM:

- [Funções do IAM](#)
- [Cenários comuns para funções: usuários, aplicativos e serviços](#)

Considerar a criptografia de dados em repouso

Você tem as seguintes opções de proteção de dados em repouso no Amazon S3:

- Criptografia no lado do servidor: peça que o Amazon S3 criptografe o objeto antes de salvá-lo em discos em seus datacenters e descriptografe-o ao fazer download dos objetos. A criptografia no lado do servidor pode ajudar a reduzir o risco dos seus dados criptografando os dados com uma chave armazenada em um mecanismo diferente do mecanismo que armazena os próprios dados.

O Amazon S3 fornece estas opções de criptografia no lado do servidor:

- Criptografia no lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3).
- Criptografia no lado do servidor com chave do KMS armazenada no AWS Key Management Service (SSE-KMS)
- Criptografia no lado do servidor com chaves fornecidas pelo cliente (SSE-C).

Para obter mais informações, consulte [Proteção de dados usando criptografia no lado do servidor \(p. 359\)](#).

- Criptografia no lado do cliente: criptografe dados no lado do cliente e faça upload dos dados criptografados no Amazon S3. Nesse caso, você gerencia o processo de criptografia, as chaves de criptografia e as ferramentas relacionadas. Assim como na criptografia no lado do servidor, a criptografia no lado do cliente pode ajudar a reduzir o risco criptografando os dados com uma chave armazenada em um mecanismo diferente do mecanismo que armazena os próprios dados.

O Amazon S3 fornece muitas opções de criptografia no lado do cliente. Para obter mais informações, consulte [Proteger dados usando a criptografia no lado do cliente \(p. 405\)](#).

Aplicar a criptografia de dados em trânsito

Você pode usar HTTPS (TLS) para ajudar a evitar que invasores espionem ou manipulem tráfego de rede usando ataques person-in-the-middle (intermediários) ou similares. Você deve permitir apenas

conexões criptografadas por HTTPS (TLS) que usam a condição [aws:SecureTransport](#) em políticas de bucket do Amazon S3.

Considere também implementar controles de detecção contínuos usando a regra gerenciada [s3-bucket-ssl-requests-only](#) do AWS Config.

Considerar o bloqueio de objetos do S3

O [Usar o bloqueio de objetos do S3 \(p. 722\)](#) permite que você armazene objetos usando um modelo "Write-Once-Read-Man" (WORM gravação única e várias leituras)". O bloqueio de objetos do S3 pode ajudar a evitar a exclusão acidental ou inadequada de dados. Por exemplo, você pode usar o bloqueio de objetos do S3 para ajudar a proteger seus logs do AWS CloudTrail.

Habilitar o 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.

Considere também implementar controles de detecção em andamento usando a regra gerenciada [s3-bucket-versioning-enabled](#) do AWS Config.

Para obter mais informações, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

Considerar a replicação entre regiões do Amazon S3

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 (CRR) permite que você replique dados entre Regiões da AWS distantes para ajudar a satisfazer esses requisitos. A CRR permite a cópia assíncrona automática de objetos em buckets, em diferentes Regiões da AWS. Para obter mais informações, consulte [Replicação de objetos \(p. 801\)](#).

Note

A CRR requer que tanto o bucket de origem quanto o de destino do S3 tenham o versionamento habilitado.

Considere também implementar controles de detecção em andamento usando a regra gerenciada [s3-bucket-replication-enabled](#) do AWS Config.

Considerar os endpoints da VPC para acesso ao Amazon S3

Um VPC endpoint para o Amazon S3 é uma entidade lógica dentro de uma VPC que permite a conectividade somente com o Amazon S3. Você pode usar políticas de bucket do Amazon S3 para controlar o acesso a buckets de VPC endpoints específicos ou VPCs específicas. Um VPC endpoint também pode ajudar a evitar que o tráfego potencialmente atravesse a internet aberta e esteja sujeito ao ambiente da internet aberta.

Os VPC endpoints para Amazon S3 fornecem várias 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.
- Você pode controlar quais VPCs ou VPC endpoints têm acesso a seus buckets do S3 usando as políticas de bucket do S3.
- Você pode ajudar a evitar a exfiltração de dados usando uma VPC que não tenha um internet gateway.

Para obter mais informações, consulte [Controlar o acesso a partir de VPC endpoints com políticas de bucket \(p. 517\)](#).

Melhores práticas de auditoria e monitoramento do Amazon S3

As práticas recomendadas a seguir para o Amazon S3 podem ajudar a detectar pontos fracos e incidentes potenciais de segurança.

Identificar e auditar todos os buckets do Amazon S3

A identificação de seus ativos de TI é um aspecto essencial de governança e segurança. É necessário ter visibilidade de seus recursos do Amazon S3 para avaliar sua postura de segurança e agir em possíveis áreas de fraqueza.

Use o Tag Editor para identificar recursos sensíveis quanto a segurança ou auditoria, depois use essas tags quando precisar procurar por esses recursos. Para obter mais informações, consulte [Pesquisa de recursos para marcar com tags](#).

Você pode usar o Amazon S3 Inventory para auditar e gerar relatórios sobre o status da replicação e criptografia de seus objetos para os negócios, a conformidade e as necessidades normativas. Para obter mais informações, consulte [Inventário do Amazon S3 \(p. 786\)](#).

Crie grupos de recursos para seus recursos do Amazon S3. Para obter mais informações, consulte [O que são Resource Groups da AWS?](#)

Implementar monitoramento usando ferramentas de monitoramento da AWS

O monitoramento é uma parte importante para manter a confiabilidade, a segurança, a disponibilidade e a performance do Amazon S3 e suas soluções da AWS. A AWS fornece várias ferramentas e serviços para ajudar a monitorar o Amazon S3 e outros serviços da AWS. Por exemplo, você pode monitorar métricas do CloudWatch para o Amazon S3, particularmente, `PutRequests`, `GetRequests`, `4xxErrors` e `DeleteRequests`. Para obter mais informações, consulte [Monitoramento de métricas com o Amazon CloudWatch \(p. 1067\)](#) e [Monitorar o Amazon S3 \(p. 1020\)](#).

Para obter um segundo exemplo, consulte, [Exemplo: atividade do bucket do Amazon S3](#). Este exemplo descreve como criar um alarme do Amazon CloudWatch que é ativado quando uma chamada API do Amazon S3 é feita para a política PUT ou DELETE do bucket, ciclo de vida do bucket, ou replicação do bucket ou para implementar um ACL do bucket.

Habilitar o registro em log de acesso ao servidor do 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 podem ajudar na auditoria de segurança e acesso, a saber mais sobre a base de clientes e entender sua fatura do Amazon S3. Para obter informações sobre como habilitar o registro em log de acesso ao servidor, consulte [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#).

Considere também implementar controles de detecção em andamento usando a regra gerenciada `s3-bucket-logging-enabled` do AWS Config.

Usar o AWS CloudTrail

O AWS CloudTrail fornece um registro das ações executadas por um usuário, uma função ou um produto da AWS no Amazon S3. Você pode usar as informações coletadas pelo CloudTrail para determinar 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. Por exemplo, você pode identificar entradas do CloudTrail para ações Put que afetam o acesso a dados, em particular, `PutBucketAcl`, `PutObjectAcl`, `PutBucketPolicy` e `PutBucketWebsite`. Quando você configura sua Conta da AWS, o CloudTrail é habilitado por padrão. Você pode visualizar eventos recentes no console do

CloudTrail. Para criar um registro contínuo de atividades e eventos para os buckets do Amazon S3, você pode criar uma trilha no console do CloudTrail. Para obter mais informações, consulte [Registro de eventos de dados em logs para trilhas](#) no Guia do usuário do AWS CloudTrail.

Ao criar uma trilha, você pode configurar o CloudTrail para registrar em log eventos de dados. Eventos de dados são registros de operações de recurso executadas no recurso ou dentro de um recurso. No Amazon S3, os eventos de dados registram a atividade da API no nível do objeto para buckets individuais. O CloudTrail oferece suporte a um subconjunto de operações de API no nível do objeto do Amazon S3 como `GetObject`, `DeleteObject` e `PutObject`. Para obter mais informações sobre como o CloudTrail funciona com o Amazon S3, consulte [Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail](#) (p. 1023). No console do Amazon S3, você também pode configurar seus buckets do S3 para [Habilitar o log de eventos do CloudTrail para buckets e objetos do S3](#) (p. 1033).

O AWS Config fornece uma regra gerenciada (`cloudtrail-s3-dataevents-enabled`) que você pode usar para confirmar que pelo menos uma trilha do CloudTrail está registrando eventos de dados para seus buckets do S3. Para obter mais informações, consulte [cloudtrail-s3-dataevents-enabled](#) no Guia do desenvolvedor do AWS Config.

Habilitar AWS Config

Muitas das práticas recomendadas listadas neste tópico sugerem criar regras do AWS Config. O AWS Config permite que você analise, faça auditoria e avalie as configurações de seus recursos da AWS. O AWS Config monitora as configurações de recurso, permitindo que você avalie as configurações gravadas, comparando-as com as configurações seguras desejadas. Com o AWS Config, você pode analisar alterações feitas nas configurações e relacionamentos entre os recursos da AWS, examinar os detalhes do histórico de configuração de recursos e determinar a conformidade geral em relação às configurações especificadas em diretrizes internas. Isso pode ajudar a simplificar a auditoria de conformidade, a análise de segurança, o gerenciamento de alterações e a solução de problemas operacionais. Para obter mais informações, consulte [Configuração do AWS Config com o console](#) no Guia do desenvolvedor do AWS Config. Ao especificar os tipos de recurso que devem ser registrados, inclua os recursos do Amazon S3.

Para obter um exemplo de como usar o AWS Config para monitorar e responder aos buckets do Amazon S3 que permitem acesso público, consulte [Como usar o AWS Config para monitorar e responder a buckets do Amazon S3 que permitem acesso público](#) no Blog de segurança da AWS.

Considerar a possibilidade de usar o Amazon Macie com o Amazon S3

O Amazon Macie é um serviço de segurança e privacidade de dados que usa machine learning e comparação de padrões para ajudar você a descobrir, monitorar e proteger seus dados confidenciais no ambiente da AWS. O Macie automatiza a descoberta de dados sigilosos, como informações de identificação pessoal (PII) e dados financeiros, para você compreender melhor os dados armazenados por sua organização no Amazon S3.

O Macie também fornece um inventário de seus buckets do S3 e os avalia e monitora automaticamente para segurança e controle de acesso. Se o Macie detectar dados confidenciais ou possíveis problemas com a segurança ou a privacidade de seus dados, ele criará descobertas detalhadas para você revisar e corrigir conforme necessário. Para obter mais informações, consulte [O que é o Amazon Macie?](#)

Monitorar as recomendações de segurança da AWS

Verifique regularmente as recomendações de segurança publicadas no Trusted Advisor para sua Conta da AWS. Especificamente, observe avisos sobre buckets do Amazon S3 com “permissões de acesso em aberto”. Você pode fazer isso programaticamente usando [describe-trusted-advisor-checks](#).

Além disso, monitore ativamente o endereço de e-mail registrado como principal para cada uma de suas Contas da AWS. A AWS usará esse e-mail para entrar em contato e notificá-lo sobre os problemas de segurança que surgiem e que possam afetar você.

Problemas operacionais da AWS com grande impacto são publicados no [AWS Service Health Dashboard](#). Problemas operacionais também são publicados em contas individuais por meio do Personal Health Dashboard. Para obter mais informações, consulte a [documentação da AWS Health](#).

Gerenciar seu armazenamento do Amazon S3

Depois de criar buckets e carregar objetos no Amazon S3, você pode gerenciar seu armazenamento de objetos usando recursos como versionamento, classes de armazenamento, bloqueio de objetos, operações em lote, replicação, tags e muito mais. As seções a seguir fornecem informações detalhadas sobre os recursos e recursos de gerenciamento de armazenamento disponíveis no Amazon S3.

Tópicos

- [Usando o versionamento em buckets do S3 \(p. 678\)](#)
- [Usar o AWS Backup para o Amazon S3 \(p. 711\)](#)
- [Trabalhando com objetos arquivados \(p. 712\)](#)
- [Usar o bloqueio de objetos do S3 \(p. 722\)](#)
- [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#)
- [Amazon S3 Intelligent-Tiering \(p. 737\)](#)
- [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#)
- [Inventário do Amazon S3 \(p. 786\)](#)
- [Replicação de objetos \(p. 801\)](#)
- [Categorizando seu armazenamento usando tags \(p. 878\)](#)
- [Usar tags de alocação de custos para buckets do S3 \(p. 888\)](#)
- [Filtragem e recuperação de dados usando o Amazon S3 Select \(p. 908\)](#)
- [Executar operações em lote de grande escala em objetos do Amazon S3 \(p. 938\)](#)

Usando o versionamento em buckets do S3

Versionamento no Amazon S3 é um meio de manter diversas variantes de um objeto no mesmo bucket. Você pode usar o recurso S3 Versioning para preservar, recuperar e restaurar todas as versões de cada objeto armazenado em seus buckets. Com o versionamento, você pode se recuperar mais facilmente de ações não intencionais do usuário e de falhas da aplicação. Depois de habilitar o versionamento para um bucket, se o Amazon S3 receber várias solicitações de gravação do mesmo objeto simultaneamente, ele armazenará todos esses objetos.

Buckets com versionamento habilitado podem ajudar a recuperar objetos de uma exclusão ou substituição acidental. Por exemplo, se você excluir um objeto, o Amazon S3 inserirá um marcador de exclusão em vez de remover o objeto permanentemente. O marcador de exclusão se torna a versão atual do objeto. Se você substituir um objeto, isso criará uma nova versão do objeto no bucket. Você sempre pode restaurar a versão anterior. Para obter mais informações, consulte [Excluir versões de objetos de um bucket com versionamento habilitado \(p. 700\)](#).

Por padrão, o Versionamento do S3 está desativado em buckets e você deve ativá-lo explicitamente. Para obter mais informações, consulte [Habilitar o versionamento em buckets \(p. 683\)](#).

Note

- A API SOAP não é compatível com o versionamento do S3. O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com SOAP.
- As taxas normais do Amazon S3 se aplicam 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.

Buckets não versionados, habilitados para versão e suspensos de versão

Os buckets podem estar em um dos três estados:

- Não versionado (o padrão)
- Habilitado para versão
- Com versionamento suspenso

Você habilita e suspende o versionamento no nível do bucket. Depois que um bucket é habilitado para versionamento, ele nunca pode voltar a um estado sem versionamento. Mas você pode suspender o versionamento nesse bucket.

O estado de versionamento aplica-se a todos (nunca alguns) os objetos nesse bucket. Quando você habilita o versionamento em um bucket, todos os novos objetos são versionados e recebem um ID de versão exclusivo. Os objetos que já existiam no bucket no momento em que o versionamento foi ativado serão sempre versionados e receberão um ID de versão exclusivo quando forem modificados por solicitações futuras. Observe o seguinte:

- 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. Para obter mais informações, consulte [Trabalhar com objetos em um bucket com versionamento habilitado \(p. 689\)](#).
- 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 [Trabalhar com objetos em um bucket com versionamento suspenso \(p. 708\)](#).

Usando o versionamento do S3 com o ciclo de vida do S3

Para personalizar sua abordagem de retenção de dados e controlar os custos de armazenamento, use o versionamento de objetos com o S3 Lifecycle. Para obter mais informações, consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#). Para obter informações sobre como criar políticas de ciclo de vida do S3 usando o AWS Management Console, a AWS CLI, os AWS SDKs ou a API REST, consulte [Definir a configuração do ciclo de vida em um bucket \(p. 753\)](#).

Important

Se você tem uma política de ciclo de vida de expiração do objeto em seu bucket sem versionamento e quer manter o mesmo comportamento de exclusão permanente quando ativar o versionamento, precisará adicionar uma política de expiração de versão desatualizada. A política de expiração do ciclo de vida gerencia as exclusões de versões desatualizadas de objeto no bucket habilitado para versionamento. (Um bucket habilitado para versionamento mantém uma versão atual e zero ou mais versões desatualizadas de objetos.) Para obter mais informações, consulte [Definir a configuração do ciclo de vida em um bucket \(p. 753\)](#).

Para obter informações sobre como trabalhar com o Versionamento do S3, consulte os tópicos a seguir.

Tópicos

- [Como funciona o Versionamento do S3 \(p. 680\)](#)
- [Habilitar o versionamento em buckets \(p. 683\)](#)
- [Configurando a exclusão de MFA \(p. 688\)](#)
- [Trabalhar com objetos em um bucket com versionamento habilitado \(p. 689\)](#)
- [Trabalhar com objetos em um bucket com versionamento suspenso \(p. 708\)](#)

Como funciona o Versionamento do S3

Você pode usar o versionamento do S3 para manter várias versões de um objeto em um bucket e permitir que você restaure objetos excluídos ou substituídos acidentalmente. Por exemplo, se você excluir um objeto em vez de removê-lo permanentemente, o Amazon S3 inserirá um marcador de exclusão, o que se torna a versão atual do objeto. Você pode então restaurar a versão anterior. Para obter mais informações, consulte [Excluir versões de objetos de um bucket com versionamento habilitado \(p. 700\)](#). Se você substituir um objeto, isso criará uma nova versão do objeto no bucket. Você sempre pode restaurar a versão anterior.

Note

As taxas normais do Amazon S3 se aplicam 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.

Cada bucket do S3 que você cria tem um sub-recurso de versionamento associado a ele. (Para obter mais informações, consulte [Opções de configuração do bucket \(p. 125\)](#).) Por padrão, seu bucket não está habilitado para versionamento e o sub-recurso de versionamento armazena uma configuração vazia de versionamento, como mostrado a seguir.

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

Para habilitar o versionamento, você pode enviar 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`.

Note

Se você ativar o versionamento em um bucket pela primeira vez, pode levar um curto período de tempo para que a alteração seja totalmente propagada. Recomendamos que você aguarde 15 minutos após ativar o versionamento antes de emitir operações de gravação (PUT ou DELETE) em objetos no bucket.

O proprietário do bucket e todos os usuários autorizados do IAM podem habilitar o versionamento. O proprietário do bucket é a Conta da AWS que criou o bucket (a conta root). Para obter mais informações sobre permissões, consulte [Identity and Access Management no Amazon S3 \(p. 419\)](#).

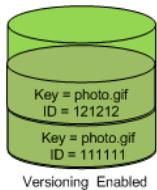
Para obter mais informações sobre como ativar e desativar o S3 Versioning usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) ou a API REST, consulte [the section called “Habilitar o versionamento em buckets” \(p. 683\)](#).

Tópicos

- [IDs de versão \(p. 681\)](#)
- [Fluxos de trabalho de versionamento \(p. 681\)](#)

IDs de versão

Se o versionamento estiver habilitado para um bucket, o Amazon S3 gerará um ID de versão exclusivo automaticamente para o objeto que está sendo armazenado. 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).



Cada objeto tem um ID de versão, mesmo que o versionamento do S3 não esteja habilitado. Se o versionamento do S3 não estiver habilitado, o Amazon S3 definirá o valor do ID da versão como null. Se o S3 Versioning for habilitado, o Amazon S3 atribuirá um valor de ID de versão para o objeto. Esse valor o distingue de outras versões da mesma chave.

Quando você habilita o S3 Versioning em um bucket existente, os objetos que já estão armazenados no bucket permanecem inalterados. Os IDs de versão (nulos), o conteúdo e as permissões continuarão os mesmos. Depois que habilitar o versionamento do S3 para um bucket, cada objeto adicionado ao bucket obterá um ID de versão, que o distinguirá de outras versões da mesma chave.

Somente o Amazon S3 gera IDs de versão, e eles não podem ser editados. Os IDs de versão são strings opacas Unicode, com codificação UTF-8 e prontas para URL que não têm mais de 1.024 bytes de comprimento. Veja um exemplo a seguir:

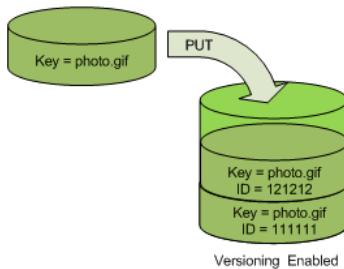
```
3sL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCxwf3vjVBH40Nr8X8gdRQBpUMLUo
```

Note

Por simplicidade, os outros exemplos neste tópico usam IDs muito mais curtos.

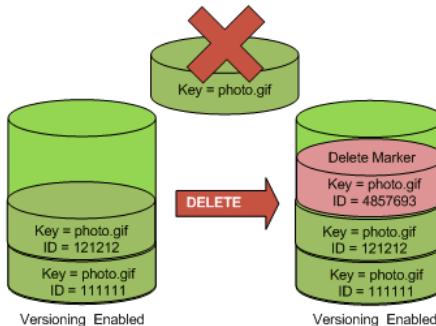
Fluxos de trabalho de versionamento

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 original (ID = 111111) permanece no bucket, o Amazon S3 gera um ID da nova versão (121212) e adiciona a versão mais recente ao bucket.

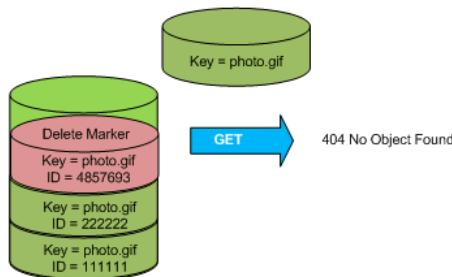


Essa funcionalidade impede que você substitua ou exclua objetos acidentalmente e permite que você recupere uma versão anterior de um objeto.

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.

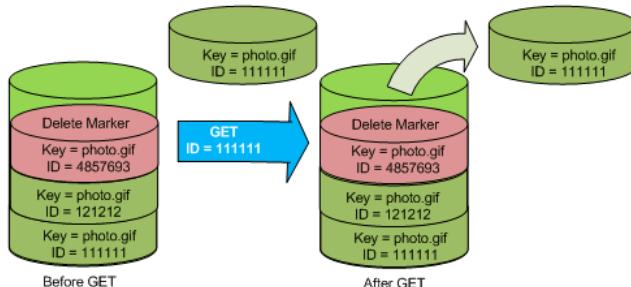


O marcador de exclusão torna-se a versão atual de objeto. Por padrão, o **GET** requisita a recuperação da versão armazenada mais recente. A execução de uma solicitação **GET Object** simples quando a versão atual é um marcador de exclusão retorna um erro **404 Not Found**, conforme exibido na figura a seguir.

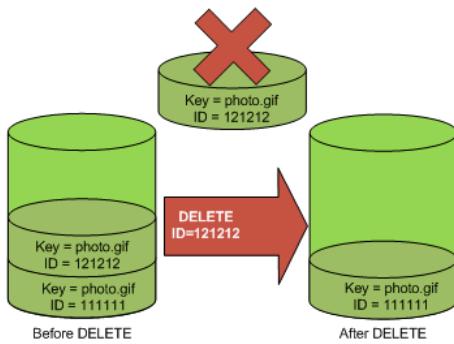


Você pode, contudo, fazer uma solicitação **GET** de uma versão desatualizada de um objeto por especificar seu ID de versão. Na figura a seguir, você faz uma solicitação **GET** de uma versão específica de objeto, 111111. O Amazon S3 retorna a versão desse objeto, apesar de não ser a versão atual.

Para obter mais informações, consulte [Recuperando versões de objeto de um bucket habilitado para versionamento \(p. 695\)](#).



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** exclui, 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 habilitar 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 [Configurando a exclusão de MFA \(p. 688\)](#).

Novas versões são criadas somente quando você `PUT` um novo objeto. Esteja ciente de que certas ações como `COPY` funcionam implementando `PUT`. Tomar ações que modifiquem o objeto atual não criará uma nova versão porque elas não `PUT` um novo objeto. Isso inclui ações como alterar as tags em um objeto.

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 o versionamento do S3 habilitado, talvez tenha um ou mais objetos no bucket para os quais há milhões de versões. Para obter mais informações, consulte [Aumentos significativos em respostas HTTP 503 para solicitações do Amazon S3 para buckets com o versionamento habilitado \(p. 1610\)](#) na seção Solução de problemas.

Habilitar o versionamento em buckets

Use o S3 Versioning para manter várias versões de um objeto em um bucket. Esta seção fornece exemplos de como habilitar o versionamento em um bucket usando o console, a API REST, os AWS SDKs e a AWS Command Line Interface (AWS CLI).

Note

Se você ativar o versionamento em um bucket pela primeira vez, pode levar um curto período de tempo para que a alteração seja totalmente propagada. Recomendamos que você aguarde 15 minutos após ativar o versionamento antes de emitir operações de gravação (`PUT` ou `DELETE`) em objetos no bucket.

Para obter mais informações sobre o S3 Versioning, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#). Para obter informações sobre como trabalhar com objetos que estão em buckets habilitados para versão, consulte [Trabalhar com objetos em um bucket com versionamento habilitado \(p. 689\)](#).

Cada bucket do S3 que você cria tem um sub-recurso de versionamento associado a ele. (Para obter mais informações, consulte [Opções de configuração do bucket \(p. 125\)](#).) Por padrão, seu bucket não está habilitado para versionamento e o sub-recurso de versionamento armazena uma configuração vazia de versionamento, como mostrado a seguir.

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

Para habilitar o versionamento, você pode enviar 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 e todos os usuários autorizados do IAM podem habilitar o versionamento. O proprietário do bucket é a Conta da AWS que criou o bucket (a conta root). Para obter mais informações sobre permissões, consulte [Identity and Access Management no Amazon S3 \(p. 419\)](#).

As seções a seguir fornecem mais detalhes sobre como ativar o versionamento do S3 usando o console, a AWS CLI e os AWS SDKs.

Uso do console do S3

Siga estas etapas para usar o AWS Management Console para habilitar o versionamento em um bucket do S3.

Para habilitar ou desabilitar o versionamento em um bucket do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket para o qual você deseja habilitar o versionamento.
3. Escolha Properties (Propriedades).
4. Em Bucket Versioning (Versionamento de bucket), escolha Edit (Editar).
5. Escolha Suspend (Suspender) ou Enable (Ativar), e selecione Save changes (Salvar alterações).

Note

Você pode usar a autenticação multifator (MFA) da AWS com versionamento. Ao usar a MFA com versionamento, você deve fornecer as chaves de acesso da sua Conta da AWS e um código válido do dispositivo MFA da conta para excluir permanentemente uma versão de objeto ou suspender ou reativar o versionamento.

Para usar a MFA com versionamento, você habilita **MFA Delete**. No entanto, não é possível habilitar o **MFA Delete** usando o AWS Management Console. É necessário usar a AWS Command Line Interface (AWS CLI) ou API. Para obter mais informações, consulte [Configurando a exclusão de MFA \(p. 688\)](#).

Usar a AWS CLI

O exemplo a seguir habilita o versionamento em um bucket do S3.

```
aws s3api put-bucket-versioning --bucket DOC-EXAMPLE-BUCKET1 --versioning-configuration
Status=Enabled
```

O exemplo a seguir ativa o versionamento do S3 e a exclusão de autenticação multifator (MFA) em um bucket.

```
aws s3api put-bucket-versioning --bucket DOC-EXAMPLE-BUCKET1 --versioning-configuration
Status=Enabled,MFADelete=Enabled --mfa "SERIAL 123456"
```

Note

O uso da exclusão de MFA requer um dispositivo de autenticação física ou virtual aprovado. Para obter mais informações sobre como usar exclusão de MFA no Amazon S3, consulte [Configurando a exclusão de MFA \(p. 688\)](#).

Para obter mais informações sobre como ativar o versionamento usando a AWS CLI, consulte [put-bucket-versioning](#) na Referência de comandos da AWS CLI.

Uso da SDKs AWS

Os exemplos a seguir permitem o versionamento em um bucket e, em seguida, recuperam o status do versionamento usando o AWS SDK for Java e o AWS SDK for .NET. Para obter informações sobre o uso de outros AWS SDKs, consulte o [Centro do Desenvolvedor da AWS](#).

.NET

Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
using System;
using Amazon.S3;
using Amazon.S3.Model;

namespace s3.amazon.com.docsamples
{
    class BucketVersioningConfiguration
    {
        static string bucketName = "*** bucket name ***";

        public static void Main(string[] args)
        {
            using (var client = new AmazonS3Client(Amazon.RegionEndpoint.USEast1))
            {
                try
                {
                    EnableVersioningOnBucket(client);
                    string bucketVersioningStatus =
                    RetrieveBucketVersioningConfiguration(client);
                }
                catch (AmazonS3Exception amazonS3Exception)
                {
                    if (amazonS3Exception.ErrorCode != null &&
                        (amazonS3Exception.ErrorCode.Equals("InvalidAccessKeyId") ||
                        amazonS3Exception.ErrorCode.Equals("InvalidSecurity")))
                    {
                        Console.WriteLine("Check the provided AWS Credentials.");
                        Console.WriteLine(
                            "To sign up for service, go to http://aws.amazon.com/s3");
                    }
                    else
                    {
                        Console.WriteLine(
                            "Error occurred. Message:{0}' when listing objects",
                            amazonS3Exception.Message);
                    }
                }
            }

            Console.WriteLine("Press any key to continue...");
            Console.ReadKey();
        }
    }
}
```

```
static void EnableVersioningOnBucket(IAmazonS3 client)
{
    PutBucketVersioningRequest request = new PutBucketVersioningRequest
    {
        BucketName = bucketName,
        VersioningConfig = new S3BucketVersioningConfig
        {
            Status = VersionStatus.Enabled
        }
    };

    PutBucketVersioningResponse response =
client.PutBucketVersioning(request);
}

static string RetrieveBucketVersioningConfiguration(IAmazonS3 client)
{
    GetBucketVersioningRequest request = new GetBucketVersioningRequest
    {
        BucketName = bucketName
    };

    GetBucketVersioningResponse response =
client.GetBucketVersioning(request);
    return response.VersioningConfig.Status;
}
}
```

Java

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
import java.io.IOException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Region;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.AmazonS3Exception;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.SetBucketVersioningConfigurationRequest;

public class BucketVersioningConfigurationExample {
    public static String bucketName = "*** bucket name ***";
    public static AmazonS3Client s3Client;

    public static void main(String[] args) throws IOException {
        s3Client = new AmazonS3Client(new ProfileCredentialsProvider());
        s3Client.setRegion(Region.getRegion(Regions.US_EAST_1));
        try {

            // 1. Enable versioning on the bucket.
            BucketVersioningConfiguration configuration =
                new BucketVersioningConfiguration().withStatus("Enabled");

            SetBucketVersioningConfigurationRequest setBucketVersioningConfigurationRequest =
                new SetBucketVersioningConfigurationRequest(bucketName, configuration);

            s3Client.setBucketVersioningConfiguration(setBucketVersioningConfigurationRequest);
        }
    }
}
```

```
// 2. Get bucket versioning configuration information.  
BucketVersioningConfiguration conf =  
s3Client.getBucketVersioningConfiguration(bucketName);  
System.out.println("bucket versioning configuration status: " +  
conf.getStatus());  
  
} catch (AmazonS3Exception amazonS3Exception) {  
    System.out.format("An Amazon S3 error occurred. Exception: %s",  
amazonS3Exception.toString());  
} catch (Exception ex) {  
    System.out.format("Exception: %s", ex.toString());  
}  
}  
}
```

Python

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Usar a AWS SDK for Python \(Boto\) \(p. 1271\)](#).

O exemplo de código Python a seguir cria um bucket do Amazon S3 bucket, habilita-o para versionamento e configura um ciclo de vida que faz com que versões não atuais de objetos expirem após 7 dias.

```
def create_versioned_bucket(bucket_name, prefix):  
    """  
    Creates an Amazon S3 bucket, enables it for versioning, and configures a lifecycle  
    that expires noncurrent object versions after 7 days.  
  
    Adding a lifecycle configuration to a versioned bucket is a best practice.  
    It helps prevent objects in the bucket from accumulating a large number of  
    noncurrent versions, which can slow down request performance.  
  
    Usage is shown in the usage_demo_single_object function at the end of this module.  
  
    :param bucket_name: The name of the bucket to create.  
    :param prefix: Identifies which objects are automatically expired under the  
        configured lifecycle rules.  
    :return: The newly created bucket.  
    """  
    try:  
        bucket = s3.create_bucket(  
            Bucket=bucket_name,  
            CreateBucketConfiguration={  
                'LocationConstraint': s3.meta.client.meta.region_name  
            }  
        )  
        logger.info("Created bucket %s.", bucket.name)  
    except ClientError as error:  
        if error.response['Error']['Code'] == 'BucketAlreadyOwnedByYou':  
            logger.warning("Bucket %s already exists! Using it.", bucket_name)  
            bucket = s3.Bucket(bucket_name)  
        else:  
            logger.exception("Couldn't create bucket %s.", bucket_name)  
            raise  
  
    try:  
        bucket.Versioning().enable()  
        logger.info("Enabled versioning on bucket %s.", bucket.name)  
    except ClientError:  
        logger.exception("Couldn't enable versioning on bucket %s.", bucket.name)  
        raise  
  
    try:
```

```
        expiration = 7
        bucket.LifecycleConfiguration().put(
            LifecycleConfiguration={
                'Rules': [
                    {
                        'Status': 'Enabled',
                        'Prefix': prefix,
                        'NoncurrentVersionExpiration': {'NoncurrentDays': expiration}
                    }
                ]
            }
        )
        logger.info("Configured lifecycle to expire noncurrent versions after %s days " %
                    "on bucket %s.", expiration, bucket.name)
    except ClientError as error:
        logger.warning("Couldn't configure lifecycle on bucket %s because %s. " %
                       "Continuing anyway.", bucket.name, error)

    return bucket
```

Configurando a exclusão de MFA

Ao trabalhar com o Versionamento do S3 em buckets do Amazon S3, você pode, opcionalmente, adicionar outra camada de segurança configurando um bucket para habilitar 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.

A exclusão de MFA requer autenticação adicional para 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 se, por exemplo, suas credenciais de segurança forem comprometidas. A exclusão de MFA pode ajudar a evitar exclusões acidentais de bucket exigindo que o usuário que inicia a ação de exclusão para provar a posse física de um dispositivo MFA com um código MFA e adicionando uma camada extra de atrito e segurança à ação de exclusão.

O proprietário do bucket, a Conta da AWS que criou o bucket (conta root) e todos os usuários autorizados do IAM podem habilitar o versionamento. No entanto, somente o proprietário do bucket (conta raiz) pode habilitar a exclusão de MFA. Para obter mais informações, consulte a publicação [Proteção do acesso à AWS usando MFA](#) no Blog de segurança da AWS.

Note

Para usar a exclusão de MFA com versionamento, habilite **MFA Delete**. No entanto, não é possível habilitar o **MFA Delete** usando o AWS Management Console. É necessário usar a AWS Command Line Interface (AWS CLI) ou API.

Para obter exemplos de como usar a exclusão de MFA com versionamento, consulte a seção de exemplos no tópico [Habilitar o versionamento em buckets \(p. 683\)](#).

Não é possível usar a exclusão de MFA com configurações de ciclo de vida. Para obter mais informações sobre as configurações de ciclo de vida e como elas interagem com outras configurações, consulte [Ciclo de vida e outras configurações de bucket \(p. 765\)](#).

Para ativar ou desativar a exclusão de MFA, use a mesma API usada 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.

```
<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.



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 ajudar a 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 exigir que qualquer operação nesses recursos do Amazon S3 seja feita com credenciais temporárias criadas usando a MFA. Para ver um exemplo, consulte [Adicionar uma política de bucket para exigir MFA \(p. 524\)](#).

Para obter mais informações sobre como comprar e ativar um dispositivo de autenticação, consulte [Autenticação multifator](#).

Como ativar o versionamento do S3 e configurar a exclusão de MFA

Usar a AWS CLI

O exemplo a seguir ativa o versionamento do S3 e a exclusão de autenticação multifator (MFA) em um bucket.

```
aws s3api put-bucket-versioning --bucket DOC-EXAMPLE-BUCKET1 --versioning-configuration
Status=Enabled,MFADelete=Enabled --mfa "SERIAL 123456"
```

Uso dos REST API

Para obter mais informações sobre como especificar a exclusão de MFA usando a API REST do Amazon S3, consulte [PutBucketVersioning](#) na Referência de API do Amazon Simple Storage Service.

Trabalhar com objetos em um bucket com versionamento habilitado

Os objetos que são armazenados em um bucket do Amazon S3 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.

Fazer a 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 S3 `Glacier Flexible`

Retrieval em um momento específico no ciclo de vida do objeto. Para obter mais informações, consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

Os tópicos nesta seção explicam várias operações de objeto em um bucket com versionamento ativado. Para obter mais informações sobre versionamento, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

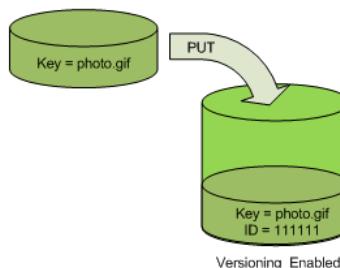
Tópicos

- [Adicionar objetos a buckets com versionamento habilitado \(p. 690\)](#)
- [Listar objetos em um bucket com versionamento habilitado \(p. 691\)](#)
- [Recuperando versões de objeto de um bucket habilitado para versionamento \(p. 695\)](#)
- [Excluir versões de objetos de um bucket com versionamento habilitado \(p. 700\)](#)
- [Configurando permissões de objeto com versão \(p. 707\)](#)

Adicionar objetos a buckets com versionamento habilitado

Depois que você habilita o versionamento em um bucket, o Amazon S3 adiciona automaticamente um ID de versão exclusivo 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 exclusivo a um objeto quando ele é adicionado a um bucket com versionamento habilitado.



Note

Os valores de ID de versão que o Amazon S3 atribui são seguros para URL (podem ser incluídos como parte de um URI).

Para obter mais informações sobre versionamento, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#). Você pode adicionar versões de objeto a um bucket habilitado para versionamento usando o console, os AWS SDKs e a API REST.

Usar o console do

Para obter instruções, consulte [Fazer upload de objetos \(p. 171\)](#).

Uso da SDKs AWS

Para ver exemplos de upload de objetos usando os AWS SDKs para Java, .NET e PHP, consulte [Fazer upload de objetos \(p. 171\)](#). Os exemplos de upload de objetos em buckets com e sem versionamento habilitado são os mesmos, embora o Amazon S3 atribua um número de versão para buckets com versionamento habilitado. Caso contrário, o número de versão é nulo.

Para obter informações sobre o uso de outros AWS SDKs, consulte o [Centro do Desenvolvedor da AWS](#).

Uso dos REST API

Para adicionar objetos a buckets com versionamento habilitado

1. Ative o versionamento de um bucket usando uma solicitação `PUT Bucket versioning`.

Para obter mais informações, consulte [PutBucketVersioning](#) na Referência da API do Amazon Simple Storage Service.

2. Envie uma solicitação `PUT`, `POST` ou `COPY` para armazenar um objeto no bucket.

Quando você adiciona um objeto a um bucket habilitado para versionamento, o Amazon S3 retorna o ID da versão do objeto no cabeçalho de resposta `x-amz-version-id`, como mostrado no exemplo a seguir.

```
x-amz-version-id: 3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY
```

Listar objetos em um bucket com versionamento habilitado

Esta seção fornece exemplos de como listar versões de objetos de um bucket com versionamento habilitado. O Amazon S3 armazena as informações de versão do objeto no sub-recurso `versões` associado ao bucket. Para obter mais informações, consulte [Opções de configuração do bucket \(p. 125\)](#). Para listar os objetos em um bucket com versionamento habilitado, a permissão `ListBucketVersions` é necessária.

Uso do console do S3

Siga estas etapas para usar o console do Amazon S3 para ver as diferentes versões de um objeto.

Para ver várias versões de um objeto

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket que contém o objeto.
3. Para ver uma lista das versões dos objetos no bucket, escolha a opção Show versions (Mostrar versões).

Para cada versão do objeto, o console exibe um ID de versão exclusivo, a data e a hora em que a versão do objeto foi criada e outras propriedades. (Os objetos armazenados em seu bucket antes da habilitação do versionamento têm um ID de versão null [nulo].)

Para listar os objetos sem as versões, escolha a opção List versions (Listar versões) .

Você também pode visualizar, fazer download e excluir versões do objeto no painel de visão geral do objeto no console. Para obter mais informações, consulte [Exibir uma visão geral do objeto no console do Amazon S3 \(p. 271\)](#).

Important

Você pode cancelar a exclusão de um objeto somente se ele foi excluído como a versão mais recente (atual). Não é possível cancelar a exclusão de uma versão anterior de um objeto que foi excluído. Para obter mais informações, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

Uso da SDKs AWS

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 o uso de outros AWS SDKs, consulte o [Centro do Desenvolvedor da AWS](#).

Java

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListVersionsRequest;
import com.amazonaws.services.s3.model.S3VersionSummary;
import com.amazonaws.services.s3.model.VersionListing;

public class ListKeysVersioningEnabledBucket {

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

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

            // Retrieve the list of versions. If the bucket contains more versions
            // than the specified maximum number of results, Amazon S3 returns
            // one page of results per request.
            ListVersionsRequest request = new ListVersionsRequest()
                .withBucketName(bucketName)
                .withMaxResults(2);
            VersionListing versionListing = s3Client.listVersions(request);
            int numVersions = 0, numPages = 0;
            while (true) {
                numPages++;
                for (S3VersionSummary objectSummary :
                    versionListing.getVersionSummaries()) {
                    System.out.printf("Retrieved object %s, version %s\n",
                        objectSummary.getKey(),
                        objectSummary.getVersionId());
                    numVersions++;
                }
                // Check whether there are more pages of versions to retrieve. If
                // there are, retrieve them. Otherwise, exit the loop.
                if (versionListing.isTruncated()) {
                    versionListing = s3Client.listNextBatchOfVersions(versionListing);
                } else {
                    break;
                }
            }
        } catch (AmazonServiceException | SdkClientException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

```
        }
    }
    System.out.println(numVersions + " object versions retrieved in " +
numPages + " pages");
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

.NET

Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

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

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

        public static void Main(string[] args)
        {
            s3Client = new AmazonS3Client(bucketRegion);
            GetObjectListWithAllVersionsAsync().Wait();
        }

        static async Task GetObjectListWithAllVersionsAsync()
        {
            try
            {
                ListVersionsRequest request = new ListVersionsRequest()
                {
                    BucketName = bucketName,
                    // You can optionally specify key name prefix in the request
                    // if you want list of object versions of a specific object.

                    // For this example we limit response to return list of 2 versions.
                    MaxKeys = 2
                };
                do
                {
                    ListVersionsResponse response = await
s3Client.ListVersionsAsync(request);
                    // Process response.
                    foreach (S3ObjectVersion entry in response.Versions)
                    {
                        Console.WriteLine("key = {0} size = {1}",
entry.Key, entry.Size);
                    }
                }
            }
        }
    }
}
```

```
        }

        // If response is truncated, set the marker to get the next
        // set of keys.
        if (response.IsTruncated)
        {
            request.KeyMarker = response.NextKeyMarker;
            request.VersionIdMarker = response.NextVersionIdMarker;
        }
        else
        {
            request = null;
        }
    } while (request != null);
}
catch (AmazonS3Exception e)
{
    Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
}
catch (Exception e)
{
    Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
}
}
}
```

Uso dos REST API

Example — Listar todas as versões de objeto em um bucket

Para listar todas as versões de todos os objetos em um bucket, você usa o sub-recurso `versions` em uma solicitação `GET Bucket`. O Amazon S3 pode recuperar no máximo 1.000 objetos, e cada versão de objeto conta como um objeto completo. 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 objetos na ordem em que foram armazenadas, com as armazenadas mais recentemente sendo retornadas primeiro.

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=
```

Example — Recuperar todas as versões de uma chave

Para recuperar um subconjunto de versões de objeto, use os parâmetros de solicitação para `GET Bucket`. Para obter mais informações, consulte [GET Bucket](#).

1. Defina o parâmetro `prefix` como a chave do objeto que você deseja recuperar.
2. Envie uma solicitação `GET Bucket` usando o sub-recurso `versions` e `prefix`.

```
GET /?versions&prefix=objectName HTTP/1.1
```

Example — Recuperar 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](#) na Referência da API do Amazon Simple Storage Service.

Example — Recuperar 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](#) na Referência da API do Amazon Simple Storage Service.

A seguir estão as respostas adicionais que atendem à solicitação GET original:

- Defina o valor de `key-marker` como a chave retornada em `NextKeyMarker` na resposta anterior.
- Defina o valor de `version-id-marker` como o ID de versão retornado em `NextVersionIdMarker` na resposta anterior.
- Envie uma solicitação `GET Bucket versions` usando `key-marker` e `version-id-marker`.

Example — Recuperar objetos que começam 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=
```

Usar a AWS CLI

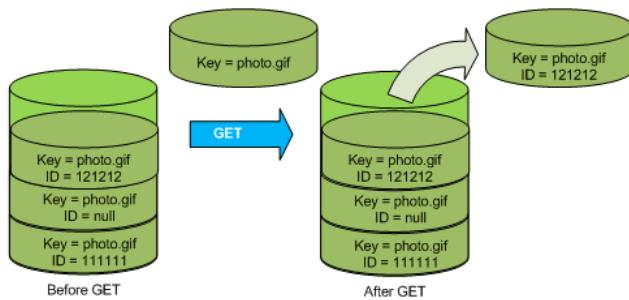
O comando a seguir retorna os metadados de todas as versões dos objetos em um bucket.

```
aws s3api list-object-versions --bucket DOC-EXAMPLE-BUCKET1
```

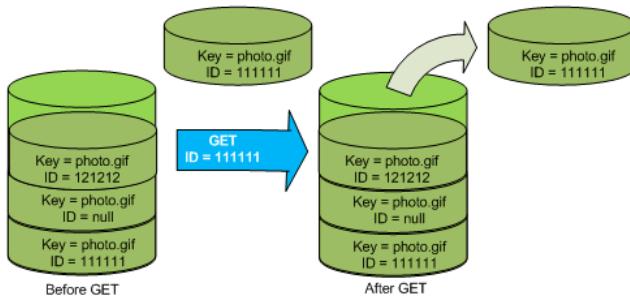
Para obter mais informações sobre `list-object-versions`, consulte [list-object-versions](#) na Referência de comandos da AWS CLI.

Recuperando versões de objeto de um bucket habilitado para versionamento

O versionamento no Amazon S3 é uma maneira de manter várias variantes de um objeto no mesmo bucket. 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).



Você pode recuperar versões de objeto no Amazon S3 usando o console, os AWS SDKs ou a API REST.

Uso do console do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket que contém o objeto.
3. Na lista Objetos , escolha o nome do objeto.
4. Escolha Versões.

O Amazon S3 mostra todas as versões do objeto.

5. Marque a caixa de seleção ao lado do ID da versão para as versões que deseja recuperar.
6. Escolha Ações, escolha Download e salve o objeto.

Você também pode visualizar, fazer download e excluir versões do objeto no painel de visão geral do objeto. Para obter mais informações, consulte [Exibir uma visão geral do objeto no console do Amazon S3 \(p. 271\)](#).

Important

Você pode cancelar a exclusão de um objeto somente se ele foi excluído como a versão mais recente (atual). Não é possível cancelar a exclusão de uma versão anterior de um objeto que foi excluído. Para obter mais informações, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

Uso da SDKs AWS

Os exemplos para carregar objetos em buckets não versionados e habilitados para versionamento são os mesmos. No entanto, para buckets habilitados para versionamento, o Amazon S3 atribui um número de versão. Caso contrário, o número de versão é nulo.

Para obter exemplos de download de objetos usando AWS SDKs for Java, .NET e PHP, consulte [Download de objetos](#).

Uso dos REST API

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 — Recuperar um objeto versionado

A seguinte solicitação recupera a versão L4kqtJlcpXroDTDmpUMLUo do `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=
```

Você pode recuperar apenas os metadados de um objeto (não o conteúdo). Para obter mais informações, consulte [the section called “Recuperação de metadados de versão” \(p. 697\)](#).

Para obter informações sobre como restaurar uma versão de objeto anterior, consulte [the section called “Restaurar versões anteriores” \(p. 698\)](#).

Recuperar 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 — Recuperar metadados de um objeto versionado

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
```

Restaurar versões anteriores

Você pode usar o versionamento para recuperar versões anteriores de um objeto. Existem duas abordagens para se fazer isso:

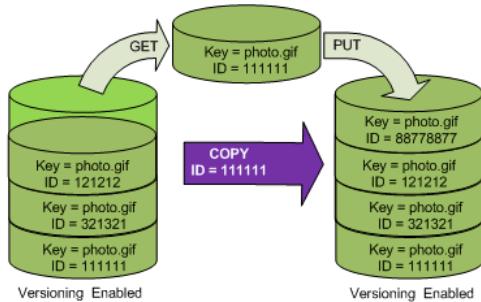
- Copie uma versão anterior do objeto para o mesmo bucket.

O objeto copiado torna-se a versão atual desse objeto e todas as versões são preservadas.

- Exclua permanentemente a versão atual do objeto.

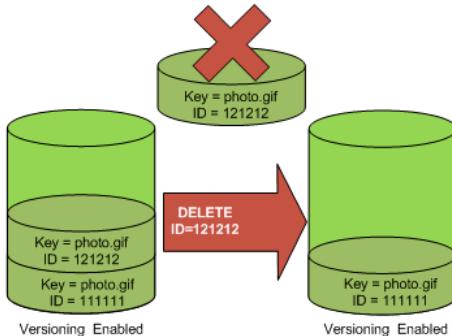
Quando você exclui a versão atual do objeto, torna a versão anterior a versão atual desse objeto.

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 para o mesmo bucket. O Amazon S3 fornece um novo ID (88778877) e ele se torna a versão atual do objeto. Assim, o bucket tem tanto a versão original (111111) quanto a cópia (88778877) do objeto. Para obter mais informações sobre como obter uma versão anterior e carregá-la para torná-la a versão atual, consulte [Recuperação de versões de objeto de um bucket habilitado para versionamento e Upload de objetos](#).



Um subsequente GET recupera a versão 88778877.

A figura a seguir mostra como excluir a versão atual (121212) de um objeto deixa a versão anterior (111111) como a atual do objeto. Para obter mais informações sobre como excluir um objeto, consulte [Excluir um único objeto](#).



Um subsequente GET recupera a versão 111111.

Para restaurar versões anteriores de objetos

Uso do console do S3

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

2. Na lista Buckets, escolha o nome do bucket que contém o objeto.
3. Na lista Objetos , escolha o nome do objeto.
4. Escolha Versões.

O Amazon S3 mostra todas as versões do objeto.

5. Marque a caixa de seleção ao lado do ID da versão para as versões que deseja recuperar.
6. Escolha Ações, escolha Downloade salve o objeto.

Você também pode visualizar, fazer download e excluir versões do objeto no painel de visão geral do objeto. Para obter mais informações, consulte [Exibir uma visão geral do objeto no console do Amazon S3 \(p. 271\)](#).

Important

Você pode cancelar a exclusão de um objeto somente se ele foi excluído como a versão mais recente (atual). Não é possível cancelar a exclusão de uma versão anterior de um objeto que foi excluído. Para obter mais informações, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

Uso da SDKs AWS

Para obter informações sobre o uso de outros AWS SDKs, consulte o [Centro do Desenvolvedor da AWS](#).

Python

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Usar a AWS SDK for Python \(Boto\) \(p. 1271\)](#).

O exemplo de código Python a seguir restaura a versão anterior de um objeto versionado excluindo todas as versões criadas após a versão de reversão especificada.

```
def rollback_object(bucket, object_key, version_id):  
    """  
    Rolls back an object to an earlier version by deleting all versions that  
    occurred after the specified rollback version.  
  
    Usage is shown in the usage_demo_single_object function at the end of this module.  
  
    :param bucket: The bucket that holds the object to roll back.  
    :param object_key: The object to roll back.  
    :param version_id: The version ID to roll back to.  
    """  
    # Versions must be sorted by last_modified date because delete markers are  
    # at the end of the list even when they are interspersed in time.  
    versions = sorted(bucket.object_versions.filter(Prefix=object_key),  
                      key=attrgetter('last_modified'), reverse=True)  
  
    logger.debug(  
        "Got versions:\n%s",  
        '\n'.join([f"\t{version.version_id}, last modified {version.last_modified}"  
                 for version in versions]))  
  
    if version_id in [ver.version_id for ver in versions]:  
        print(f"Rolling back to version {version_id}")  
        for version in versions:  
            if version.version_id != version_id:  
                version.delete()  
                print(f"Deleted version {version.version_id}")  
            else:
```

```
break

    print(f"Active version is now {bucket.Object(object_key).version_id}")
else:
    raise KeyError(f"{version_id} was not found in the list of versions for "
                   f"{object_key}.")
```

Excluir versões de objetos de um bucket com versionamento habilitado

Você pode excluir versões de objeto de buckets do Amazon S3 sempre que desejar. 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 atuais dos objetos ou que remova permanentemente as versões anteriores dos objetos. Quando seu bucket tem versionamento ativado ou suspenso, as ações da configuração do ciclo de vida funcionam do seguinte modo:

- A `Expiration` ação se aplica à versão atual do objeto. Em vez de excluir a versão atual do objeto, o Amazon S3 a retém como uma versão desatualizada adicionando 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 [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

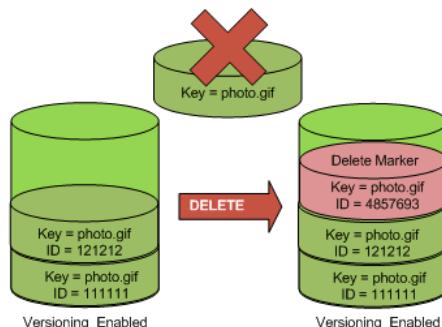
Excluir casos de uso de solicitação

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.

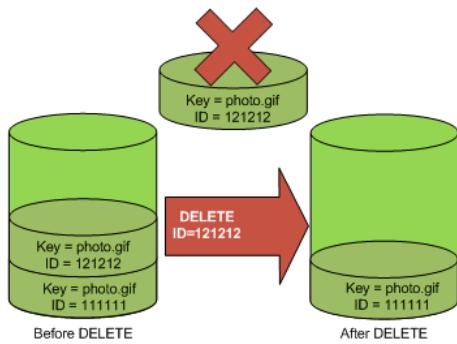
Quando você tenta um `GET` de um objeto cuja versão atual é um marcador de exclusão, o Amazon S3 se comporta como se o objeto tivesse sido excluído (mesmo que não tenha sido apagado) e retorna um erro 404. Para obter mais informações, consulte [Trabalhar com marcadores de exclusão \(p. 702\)](#).

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 objetos com versões permanentemente, você deve usar `DELETE Object versionId`.

A figura a seguir mostra que excluir uma versão do objeto especificada remove permanentemente esse objeto.



Para excluir versões de objeto

Você pode excluir versões de objeto no Amazon S3 usando o console, os AWS SDKs, a API REST ou a AWS Command Line Interface.

Uso do console do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
 2. Na lista Buckets, escolha o nome do bucket que contém o objeto.
 3. Na lista Objetos , escolha o nome do objeto.
 4. Escolha Versões.
- O Amazon S3 mostra todas as versões do objeto.
5. Marque a caixa de seleção ao lado do ID da versão para as versões que deseja recuperar permanentemente.
 6. Escolha Delete (Excluir).
 7. Em Excluir objetos permanentemente? , insira **permanently delete**.

Warning

Quando você exclui permanentemente uma versão de objeto, a ação não pode ser desfeita.

8. Escolha Delete objects (Excluir objetos).

O Amazon S3 exclui a versão do objeto.

Uso da SDKs AWS

Para ver exemplos de exclusão de objetos usando os AWS SDKs para Java, o .NET e o PHP, consulte [Excluir objetos do Amazon S3 \(p. 237\)](#). Os exemplos para excluir objetos em buckets não versionados e habilitados para versionamento são os mesmos. No entanto, para buckets habilitados para versionamento, o Amazon S3 atribui um número de versão. Caso contrário, o número de versão é nulo.

Para obter informações sobre o uso de outros AWS SDKs, consulte o [Centro do Desenvolvedor da AWS](#).

Python

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Usar a AWS SDK for Python \(Boto\) \(p. 1271\)](#).

O exemplo de código Python a seguir exclui permanente um objeto versionado ao remover todas as suas versões.

```
def permanently_delete_object(bucket, object_key):
    """
    Permanently deletes a versioned object by deleting all of its versions.

    Usage is shown in the usage_demo_single_object function at the end of this module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to delete.
    """
    try:
        bucket.object_versions.filter(Prefix=object_key).delete()
        logger.info("Permanently deleted all versions of object %s.", object_key)
    except ClientError:
        logger.exception("Couldn't delete all versions of %s.", object_key)
        raise
```

Uso dos REST API

Para excluir uma versão específica de um objeto

- Em `DELETE`, especifique o ID da versão.

Example — Excluir uma versão específica

O exemplo a seguir exclui a versão `UIORUnfnd89493jJFJ` do `photo.gif`.

```
DELETE /photo.gif?versionId=UIORUnfnd89493jJFJ HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:xQE0diMbLRepdf3YB+FIEXAMPLE=
Content-Type: text/plain
Content-Length: 0
```

Usar a AWS CLI

O comando a seguir exclui um objeto chamado `test.txt` de um bucket denominado `DOC-EXAMPLE-
BUCKET1`. Para remover uma versão específica de um objeto, você deve ser o proprietário do bucket e usar o sub-recurso de ID de versão.

```
aws s3api delete-object --bucket DOC-EXAMPLE-  
BUCKET1 --key test.txt --version-id versionID
```

Para obter mais informações sobre `delete-object`, consulte [delete-object](#) na Referência de comandos da AWS CLI.

Para obter mais informações sobre como excluir versões de objetos, consulte os seguintes tópicos:

- [Trabalhar com marcadores de exclusão \(p. 702\)](#)
- [Como remover marcadores de exclusão para tornar uma versão mais antiga atual \(p. 704\)](#)
- [Excluir um objeto de um bucket com exclusão de MFA habilitada \(p. 706\)](#)

Trabalhar com marcadores de exclusão

Um marcador de exclusão no Amazon S3 é um espaço reservado (ou marcador) para um objeto com versões que foi nomeado em uma solicitação `DELETE` simples. Como o objeto está em um bucket com

versionamento habilitado, o objeto não é excluído. Mas o marcador de exclusão faz com que o Amazon S3 se comporte como se ele 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 pode ser usada em um marcador de exclusão é uma chamada DELETE da API do Amazon S3. Para fazer isso, será necessário fazer a solicitação DELETE usando um usuário ou uma função do AWS Identity and Access Management (IAM) com as permissões adequadas.

Os marcadores de exclusão acumulam uma cobrança nominal para 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.

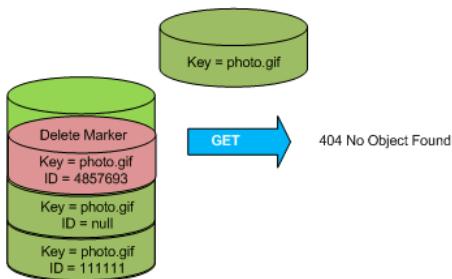
Para obter mais informações sobre nomes de chave, consulte [Criar nomes de chave de objeto \(p. 162\)](#). Para obter mais informações sobre exclusão de um marcador de exclusão, consulte [Gerenciamento de marcadores de exclusão \(p. 704\)](#).

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 habilitado 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:

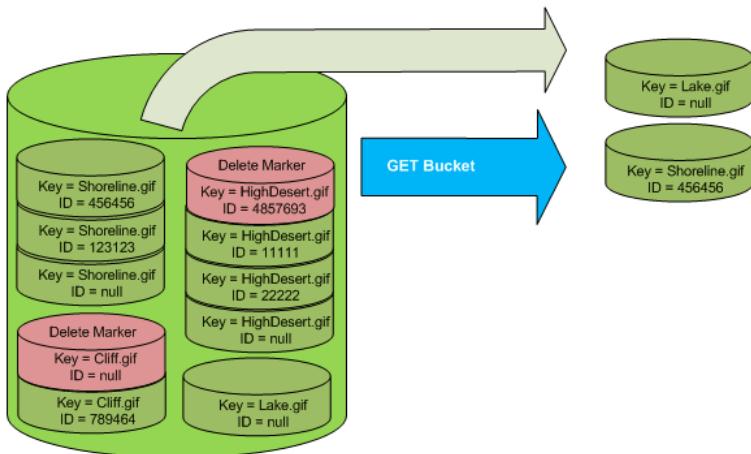
- Um erro 404 (objeto não encontrado)
- Um cabeçalho de resposta, `x-amz-delete-marker: true`

O cabeçalho de resposta mostra que o objeto acessado era um marcador de exclusão. Este cabeçalho de resposta nunca retorna `false`. Se o valor for `false`, o Amazon S3 não incluirá esse cabeçalho de resposta 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 `versions` sub-recurso em uma solicitação GET Bucket `versions`. Um GET simples não recupera objetos com marcadores de exclusão. A figura a seguir mostra que uma solicitação GET Bucket não retorna objetos cuja versão atual é um marcador de exclusão.



Gerenciamento de marcadores de exclusão

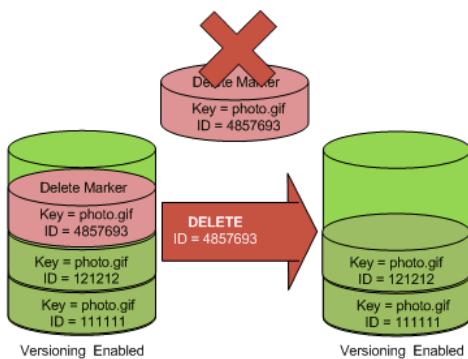
Como configurar o ciclo de vida para limpar marcadores de exclusão expirados automaticamente

Em um marcador de exclusão de objeto expirado, todas as versões de objeto são excluídas e resta apenas um único marcador de exclusão. Se a política de ciclo de vida estiver definida para excluir versões atuais ou a ação `ExpiredObjectDeleteMarker` estiver explicitamente definida, o Amazon S3 removerá o marcador de exclusão do objeto expirado. Para ver um exemplo, consulte [Exemplo 7: Remover marcadores de exclusão de objetos expirados \(p. 783\)](#).

Como remover marcadores de exclusão para tornar uma versão mais antiga atual

Quando você exclui um objeto em um bucket ativado para versionamento, todas as versões permanecem no bucket, e o Amazon S3 cria um marcador de exclusão para o objeto. Para cancelar a exclusão do objeto, você deve excluir esse marcador de exclusão. Para obter mais informações sobre versionamento e marcadores de exclusão, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

Para excluir permanentemente um marcador de exclusão, inclua seu respectivo ID de versão em uma solicitação `DeleteObject` `versionId`. A figura a seguir mostra como uma solicitação `DeleteObject` `versionId` simples remove, permanentemente, um marcador de exclusão.

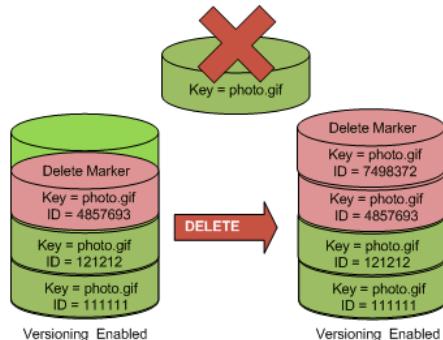


O efeito da remoção do marcador de exclusão é que uma simples solicitação `GET` não recuperará o ID da versão atual do objeto (121212).

Note

Se você usar uma solicitação `DeleteObject` em que a versão atual seja um marcador de exclusão (sem especificar o ID da versão do marcador de exclusão), o Amazon S3 não excluirá o marcador de exclusão, mas outro marcador de exclusão `PUTs`.

Para excluir um marcador de exclusão com um ID de versão `NULL`, você deve aprovar o `NULL` como o ID da versão na solicitação `DeleteObject`. A figura a seguir mostra como uma simples solicitação `DeleteObject` feita sem um ID de versão em que a versão atual é um marcador de exclusão não remove nada. Em vez disso, ela adiciona um marcador de exclusão extra com um ID de versão exclusivo (7498372).



Uso do console do S3

Use as etapas a seguir para recuperar objetos excluídos que não são pastas do seu bucket do S3, incluindo objetos que estão dentro dessas pastas.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket desejado.
3. Para ver uma lista das versões dos objetos no bucket, escolha a opção List versions (Listar versões). Você conseguirá ver os marcadores de exclusão dos objetos excluídos.
4. Para cancelar a exclusão de um objeto, você deve excluir o marcador de exclusão. Marque a caixa de seleção ao lado do delete marker (marcador de exclusão) do objeto a ser recuperado e escolha Delete (excluir).
5. Confirme a exclusão na página Delete objects (Excluir objetos).
 - a. Para a opção Permanently delete objects? (Excluir objetos permanentemente?) insira **permanently delete**.
 - b. Escolha Delete objects (Excluir objetos).

Note

Você não pode usar o console do Amazon S3 para cancelar a exclusão de pastas. Você deve usar a AWS CLI ou o SDK. Para ver exemplos, consulte [Como faço para recuperar um objeto do Amazon S3 que foi excluído em um bucket habilitado para versionamento?](#) na Central de Conhecimento da AWS.

Uso dos REST API

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 — Remover 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 o seguinte na resposta:

```
204 NoContent
x-amz-version-id: versionID
x-amz-delete-marker: true
```

Uso da SDKs AWS

Para obter informações sobre o uso de outros AWS SDKs, consulte o [Centro do Desenvolvedor da AWS](#).

Python

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Usar a AWS SDK for Python \(Boto\) \(p. 1271\)](#).

O exemplo de código Python a seguir demonstra como remover um marcador de exclusão de um objeto e, portanto, transforma a versão não atual mais recente na versão atual do objeto.

```
def revive_object(bucket, object_key):
    """
    Revives a versioned object that was deleted by removing the object's active
    delete marker.
    A versioned object presents as deleted when its latest version is a delete marker.
    By removing the delete marker, we make the previous version the latest version
    and the object then presents as *not* deleted.

    Usage is shown in the usage_demo_single_object function at the end of this module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to revive.
    """
    # Get the latest version for the object.
    response = s3.meta.client.list_object_versions(
        Bucket=bucket.name, Prefix=object_key, MaxKeys=1)

    if 'DeleteMarkers' in response:
        latest_version = response['DeleteMarkers'][0]
        if latest_version['IsLatest']:
            logger.info("Object %s was indeed deleted on %s. Let's revive it.",
                        object_key, latest_version['LastModified'])
            obj = bucket.Object(object_key)
            obj.Version(latest_version['VersionId']).delete()
            logger.info("Revived %s, active version is now %s with body '%s'",
                        object_key, obj.version_id, obj.get()['Body'].read())
        else:
            logger.warning("Delete marker is not the latest version for %s!",
                           object_key)
    elif 'Versions' in response:
        logger.warning("Got an active version for %s, nothing to do.", object_key)
    else:
        logger.error("Couldn't get any version info for %s.", object_key)
```

Excluir um objeto de um bucket com exclusão de MFA habilitada

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 [Autenticação multifator](#).

Example — Excluir um objeto de um bucket com exclusão de MFA habilitada

O exemplo a seguir exclui `my-image.jpg` (com a versão especificada), que está em um bucket configurado com a exclusão de MFA habilitada.

Observe o espaço entre `[SerialNumber]` e `[AuthenticationCode]`. Para obter mais informações, consulte [DeleteObject](#) na Referência da API do Amazon Simple Storage Service.

```
DELETE /my-image.jpg?versionId=3HL4kqCxf3vjVBH40Nrjfkd HTTPS/1.1
Host: bucketName.s3.amazonaws.com
x-amz-mfa: 20899872 301749
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Para obter mais informações sobre ativação de exclusão de MFA, consulte [Configurando a exclusão de MFA \(p. 688\)](#).

Configurando permissões de objeto com versão

As permissões para objetos no Amazon S3 são definidas no nível de versão. Cada versão tem seu próprio proprietário do objeto. A Conta da AWS que cria a versão do objeto é a proprietária. 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 você deseja definir em uma solicitação `PUT Object versionId acl`. Para uma descrição detalhada e instruções de uso de ACLs, consulte [Identity and Access Management no Amazon S3 \(p. 419\)](#).

Example — Configuração 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, `3HL4kqtJvjVBH40Nrjfkd`.

```
PUT /my-image.jpg?acl&versionId=3HL4kqtJvjVBH40Nrjfkd HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
Content-Length: 124

<AccessControlPolicy>
  <Owner>
    <ID>75cc57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>a9a7b886d6fd24a52fe8ca5bef65f89a64e0193f23000e241bf9b1c61be666e9</ID>
        <DisplayName>BucketOwner@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

```
</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 — Recuperar 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, DV BH40Nr8X8gUMLUo.

```
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 [GetObjectAcl](#) na Referência da API do Amazon Simple Storage Service.

Trabalhar com objetos em um bucket com versionamento suspenso

No Amazon S3, você pode suspender o versionamento para parar de ter novas versões do mesmo objeto em um bucket. Você pode fazer isso porque você quer apenas uma única versão de um objeto em um bucket. Ou talvez você não queira acumular cobranças para várias 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 suspenso de versão, incluindo adição, recuperação e exclusão de objetos.

Para obter mais informações sobre o S3 Versioning, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#). Para obter mais informações sobre como recuperar as versões de um objeto, consulte [Recuperando versões de objeto de um bucket habilitado para versionamento \(p. 695\)](#).

Tópicos

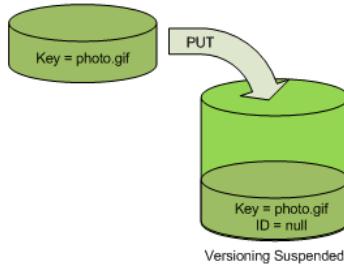
- [Adicionar objetos a buckets com versionamento suspenso \(p. 708\)](#)
- [Recuperar objetos de buckets com versionamento suspenso \(p. 709\)](#)
- [Excluir objetos de buckets com versionamento suspenso \(p. 710\)](#)

Adicionar objetos a buckets com versionamento suspenso

Você pode adicionar objetos a buckets com versionamento suspenso no Amazon S3 para criar o objeto com um ID de versão nulo ou substituir qualquer versão de objeto por um ID de versão correspondente.

Depois que você suspender o versionamento em um bucket, o Amazon S3 adicionará automaticamente um ID de versão null a cada objeto subsequente armazenado depois disso (usando PUT, POST ou COPY) nesse bucket.

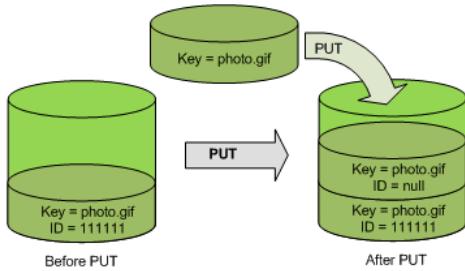
A figura a seguir mostra como o Amazon S3 adiciona um ID de versão null a cada objeto quando ele é adicionado a um bucket com versionamento suspenso.



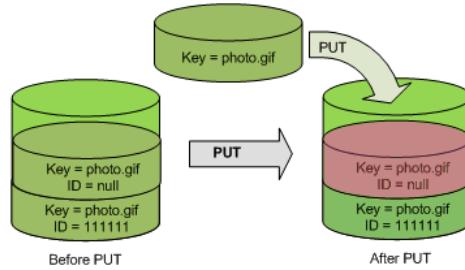
Se um versão nula já existir no bucket e você adicionar outro objeto com a mesma chave, o objeto adicionado substituirá a versão original nula.

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.

Neste caso, a versão 111111 já estava no bucket. O Amazon S3 anexa o ID de versão nula ao objeto que está sendo adicionado e armazena o objeto no bucket. A versão 111111 não é substituída.



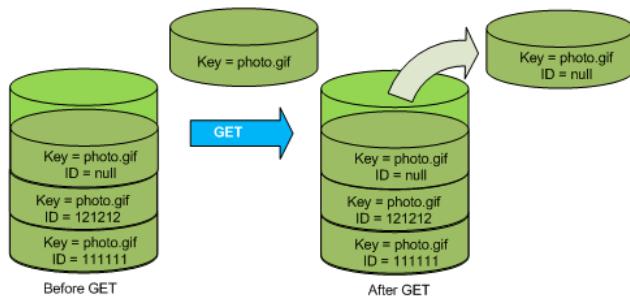
Se uma versão nula já existir em um bucket, a versão nula será substituída, como mostrado na figura a seguir.



Embora a chave e o ID (`null`) da versão nula sejam iguais antes e depois de `PUT`, o conteúdo da versão nula originalmente armazenado no bucket é substituído pelo conteúdo do objeto `PUT` no bucket.

Recuperar 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.



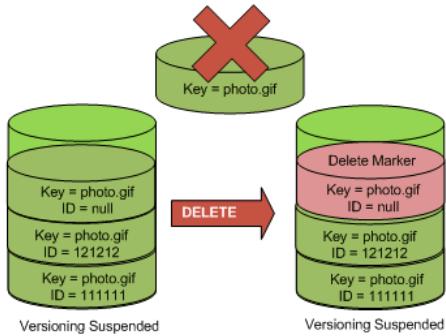
Excluir objetos de buckets com versionamento suspenso

Você pode excluir objetos de buckets com versionamento suspenso para remover um objeto com um ID de versão nulo.

Se o controle de versão for suspenso para um bucket, uma **DELETE** solicitação:

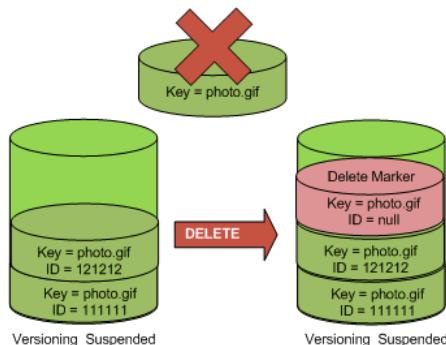
- Pode remover apenas um objeto cujo ID de versão seja **null**.
- Não removerá nada se não existir uma versão nula do objeto no bucket.
- Insere um marcador de exclusão no bucket.

A figura a seguir mostra como um simples **DELETE** remove uma versão nula. O Amazon S3 insere um marcador de exclusão em seu lugar com um ID de versão do **null**.

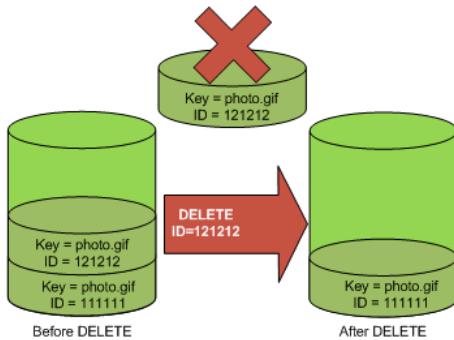


Lembre-se de que um marcador de exclusão não tem conteúdo, de modo que você perde o conteúdo da versão nula quando um marcador de exclusão a substitui.

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, incluindo o ID da versão na solicitação de `DELETE`. A figura a seguir mostra que excluir uma versão do objeto especificada remove permanentemente essa versão do objeto. Apenas o proprietário do bucket pode excluir uma versão de objeto especificada.



Usar o AWS Backup para o Amazon S3

O Amazon S3 é integrado nativamente ao AWS Backup, um serviço totalmente gerenciado e baseado em políticas que você pode usar para definir políticas de backup centralmente e proteger dados do Amazon S3. Depois de definir suas políticas de backup e atribuir recursos do Amazon S3 às políticas, o AWS Backup automatiza a criação de backups do Amazon S3 e os armazena com segurança em um cofre de backup criptografado designado pelo seu plano de backup.

Ao usar o AWS Backup para o Amazon S3, você pode executar as seguintes ações:

- Criar backups contínuos e backups periódicos. Backups contínuos são úteis para restauração em um ponto anterior no tempo, e backups periódicos são úteis para atender às suas necessidades de retenção de dados de longo prazo.
- Automatizar o agendamento e a retenção de backups configurando as políticas de backup de modo centralizado.
- Restaurar backups de dados do Amazon S3 em um ponto anterior no tempo que você especifica.

Com o AWS Backup, é possível usar o versionamento do S3 e a replicação do Amazon S3 para ajudar a se recuperar de exclusões acidentais e executar suas próprias operações de autorrecuperação.

Pré-requisitos

Você deve ativar o [Versionamento do S3](#) em seu bucket para o AWS Backup poder fazer backup.

Note

Recomendamos que você [defina uma regra de validade do ciclo de vida para buckets habilitados para versionamento](#) dos quais está sendo feito backup. Se você não definir um período de validade do ciclo de vida, os custos de armazenamento do Amazon S3 poderão aumentar, pois o AWS Backup manterá todas as versões dos dados do Amazon S3.

Começar a usar

Para começar a usar o AWS Backup para o Amazon S3, consulte [Criar backups do Amazon S3](#) no Guia do desenvolvedor do AWS Backup.

Restrições e limitações

Para saber mais sobre as limitações, consulte [Criar backups do Amazon S3](#) no Guia do desenvolvedor do AWS Backup.

Trabalhando com objetos arquivados

Quando você arquiva objetos do Amazon S3 na classe de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive ou quando os objetos são arquivados nos níveis S3 Intelligent-Tiering Archive Access ou Deep Archive Access, os objetos não ficam acessíveis em tempo real. Para restaurar os objetos, você deve fazer o seguinte:

- Para objetos nos níveis Archive Access e Deep Archive Access, você deve iniciar a solicitação de restauração e aguardar até que o objeto seja movido para o nível Frequent Access.
- Para objetos nas classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive, você deve iniciar a solicitação de restauração e aguardar até que uma cópia temporária do objeto esteja disponível.

Para obter mais informações sobre como todas as classes de armazenamento do Amazon S3 se comparam, consulte [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#).

Quando você está restaurando a partir da camada de Acesso ao Arquivamento Intelligent-Tiering do S3 ou da camada de Acesso a Arquivamento Deep Archive S3 Intelligent-Tiering, o objeto volta para a camada de Acesso Frequent por Camadas Inteligente do S3. Depois, se o objeto não for acessado após 30 dias consecutivos, ele se move automaticamente para o nível de Acesso Infrequente. Ele passa para o nível S3 Intelligent-Tiering Archive Access após um mínimo de 90 dias consecutivos sem acesso, e passa para o nível de Acesso a arquivamento profundo após um mínimo de 180 dias consecutivos sem acesso.

Note

Ao contrário das classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive, as solicitações de restauração para objetos S3 Intelligent-Tiering não aceitam o valor `days`.

Quando você usa o S3 Glacier Flexible Retrieval ou o S3 Glacier Deep Archive, o Amazon S3 restaura uma cópia temporária do objeto somente pela duração especificada. Depois disso, ele exclui a cópia restaurada do objeto. Você pode modificar o período de expiração de uma cópia restaurada reeditando uma restauração. Nesse caso, o Amazon S3 atualiza o período de expiração relativo à hora atual.

Note

Ao restaurar um arquivo do S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, você paga pelo objeto arquivado e por uma cópia restaurada temporariamente. Para obter mais informações sobre definição de preços, consulte [Definição de preços do Amazon S3](#).

O Amazon S3 calcula o tempo de validade da cópia do objeto restaurado adicionando o número de dias especificado na solicitação de restauração à hora de conclusão da restauração solicitada. Depois, ele arredonda a hora resultante para o próximo dia à meia-noite do Tempo Universal Coordenado (UTC). Por exemplo, suponha que a cópia de um objeto restaurado tenha sido criada em 15 de outubro de 2012, às 10h30 UTC e o período de restauração tenha sido especificado como três dias. Nesse caso, a cópia restaurada expira em 19 de outubro de 2012, às 00h00 UTC, hora em que o Amazon S3 vai excluir a cópia do objeto.

Se uma cópia temporária do objeto restaurado for criada, a classe de armazenamento do objeto permanecerá a mesma. (Uma solicitação de operações da API [HEAD Object](#) ou [GetObject](#) retornará S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive como a classe de armazenamento.)

O tempo necessário para concluir um trabalho de restauração depende da classe de armazenamento de arquivo ou da camada de armazenamento que você usa e qual opção de recuperação você especificar: `Expedited` (disponível apenas para o S3 Glacier Flexible Retrieval e o S3 Intelligent-Tiering Archive Access), `Standard` ou `Bulk`. Para obter mais informações, consulte [Opções de recuperação de arquivamento \(p. 713\)](#).

Você pode ser notificado quando sua restauração for concluída usando as notificações de eventos do Amazon S3. Para obter mais informações, consulte [Notificações de eventos do Amazon S3 \(p. 1083\)](#).

Quando necessário, é possível restaurar grandes segmentos dos dados armazenados para uma cópia secundária. No entanto, lembre-se de que as classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive e os níveis Archive Access e Deep Archive foram projetados para 35 solicitações de restauração aleatórias por pebibyte (PiB) armazenadas por dia.

Uso de operações em lote com solicitações de restauração

Para restaurar mais de um objeto do Amazon S3 com uma única solicitação, é possível usar operações em lote do S3. Você fornece às operações em lote do S3 uma lista de objetos nos quais operar. O S3 Batch Operations chama a respectiva API para executar a operação especificada. Um único trabalho de operações em lote pode realizar a operação especificada em bilhões de objetos contendo exabytes de dados.

O recurso S3 Batch Operations rastreia o progresso, envia notificações e armazena um relatório de conclusão detalhado de todas as ações, fornecendo uma experiência totalmente gerenciada, auditável e sem servidor. Use as operações em lote do S3 por meio do AWS Management Console, da AWS CLI, dos AWS SDKs ou da API REST. Para obter mais informações, consulte [the section called “Conceitos básicos do Batch Operations” \(p. 938\)](#).

As seções a seguir fornecem mais informações sobre a restauração de objetos arquivados.

Tópicos

- [Opções de recuperação de arquivamento \(p. 713\)](#)
- [Restaurar um objeto arquivado \(p. 715\)](#)
- [Consultar objetos arquivados \(p. 719\)](#)

Opções de recuperação de arquivamento

Veja a seguir as opções de recuperação disponíveis ao restaurar um objeto arquivado no Amazon S3:

- **Expedited**: acesse rapidamente seus dados armazenados na classe de armazenamento S3 Glacier Flexible Retrieval ou no nível S3 Intelligent-Tiering Archive Access quando forem necessárias solicitações urgentes ocasionais para um subconjunto de arquivos. Para todos os objetos arquivados, exceto os maiores (mais de 250 MB), os dados acessados por meio de recuperações expressas geralmente são disponibilizados dentro de um a cinco minutos.

A capacidade provisionada ajuda a garantir que a capacidade de recuperação para recuperações expressas do S3 Glacier Flexible Retrieval esteja disponível quando você precisar dela. Para obter mais informações, consulte [Capacidade provisionada \(p. 714\)](#).

- **Standard** - permite acessar qualquer um dos objetos arquivados em algumas horas. Esta é a opção padrão para solicitações de recuperação que não especificam a opção de recuperação. As recuperações padrão geralmente terminam em até 3 a 5 horas para objetos armazenados na classe de armazenamento S3 Glacier Flexible Retrieval ou no nível Archive Access do S3 Intelligent-Tiering. Normalmente, eles terminam dentro de 12 horas para objetos armazenados na classe de armazenamento S3 Glacier Deep Archive ou S3 Intelligent-Tiering Deep Archive Access. As recuperações padrão são gratuitas para objetos armazenados no S3 Intelligent-Tiering.
- **Bulk** - as recuperações em massa são a opção de recuperação de menor custo do Amazon S3 Glacier, permitindo recuperar grandes quantidades de dados, até mesmo petabytes, com um custo baixo. As recuperações em massa geralmente terminam em até 5 a 12 horas para objetos armazenados na classe de armazenamento S3 Glacier Flexible Retrieval ou no nível Archive Access do S3 Intelligent-Tiering. Normalmente, eles terminam dentro de 48 horas para objetos armazenados na classe de armazenamento S3 Glacier Deep Archive ou no nível S3 Intelligent-Tiering Deep Archive Access. As

recuperações em massa são gratuitas para objetos armazenados no S3 Glacier Flexible Retrieval e S3 Intelligent-Tiering.

A tabela a seguir resume as opções de recuperação de arquivos. Para obter informações completas sobre a definição de preço, consulte [Definição de preço do Amazon S3](#).

Para fazer uma recuperação de `Expedited`, `Standard` ou `Bulk`, defina o elemento `Tier` da solicitação na solicitação de API REST `POST Object restore` 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 um objeto arquivado de forma programática ou usando o console do Amazon S3. O Amazon S3 só processa uma solicitação de restauração por vez por objeto. É possível 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.

Para obter mais informações, consulte [Restaurar um objeto arquivado \(p. 715\)](#).

Capacidade provisionada

A capacidade provisionada ajuda a garantir que a capacidade de recuperação para recuperações aceleradas do S3 Glacier Flexible Retrieval esteja disponível quando você precisar. 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.

Se sua workload exigir acesso altamente confiável e previsível a um subconjunto de seus dados em minutos, será necessário adquirir a capacidade de recuperação provisionada. Sem capacidade provisionada, as recuperações expressas podem não ser aceitas durante períodos de alta demanda. Se precisar de acesso a recuperações expressas em qualquer circunstância, recomendamos que você compre a capacidade de recuperação provisionada.

É possível comprar a capacidade provisionada usando o console do Amazon S3, o console do Amazon S3 Glacier, a API REST [Purchase Provisioned Capacity](#), os AWS SDKs ou a AWS CLI. Para obter informações sobre a definição de preços da capacidade provisionada, consulte [Definição de preço do Amazon S3](#).

Atualizar a velocidade de uma restauração em andamento

Usando a atualização rápida de restauração do Amazon S3, é possível 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 que define um novo elemento de solicitação de `Tier` na API REST da [restauração de objeto POST](#) ou o equivalente na AWS CLI ou nos AWS SDKs. 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`.

Note

As restaurações padrão e a granel para o S3 Intelligent-Tiering são gratuitas. No entanto, as solicitações de restauração subsequentes chamadas em um objeto que já está sendo restaurado são cobradas como uma solicitação GET.

Você pode ser notificado quando sua restauração for concluída usando as notificações de eventos do Amazon S3. Para obter mais informações, consulte [Notificações de eventos do Amazon S3 \(p. 1083\)](#). As restaurações são cobradas pelo preço do nível atualizado. Para obter informações sobre preços de restauração, consulte [Preços do Amazon S3](#)

Restaurar um objeto arquivado

Os objetos do Amazon S3 armazenados nas classes de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive não são acessíveis imediatamente. Para acessar um objeto nessas classes de armazenamento, você deve restaurar uma cópia temporária dele no bucket do S3 por uma duração especificada (número de dias). Para obter informações sobre como usar essas classes de armazenamento, consulte [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#) e [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

Objetos restaurados do S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive são armazenados apenas pelo número de dias que você especificar. Para obter uma cópia permanente do objeto, crie uma cópia dele no bucket do Amazon S3. A menos que você faça uma cópia, o objeto ainda será armazenado nas classes de armazenamento do S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive.

Para calcular a data de expiração, o Amazon S3 adiciona o número de dias que você especifica à hora em que você solicita a restauração do objeto e, em seguida, arredonda para o dia seguinte à meia-noite UTC. Este cálculo se aplica à restauração inicial do objeto e a qualquer extensão na disponibilidade que você solicitar. Por exemplo, se um objeto foi restaurado em 15 de outubro de 2012 às 10h30 UTC e você especificou o número de dias como **3**, o objeto estará disponível até 19 de outubro de 2012 às 00h00 UTC. Se em 16 de outubro de 2012 às 11h UTC você alterar o número de dias que deseja que ele esteja acessível para **1**, o Amazon S3 manterá o objeto restaurado disponível até 18 de outubro de 2012 às 00h00 UTC.

Ao restaurar um objeto arquivado, você está pagando pelo arquivo e por uma cópia restaurada temporariamente. Para obter mais informações sobre definição de preços, consulte [Definição de preços do Amazon S3](#).

Você pode restaurar um objeto arquivado usando o console do Amazon S3, a API REST, os AWS SDKs e a AWS Command Line Interface (AWS CLI).

Uso do console do S3

Use as etapas a seguir para restaurar um objeto que foi arquivado nas classes de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, para verificar o status e atualizar uma restauração em andamento. (O console usa os nomes S3 Glacier Flexible Retrieval e Glacier Deep Archive para essas classes de armazenamento.)

Para restaurar um objeto arquivado

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket que contém os objetos que você deseja restaurar.
3. Na lista Objects (Objetos), selecione o objeto ou os objetos que você deseja restaurar, selecione Actions (Ações) e escolha Initiate restore (Iniciar restauração).
4. Se você estiver restaurando do S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, insira o número de dias em que deseja que seus dados arquivados sejam acessíveis na caixa de diálogo Initiate restore (Iniciar restauração).
5. Em Opções de recuperação, siga um destes procedimentos:
 - Escolha Bulk retrieval (Recuperação em massa) ou Standard retrieval (Recuperação padrão), em seguida, e escolha Restore (Restaurar).
 - Escolha Expedited retrieval (Recuperação acelerada) (disponível somente para o S3 Glacier Flexible Retrieval ou S3 Intelligent-Tiering Archive Access).
6. A capacidade provisionada só está disponível para objetos no S3 Glacier Flexible Retrieval. Se você tiver a capacidade provisionada, escolha Restore (Restaurar) para iniciar uma recuperação provisionada.

Se você tiver a capacidade provisionada, todas as recuperações expressas serão atendidas pela capacidade provisionada. Para obter mais informações, consulte [Capacidade provisionada \(p. 714\)](#).

- Se você não tiver a capacidade provisionada e não desejar comprá-la, escolha Restore (Restaurar).
- Se você não tiver a capacidade provisionada, mas quiser comprá-la, escolha Add capacity unit (Adicionar unidade de capacidade) e, em seguida, escolha Buy (Comprar). Quando receber a mensagem Purchase succeeded (Compra bem-sucedida), escolha Restore (Restaurar) para iniciar a recuperação provisionada.

Atualize a velocidade da restauração enquanto ela está em andamento.

Para atualizar uma restauração em andamento para um nível mais rápido

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Bucket name (Nome do bucket), escolha o nome do bucket que contém os objetos que você deseja restaurar.
3. Na lista Objects (Objetos), selecione um ou mais dos objetos que você está restaurando, escolha Actions (Ações) e Restore from Glacier Flexible Retrieval (Restaurar do S3 Glacier Flexible Retrieval). Para obter informações sobre como verificar o status de restauração de um objeto, consulte [Verificar o status de restauração e a data de expiração \(p. 716\)](#).
4. Escolha o nível para o qual você deseja atualizar e Restore (Restaurar).

Para obter informações sobre como fazer upgrade para um nível de restauração mais rápido, consulte [Atualizar a velocidade de uma restauração em andamento \(p. 714\)](#).

Note

As restaurações padrão e a granel para o S3 Intelligent-Tiering são gratuitas. No entanto, as solicitações de restauração subsequentes chamadas em um objeto que já está sendo restaurado são cobradas como uma solicitação GET.

[Verificar o status de restauração e a data de expiração](#)

Você pode verificar o andamento da restauração na página Object overview (Visão geral do objeto). Para obter mais informações, consulte [Exibir uma visão geral do objeto no console do Amazon S3 \(p. 271\)](#). Esta página mostrará que a restauração está In progress (Em andamento).

Se você estiver restaurando do S3 Glacier Flexible Retrieval ou do S3 Glacier Deep Archive, a cópia temporária de Object overview (Visão geral do objeto) exibirá a Restoration expiry date (Data de validade da restauração). O Amazon S3 removerá a cópia restaurada do arquivamento nessa data.

Objetos restaurados do S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive são armazenados apenas pelo número de dias que você especificar. Para obter uma cópia permanente do objeto, crie uma cópia dele no bucket do Amazon S3.

Depois de restaurar um objeto, você pode baixá-lo na página Overview (Visão geral). Para obter mais informações, consulte [Exibir uma visão geral do objeto no console do Amazon S3 \(p. 271\)](#).

Uso da SDKs AWS

Java

No exemplo a seguir, é restaurada uma cópia de um objeto que foi arquivado 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 instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

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

import java.io.IOException;

public class RestoreArchivedObject {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_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();
        }
    }
}
```

.NET

O exemplo de C# a seguir 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 como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
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");
        }
    }
}
```

Uso dos REST API

O Amazon S3 fornece uma API para que você inicie uma restauração de arquivos. Para obter mais informações, consulte [RestoreObject](#) na Referência da API do Amazon Simple Storage Service.

Uso do AWS CLI

Use o comando `restore-object` para restaurar objetos do S3 Glacier Flexible Retrieval.

O exemplo a seguir restaura o objeto `dir1/example.obj` em `awsexamplebucket` por 25 dias.

```
aws s3api restore-object --bucket awsexamplebucket --key dir1/example.obj --restore-request '{"Days":25,"GlacierJobParameters":{"Tier":"Standard"}}'
```

Se a sintaxe JSON usada no exemplo resultar em um erro em um cliente Windows, substitua a solicitação de restauração pela seguinte sintaxe:

```
--restore-request Days=25,GlacierJobParameters={"Tier"]="Standard"}
```

Você pode usar o seguinte comando para monitorar o status de sua solicitação `restore-object`:

```
aws s3api head-object --bucket awsexamplebucket --key dir1/example.obj
```

Para obter mais informações, consulte [restore-object](#) na Referência de comandos da AWS CLI.

Consultar objetos arquivados

Com o tipo de seleção [Restauração do objeto POST](#), execute as operações de filtro usando declarações simples em linguagem de consulta estruturada (SQL) diretamente em seus dados que arquivados pelo Amazon S3 no S3 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. É possível executar consultas e análises personalizadas dos dados armazenados no S3 Glacier, sem precisar restaurar o objeto todo para o Amazon S3.

Ao executar consultas de seleção, o S3 Glacier oferece três níveis de acesso aos dados: expresso, padrão e em massa. Todos esses níveis fornecem diferentes tempos de acesso a dados e custos, e você pode escolher qualquer um deles dependendo da rapidez com que deseja que seus dados sejam disponibilizados. Para obter mais informações, consulte [Níveis de acesso aos dados \(p. 722\)](#).

Você pode usar o tipo de seleção de restauração com os AWS SDKs, a API REST do S3 Glacier e a AWS Command Line Interface (AWS CLI).

Tópicos

- [Requisitos e limites ao usar select \(p. 719\)](#)
- [Consulta de dados usando select \(p. 720\)](#)
- [Tratamento de erros \(p. 721\)](#)
- [Níveis de acesso aos dados \(p. 722\)](#)

Requisitos e limites ao usar select

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.

- Você precisa de um bucket do S3 para saída. A Conta da AWS usada para iniciar um trabalho de seleção do S3 Glacier deve ter permissões para gravação no bucket do S3. O bucket deve estar na mesma Região da AWS que o 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 [Exemplo: operações de sub-recursos de bucket \(p. 443\)](#).
- 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.

Consulta de dados usando select

Com a seleção, é possível usar comandos SQL para consultar objetos de arquivo do S3 Glacier que estão criptografados em formato CSV descompactado. Com essa restrição, é possível executar operações de consulta simples nos dados baseados em textos no S3 Glacier. Por exemplo, você pode procurar um ID ou um nome específico em um conjunto de arquivos de texto.

Para consultar os dados do S3 Glacier, crie uma solicitação de seleção usando a operação [Restauração do objeto POST](#). 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 [Restauração do objeto POST](#).

```
SELECT * FROM object
```

O S3 Glacier Select é compatível com 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 para SQL, consulte [Referência SQL para Amazon S3 Select e S3 Glacier Select. \(p. 912\)](#).

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 S3; na mesma Região da AWS que o 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.

Você pode especificar a classe de armazenamento e a criptografia do Amazon S3 nos objetos de saída armazenados nele. Selecione o suporte para criptografia do AWS Key Management Service (SSE-KMS) e Amazon S3 (SSE-S3). Ela não oferece suporte às criptografias SSE-C e no lado do cliente. Para obter mais informações sobre classes de armazenamento e criptografias do Amazon S3, consulte [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#) e [Proteção de dados usando criptografia no lado do servidor \(p. 359\)](#).

Os resultados do S3 Glacier Select são armazenados no bucket do S3 usando o prefixo fornecido no local de saída especificado em [Restauração do objeto POST](#). 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 comum.) Nesse prefixo exclusivo, há dois novos prefixos criados, `results` para resultados e `errors` para logs e erros. Quando o trabalho é concluído, um manifesto de resultado é gravado 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, é possível 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 ao modo de ordenação dos resultados.

Note

O tamanho de um nome de objeto do Amazon S3 também conhecido como chave, não pode ter mais do que 1.024 bytes. O S3 Glacier Select reserva 128 bytes para prefixos. Além disso, o tamanho do caminho do local do Amazon S3 não pode ser maior do que 512 bytes. Uma solicitação superior a 512 bytes retornará uma exceção, e a solicitação não será aceita.

Tratamento de 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 [Restauração do objeto POST](#). 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 Select interrompe a execução da consulta após a detecção de um erro. Para executar a consulta com êxito, é necessário 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.

Níveis 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 \(p. 722\)](#).
- **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 S3 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 SDKs da AWS. 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 por nível, consulte [Definição de preços do S3 Glacier](#).

Capacidade provisionada

A capacidade provisionada ajuda a garantir que sua capacidade de recuperação para recuperações expressas esteja disponível quando você precisar dela. Cada unidade de capacidade 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. Para obter mais informações, consulte [the section called "Capacidade provisionada" \(p. 714\)](#).

Usar o bloqueio de objetos do S3

Com o bloqueio de objetos do S3, é possível armazenar objetos usando um modelo Gravar uma vez, ler muitas (WORM, write-once-read-many). O bloqueio de objetos pode ajudar a evitar que os objetos sejam excluídos ou substituídos por um período de tempo fixo ou indefinidamente. Você pode usar o bloqueio de objetos para ajudar a atender aos requisitos regulamentares que exigem armazenamento WORM ou simplesmente adicionar outra camada de proteção contra alterações e exclusão de objetos.

O bloqueio de objetos do S3 foi avaliado pela Cohasset Associates para uso em ambientes sujeitos aos regulamentos SEC 17a-4, CFTC e FINRA. Para obter mais informações sobre como o bloqueio de objetos está relacionado a essas regulamentações, consulte o [Cohasset Associates Compliance Assessment](#).

O bloqueio de objetos fornece duas maneiras de gerenciar a retenção de objetos: períodos de retenção e retenções legais.

- 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. Para obter mais informações, consulte [Períodos de retenção \(p. 724\)](#)
- 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 independentem dos períodos de retenção. Para obter mais informações, consulte [Retenções legais \(p. 725\)](#).

Uma versão do objeto pode ter um período de retenção e uma retenção legal, um, mas não o outro ou nenhum dos dois. Para obter mais informações, consulte [Como o bloqueio de objetos do S3 funciona \(p. 723\)](#).

O bloqueio de objetos só funciona em buckets com versionamento, e os períodos de retenção e as retenções legais se aplicam a versões de objetos individuais. Quando você bloqueia uma versão do objeto, o Amazon S3 armazena as informações de 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. Isso não impede 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, armazenará ela no bucket conforme solicitado e relatará a solicitação como concluída com êxito. A versão protegida existente do objeto permanece bloqueada de acordo com a configuração da retenção.

Para usar o bloqueio de objetos do S3, siga estas etapas básicas:

1. Crie um novo bucket com o bloqueio de objetos 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.

Para obter informações sobre como configurar e gerenciar o bloqueio de objetos do S3, consulte as seguintes seções:

Tópicos

- [Como o bloqueio de objetos do S3 funciona \(p. 723\)](#)
- [Configurar o bloqueio de objetos do S3 usando o console \(p. 727\)](#)
- [Gerenciar o bloqueio de objetos \(p. 728\)](#)

Como o bloqueio de objetos do S3 funciona

Você pode usar o bloqueio de objetos do S3 para armazenar objetos usando um modelo gravar uma vez, ler muitas (WORM, write once read many). O bloqueio de objetos pode ajudar a evitar que os objetos sejam excluídos ou substituídos por um período de tempo fixo ou indefinidamente. Você pode usar o bloqueio de objetos do S3 para atender a requisitos regulamentares que exigem armazenamento WORM ou adicionar uma camada extra de proteção contra alterações e exclusões de objetos.

Para obter informações sobre como gerenciar o status de bloqueio de objetos do Amazon S3, consulte [the section called “Gerenciar o bloqueio de objetos” \(p. 728\)](#).

Note

Os buckets do S3 com bloqueio de objeto do S3 não podem ser usados como buckets de destino para logs de acesso ao servidor. Para obter mais informações, consulte [the section called “Registrando acesso ao servidor” \(p. 1041\)](#).

As seções a seguir descrevem os principais recursos do bloqueio de objetos do S3.

Tópicos

- [Modos de retenção \(p. 724\)](#)
- [Períodos de retenção \(p. 724\)](#)
- [Retenções legais \(p. 725\)](#)
- [Configuração do bucket \(p. 726\)](#)
- [Permissões obrigatórias \(p. 727\)](#)

Modos de retenção

O bloqueio de objetos do S3 fornece dois modos de retenção:

- Modo de governança
- Modo de conformidade

Esses modos de retenção aplicam níveis diferentes de proteção aos objetos. Aplique um dos modos de retenção a qualquer versão de objeto protegida pelo bloqueio de objetos.

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. Com o modo de governança, você protege objetos contra a exclusão da maioria dos usuários, mas ainda pode 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 no modo de conformidade.

Para substituir ou remover as configurações de retenção do modo de governança, um usuário deve ter a permissão `s3:BypassGovernanceRetention` 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.

Note

Por padrão, o console do Amazon S3 inclui o cabeçalho `x-amz-bypass-governance-retention:true`. Se você tentar excluir objetos protegidos pelo modo governança e tiver permissões `s3:BypassGovernanceRetention`, a operação terá êxito.

No modo de conformidade, uma versão do objeto protegida não pode ser substituída nem excluída por qualquer usuário, inclusive o usuário root na Conta da AWS. Quando 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 ajuda a garantir que uma versão do objeto 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 timestamp `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 timestamp 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.

É possível colocar um período de retenção em uma versão do objeto explicitamente ou por meio de uma configuração padrão do bucket. Ao aplicar um período de retenção a uma versão de objeto explicitamente,

especifique a opção de *Retain Until Date* (Reter até uma determinada data) para a versão do objeto. O Amazon S3 armazena a configuração *Retain Until Date* nos metadados da versão do objeto e protege a versão do objeto até que o período de retenção expire.

Ao usar as configurações padrão do bucket, você não especifica uma *Retain Until Date*. 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 *Retain Until Date* para a versão do objeto adicionando a duração especificada ao timestamp da criação da versão do objeto. Ele armazena a *Retain Until Date* 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 de bloqueio de objetos, os períodos de retenção se aplicam a versões de objetos individuais. As versões diferentes de um único objeto podem ter modos e períodos de retenção diferentes.

Por exemplo, suponha que você tenha um objeto de 15 dias em um período de retenção de 30 dias e você `PUT` um objeto no Amazon S3 com o mesmo nome e um período de retenção de 60 dias. Nesse caso, o `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 excluível em 15 dias.

Prolongue um período de retenção depois que você tiver aplicado uma configuração de retenção a uma versão do objeto. Para fazer isso, envie uma nova solicitação de bloqueio para a versão do objeto com uma *Retain Until Date* que seja posterior à configurada atualmente para a versão do objeto. O Amazon S3 substitui o período de retenção existente pelo novo período 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

Com o bloqueio de objetos, você também pode colocar uma retenção legal em uma versão de 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`. Para obter uma lista completa das permissões do Amazon S3, consulte [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#).

As retenções legais independem dos períodos de retenção. Desde que o bucket que contém o objeto tenha o bloqueio de objetos habilitado, é possível colocar e remover retenções legais, independentemente da versão do objeto especificado 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, suponha que você coloque uma retenção legal em uma versão do objeto enquanto essa versão também está protegida por um período de retenção. Se 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.

Para usar o bloqueio de objetos, habilite-o para um bucket. Também é possível configurar um modo e um período de retenção padrão que se aplicam a novos objetos colocados no bucket. Para obter mais informações, consulte [Configurar o bloqueio de objetos do S3 usando o console \(p. 727\)](#).

Configuração do bucket

Para usar o bloqueio de objetos, habilite-o para um bucket. Também é possível configurar um modo e um período de retenção padrão que se aplicam a novos objetos colocados no bucket.

Note

Ao usar o bloqueio de objetos do S3, certifique-se de levar sua técnica de criptografia em consideração. Por exemplo, se você estiver usando criptografia no lado do servidor com as chaves do AWS KMS, considere como a possível exclusão da chave pode interagir com o bloqueio de objetos do S3. Pode ser importante considerar a proteção para a chave também.

Habilitar o bloqueio de objetos do S3

Para bloquear todos os objetos, você precisa configurar um bucket para usar o bloqueio de objetos do S3. Para fazer isso, especifique que deseja habilitar o bloqueio de objetos ao criar o bucket. Depois de configurar um bucket para bloqueio de objetos, você poderá bloquear objetos nesse bucket com períodos de retenção, retenções legais ou ambos.

Note

- Só é possível habilitar o bloqueio de objetos para novos buckets. Para habilitar o bloqueio de objetos para um bucket existente, entre em contato com o Suppor AWS Support.
- Ao criar um bucket com o bloqueio de objetos habilitado, o Amazon S3 habilita automaticamente o versionamento para o bucket.
- Se você criar um bucket com o bloqueio de objetos habilitado, não poderá desabilitá-lo nem suspender o versionamento para o bucket.

Para obter informações sobre como ativar o Object Lock no console, consulte [Configurar o bloqueio de objetos do S3 usando o console \(p. 727\)](#).

Configurações de retenção padrão

Ao ativar o bloqueio de objetos para um bucket, o bucket pode armazenar objetos protegidos. No entanto, a configuração não protege automaticamente os 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

Para impor o modo de retenção padrão do bucket e o período para todas as novas versões de objetos colocados em um bucket, defina os padrões do bucket e negue permissão aos usuários para configurarem as configurações de retenção de objetos. O Amazon S3 aplica o modo e o período de retenção padrão a novas versões de objeto colocados no bucket e rejeita qualquer solicitação para colocar um objeto que inclua um modo de retenção e configuração.

As configurações padrão do bucket exigem um modo e um período. Um modo padrão do bucket é de governança ou de conformidade. Para obter mais informações, consulte [Modos de retenção \(p. 724\)](#).

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 padrão, o bloqueio de objetos calcula uma retenção até uma determinada data. Ele faz isso adicionando o período de retenção padrão ao timestamp de criação da versão do objeto. O Amazon S3 armazena o timestamp resultante como a opção de `Retain Until Date`, como se você mesmo tivesse calculado o timestamp manualmente e colocado-o na versão do objeto.

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 criará 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 altera o período de retenção existente para nenhum objeto nesse bucket.

Note

Se você configurar um período de retenção padrão em um bucket, as solicitações para fazer upload de objetos nesse bucket devem incluir o cabeçalho Content-MD5. Para obter mais informações, consulte [PutObject](#) na Referência da API do Amazon Simple Storage Service.

Permissões obrigatórias

As operações do bloqueio de objetos exigem permissões específicas. Dependendo da operação exata que você está tentando realizar, talvez algumas das seguintes permissões poderão ser necessárias:

- `s3:BypassGovernanceRetention`
- `s3:GetBucketObjectLockConfiguration`
- `s3:GetObjectLegalHold`
- `s3:GetObjectRetention`
- `s3:PutBucketObjectLockConfiguration`
- `s3:PutObjectLegalHold`
- `s3:PutObjectRetention`

Para obter informações sobre como usar condições com permissões, consulte [Exemplos de chave de condição do Amazon S3 \(p. 447\)](#).

Configurar o bloqueio de objetos do S3 usando o console

Com o bloqueio de objetos do S3, é possível armazenar objetos no Amazon S3 usando um modelo write-once-read-many (WORM). Você pode usar o bloqueio de objetos do S3 para evitar que um objeto seja excluído ou substituído por um período fixo ou indefinidamente. Para obter mais informações sobre os recursos do bloqueio de objetos do S3, consulte [Como o bloqueio de objetos do S3 funciona \(p. 723\)](#).

Para usar o bloqueio de objetos do S3, siga estas etapas básicas:

1. Crie um novo bucket com o bloqueio de objetos 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.

Antes de bloquear qualquer objeto, você precisa habilitar um bucket para usar o bloqueio de objetos do S3. Você habilita o bloqueio de objetos ao criar um bucket. Depois de habilitar um bloqueio de objeto do

em um bucket, você poderá bloquear objetos nesse bucket. Assim que criar um bucket com bloqueio de objetos habilitado, você não poderá desabilitá-lo nem suspender o versionamento para o bucket.

Para obter informações sobre como criar um bucket com o S3 Object Lock habilitado, consulte [Criação de um bucket \(p. 129\)](#).

Para habilitar a retenção legal do Object Lock

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket desejado.
3. Na lista Objects (Objetos), escolha o nome do objeto para o qual você deseja ativar ou desativar a retenção legal.

A Object overview (Visão geral do objeto) será exibida, mostrando as propriedades do objeto.

4. Em Object Lock legal hold (Retenção legal do Object Lock), escolha Edit (Editar).
5. Em Legal hold (Retenção legal), escolha Enable (Habilitar) ou Disable (Desabilitar).
6. Selecione Save changes.

Para editar as configurações de retenção do Object Lock

1. Na lista Objects (Objetos), escolha o nome do objeto para o qual você deseja editar as configurações de retenção do Object Lock.

A Object overview (Visão geral do objeto) será exibida, mostrando as propriedades do objeto.

2. Em Object Lock retention (Retenção do Object Lock), escolha Edit (Editar).
3. Em Retention (Retenção), escolha Enable (Habilitar) ou Disable (Desabilitar).
4. Em Retention mode (Modo de retenção), escolha Governance mode (Modo de governança) ou Compliance mode (Modo de conformidade).
5. Na caixa Retain until date (Reter até a data), insira a data em que o objeto não estará mais protegido pelo modo de retenção escolhido.
6. Selecione Save changes.

Para obter mais informações sobre as configurações de retenção e retenção legal, consulte [Como o bloqueio de objetos do S3 funciona \(p. 723\)](#).

Para obter informações sobre como gerenciar o bloqueio de objetos usando a AWS CLI, os AWS SDKs e as APIs REST do Amazon S3, consulte [\(p. 728\)](#).

Gerenciar o bloqueio de objetos

Você pode usar a AWS CLI, AWS SDKs e as APIs REST do Amazon S3 para configurar e visualizar informações de bloqueio, definir limites de retenção, gerenciar exclusões e ciclos de vida e muito mais.

Tópicos

- [Visualizar as informações de bloqueio de um objeto \(p. 729\)](#)
- [Ignorar modo de governança \(p. 729\)](#)
- [Configurar eventos e notificações \(p. 729\)](#)
- [Definir limites de retenção \(p. 729\)](#)
- [Gerenciar marcadores de exclusão e ciclos de vida de objetos \(p. 730\)](#)
- [Usar o bloqueio de objetos do S3 com replicação \(p. 730\)](#)

Visualizar as informações de bloqueio de um objeto

É possível visualizar o status do bloqueio de objetos de uma versão de objeto do Amazon S3 usando o comando `GET Object` ou `HEAD Object`. Ambos os comandos retornam o modo de retenção, `Retain Until Date`, 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 `GET` ou `HEAD` uma versão do objeto, mas não tiver as permissões necessárias para ver o status de bloqueio, a solicitação será bem-sucedida. No entanto, as informações que você não tem permissão para ver não são retornadas.

Para visualizar a configuração de retenção padrão do bucket (caso exista), solicite a configuração do bloqueio de objetos do bucket. Para fazer isso, você deve ter a permissão `s3:GetBucketObjectLockConfiguration`. Caso você faça uma solicitação de uma configuração de bloqueio de objetos em um bucket que não tenha o bloqueio de objetos do S3 habilitado, o Amazon S3 retornará um erro. Para obter mais informações sobre permissões, consulte [Exemplo: operações de objeto \(p. 442\)](#).

Você pode configurar os relatórios do Amazon S3 Inventory nos buckets para incluir as opções `Retain Until Date`, `object lock Mode` e `Legal Hold Status` para todos os objetos em um bucket. Para obter mais informações, consulte [Inventário do Amazon S3 \(p. 786\)](#).

Ignorar modo de governança

Você pode realizar operações em versões de objeto bloqueadas em modo de governança como se elas estivessem desprotegidas, caso tenha a permissão `s3:BypassGovernanceRetention`. Essas operações incluem 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 esse modo. Para fazer isso, inclua o cabeçalho `x-amz-bypass-governance-retention:true` com a sua solicitação ou use o parâmetro equivalente com solicitações feitas por meio da AWS CLI ou dos AWS SDKs. O AWS Management Console aplicará 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.

Configurar eventos e notificações

Use notificações de eventos do Amazon S3 para rastrear o acesso e as alterações feitas nas configurações do bloqueio de objetos e nos dados que usam o AWS CloudTrail. Para obter informações sobre o CloudTrail, consulte a [documentação do AWS CloudTrail](#).

Você também pode usar o Amazon CloudWatch para gerar alertas com base nesses dados. Para obter mais informações sobre o CloudWatch, consulte a [Documentação do Amazon 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 período máximo de retenção é de cem anos.

O exemplo a seguir mostra uma política de bucket que usa a chave de condição `s3:object-lock-remaining-retention-days` para definir um período de retenção máximo de 10 dias.

```
{  
    "Version": "2012-10-17",  
    "Id": "<SetRetentionLimits",  
    "Statement": [  
        {  
            "Sid": "<SetRetentionPeriod",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": [  
                "s3:PutObjectRetention"  
            ],  
            "Resource": "arn:aws:s3:::<awsexamplebucket1>/*",  
            "Condition": {  
                "NumericGreaterThan": {  
                    "s3:object-lock-remaining-retention-days": "10"  
                }  
            }  
        }  
    ]  
}
```

Note

Se seu bucket for o bucket de destino de uma política de replicação e você quiser definir períodos de retenção mínimos e máximos permitidos para as réplicas de objetos que foram criados usando a replicação, você deverá incluir a ação `s3:ReplicateObject` em sua política de bucket.

Para obter mais informações, consulte os tópicos a seguir:

- [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#)
- [Exemplo: operações de objeto \(p. 442\)](#)
- [Exemplos de chave de condição do Amazon S3 \(p. 447\)](#)

Gerenciar marcadores de exclusão e ciclos de vida de objetos

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 o objeto nem suas versões. No entanto, isso faz com que o Amazon S3 se comporte na maioria das vezes como se o objeto tivesse sido excluído. Para obter mais informações, consulte [Trabalhar com marcadores de exclusão \(p. 702\)](#).

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 de gerenciamento de ciclo de vida do objeto 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 objetos, consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

Usar o bloqueio de objetos do S3 com replicação

Você pode usar o bloqueio de objetos do S3 com replicação para habilitar a cópia automática e assíncrona de objetos bloqueados e seus metadados de retenção em buckets do S3, em diferentes ou nas mesmas

Regiões da AWS. Ao usar a replicação, os objetos em um bucket de origem são replicados para um bucket de destino. Para obter mais informações, consulte [Replicação de objetos \(p. 801\)](#).

Para definir um bloqueio de objetos do S3 com replicação, escolha uma das opções a seguir.

Opção 1: habilitar o bloqueio de objeto primeiro

1. Habilite o bloqueio de objetos no bucket de destino ou nos buckets de origem e de destino.
2. Defina a replicação entre os buckets de origem e destino.

Opção 2: configurar a replicação primeiro

1. Defina a replicação entre os buckets de origem e destino.
2. Habilite o bloqueio de objetos apenas no bucket de destino ou nos buckets de origem e de destino.

Para ativar o bloqueio de objetos nas opções anteriores, isso deve ser feito no momento da criação do bucket ou você deve entrar em contato com o AWS Support se estiver usando um bucket existente. Isso é necessário para assegurar que a replicação seja configurada corretamente.

Antes de entrar em contato com o AWS Support, revise os seguintes requisitos para configurar o bloqueio de objetos com replicação:

- O bucket de destino do Amazon S3 deve ter o bloqueio de objetos habilitado.
- É necessário conceder duas novas permissões no bucket de origem do S3 na função do AWS Identity and Access Management (IAM) usada para configurar a replicação. As duas novas permissões são `s3:GetObjectRetention` e `s3:GetObjectLegalHold`. Se a função tiver uma permissão `s3:Get*`, ela satisfaz os requisitos. Para obter mais informações, consulte [Configuração de permissões \(p. 818\)](#).

Para obter mais informações sobre bloqueio de objetos do S3, consulte [Como o bloqueio de objetos do S3 funciona \(p. 723\)](#).

Uso de classes de armazenamento do Amazon S3

Cada objeto no Amazon S3 tem uma classe de armazenamento associada a ele. 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. O Amazon S3 oferece uma variedade de classes de armazenamento para os objetos que você armazena. Escolha uma classe de acordo com seu cenário de caso de uso e dos requisitos de acesso de performance. Todas essas classes de armazenamento oferecem alta durabilidade.

As seções a seguir fornecem detalhes das várias classes de armazenamento e como definir a classe de armazenamento para seus objetos.

Tópicos

- [Classes de armazenamento de objetos acessados com frequência \(p. 732\)](#)
- [Classe de armazenamento para otimizar automaticamente dados com padrões de acesso alterados ou desconhecidos \(p. 732\)](#)
- [Classes de armazenamento de objetos acessados com pouca frequência \(p. 733\)](#)
- [Classes de armazenamento para arquivamento de objetos \(p. 734\)](#)
- [Classe de armazenamento para o Amazon S3 no Outposts \(p. 735\)](#)
- [Comparar as classes de armazenamento do Amazon S3 \(p. 735\)](#)
- [Configurar a classe de armazenamento de um objeto \(p. 736\)](#)

Classes de armazenamento de objetos acessados com frequência

Para casos de uso nos quais a performance é importante (aqueles que exigem tempo de acesso de milissegundos) e dados acessados com frequência, o Amazon S3 fornece as seguintes classes de armazenamento:

- S3 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 S3 Standard.
- Redundância reduzida:a classe Armazenamento de redundância reduzida (RRS) foi criada para dados reproduzíveis não críticos que podem ser armazenados em níveis de redundância menores do que a classe S3 Standard.

Important

Não recomendamos o uso dessa classe de armazenamento. A classe de armazenamento S3 Standard é mais econômica.

Para durabilidade, os objetos RRS têm uma perda anual média prevista de 0,01%. Se um objeto de RRS for perdido, quando forem feitas solicitações a ele, o Amazon S3 retornará um erro 405.

Classe de armazenamento para otimizar automaticamente dados com padrões de acesso alterados ou desconhecidos

O S3 Intelligent-Tiering é uma classe de armazenamento do Amazon S3 projetada para otimizar os custos de armazenamento ao mover automaticamente os dados para o nível de acesso mais econômico, sem impacto na performance ou sobrecarga operacional. É o único armazenamento na nuvem que oferece economia de custo automática ao mover dados em um nível granular de objeto entre níveis de acesso quando há alteração nos padrões de acesso. O S3 Intelligent-Tiering é a classe de armazenamento perfeita quando você quer otimizar os custos de armazenamento dos dados com padrões de acesso desconhecidos ou variáveis. Não há taxas de recuperação para o S3 Intelligent-Tiering.

Por uma pequena taxa mensal de automação e monitoramento de objetos, o S3 Intelligent-Tiering monitora os padrões de acesso e move automaticamente os objetos que não foram acessados para níveis de acesso de baixo custo. O S3 Intelligent-Tiering oferece economia automática de custos de armazenamento em dois níveis de acesso de baixa latência e alta taxa de transferência. Para dados que podem ser acessados de forma assíncrona, os clientes podem optar por ativar os recursos de arquivamento automático na classe de armazenamento do S3 Intelligent-Tiering. O S3 Intelligent-Tiering foi projetado para oferecer 99,9% de disponibilidade e 99,9999999% de durabilidade.

O S3 Intelligent-Tiering armazena objetos automaticamente em três níveis de acesso: Frequent Access (Acesso frequente), Infrequent Access(Acesso infrequente) e Archive Instant Access (Acesso instantâneo ao arquivo). Os objetos que foram carregados ou transferidos para o S3 Intelligent-Tiering são automaticamente armazenados no nível Acesso frequente. O S3 Intelligent-Tiering trabalha monitorando os padrões de acesso e, em seguida, move os objetos que não foram acessados durante 30 dias consecutivos para o nível Acesso infrequente. Com o S3 Intelligent-Tiering, todos os objetos que não foram acessados durante 90 dias consecutivos serão movidos automaticamente para o nível Archive Instant Access (Acesso instantâneo ao arquivo). Você pode configurar o S3 Intelligent-Tiering como sua classe de armazenamento padrão para dados recém-criados ou pode optar por ativar um ou ambos os níveis de acesso de arquivamento usando a API com [PutBucketIntelligentTieringConfiguration](#), a CLI ou o console do Amazon S3. Depois que você ativa um ou ambos os níveis de acesso de arquivamento, o S3 Intelligent-Tiering move automaticamente os objetos que não foram acessados durante 90 dias consecutivos para o

nível de Acesso de arquivamento e depois de 180 dias consecutivos sem acesso para o nível de Acesso de arquivamento profundo. Para obter mais informações sobre como usar o S3 Intelligent-Tiering, consulte [Como usar o S3 Intelligent-Tiering \(p. 739\)](#)

Para acessar objetos arquivados posteriormente, primeiro você precisa restaurá-los. Para obter mais informações, consulte [Restaurar objetos dos níveis Archive Access e Deep Archive Access do S3 Intelligent-Tiering \(p. 744\)](#).

Note

Se o tamanho de um objeto for menor que 128 KB, ele não será monitorado nem qualificado para o nivelamento automático. Objetos menores são sempre armazenados no nível Acesso frequente. Para obter mais informações sobre o S3 Intelligent-Tiering, consulte [Níveis de acesso do S3 Intelligent-Tiering \(p. 738\)](#)

Classes de armazenamento de objetos acessados com pouca frequência

As classes de armazenamento S3 Standard – IA e S3 One Zone – IA foram desenvolvidas para dados duradouros e acessados com pouca frequência. (IA é a sigla em inglês para acesso pouco frequente.) Os objetos S3 Standard – IA e S3 One Zone – IA estão disponíveis para acesso de milissegundos (o mesmo que a classe de armazenamento S3 Standard). O Amazon S3 cobra uma taxa de recuperação para esses objetos, portanto, eles são mais adequados para dados acessados com pouca frequência. Para obter informações sobre a definição de preços, consulte [Definição de preços do Amazon S3](#).

Por exemplo, é possível escolher as classes de armazenamento S3 Standard – IA e S3 One Zone – IA para fazer o seguinte:

- 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 S3 Standard e usar a configuração de ciclo de vida para solicitar que o Amazon S3 transfira os objetos para a classe S3 Standard – IA ou S3 One Zone – IA.

Para obter mais informações sobre gerenciamento de ciclo de vida, consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

Note

As classes de armazenamento S3 Standard – IA e S3 One Zone – IA são adequadas para objetos maiores que 128 KB que você planeja 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 fim do período mínimo de duração do armazenamento de 30 dias, você será cobrado por 30 dias. Para obter informações sobre a definição de preços, consulte [Definição de preços do Amazon S3](#).

As diferenças entre essas classes de armazenamento são:

- S3 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 S3 Standard). Os objetos S3 Standard – IA são resistentes à perda de uma zona de disponibilidade. Essa classe de armazenamento oferece maior disponibilidade e resilienteza que a classe S3 One Zone – IA.
- S3 One Zone – IA: o Amazon S3 armazena dados de objeto em apenas uma zona de disponibilidade, e isso a torna menos cara que a classe S3 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 S3 One Zone – 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 [Comparar as classes de armazenamento do Amazon S3 \(p. 735\)](#) no fim desta seção. Para obter informações sobre a definição de preços, consulte [Definição de preços do Amazon S3](#).

Recomendamos o seguinte:

- S3 Standard – IA: use para a cópia principal ou única de dados, que não pode ser recriada.
- S3 One Zone-IA: use se você puder reciar os dados em caso de falha da zona de disponibilidade, e para réplicas de objeto ao configurar a replicação entre regiões (CRR) do S3.

Classes de armazenamento para arquivamento de objetos

As classes de armazenamento S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive são projetadas para o arquivamento de dados de baixo custo. Essas classes de armazenamento oferecem a mesma durabilidade e resiliência que a classe de armazenamento S3 Standard e S3 Standard – IA. Para obter uma comparação entre a durabilidade e a disponibilidade da classe de armazenamento, consulte [Comparar as classes de armazenamento do Amazon S3 \(p. 735\)](#).

As diferenças entre essas classes de armazenamento são:

- S3 Glacier Instant Retrieval: use para arquivar dados que raramente são acessados e exigem recuperação de milissegundos. Os dados armazenados na classe de armazenamento S3 Glacier Instant Retrieval oferecem economia de custos em comparação à classe de armazenamento S3 Standard – IA, com a mesma latência e performance de taxa de transferência que a classe de armazenamento S3 Standard – IA. Os custos de acesso a dados do S3 Glacier Instant Retrieval são mais altos que os do S3 Standard – IA. Para obter informações sobre a definição de preços, consulte [Definição de preços do Amazon S3](#).
- S3 Glacier Flexible Retrieval: use para arquivos nos quais partes dos dados podem precisar ser recuperadas em minutos. Os dados armazenados na classe de armazenamento S3 Glacier Flexible Retrieval têm um período mínimo de duração do armazenamento de 90 dias e podem ser acessados em apenas 1 a 5 minutos usando a recuperação acelerada. O tempo de recuperação é flexível e você pode solicitar recuperações em massa gratuitas em até 5 a 12 horas. Se você excluiu, substituiu ou fez a transição de um objeto para outra classe de armazenamento antes do período mínimo de 90 dias, será cobrado pelos 90 dias. Para obter informações sobre a definição de preços, consulte [Definição de preços do Amazon S3](#).
- S3 Glacier Deep Archive: use para arquivar dados que raramente precisam ser acessados. Os dados armazenados na classe de armazenamento S3 Glacier Deep Archive têm um período mínimo de duração do armazenamento de 180 dias e um tempo de recuperação padrão de 12 horas. Se você excluiu, substituiu ou fez a transição de um objeto para outra classe de armazenamento antes do período mínimo de 180 dias, será cobrado pelos 180 dias. Para obter informações sobre a definição de preços, consulte [Definição de preços do Amazon S3](#).

S3 Glacier Deep Archive é a opção de armazenamento com o menor custo na AWS. Os custos de armazenamento do S3 Glacier Deep Archive são mais baratos que os da classe de armazenamento S3 Glacier Flexible Retrieval. É possível reduzir os custos de recuperação do S3 Glacier Deep Archive usando a recuperação em lote, o que retorna dados em até 48 horas.

Recuperar objetos arquivados

É possível definir a classe de armazenamento de um objeto como S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive da mesma maneira que você faz para outras classes de armazenamento, conforme descrito na seção [Configurar a classe de armazenamento de um objeto \(p. 736\)](#). Porém, os objetos do

S3 Glacier Flexible Retrieval e do S3 Glacier Deep Archive não estão disponíveis para acesso em tempo real. Primeiro, é necessário restaurar os objetos do S3 Glacier Flexible Retrieval e do S3 Glacier Deep Archive antes de acessá-los. (Os objetos do S3 Standard, RRS, S3 Standard – IA, S3 One Zone – IA, S3 Glacier Instant Retrieval e S3 Intelligent-Tiering estão disponíveis para acesso a qualquer momento.) Para obter mais informações sobre a recuperação de objetos em arquivamento, consulte [Restaurar um objeto arquivado \(p. 715\)](#).

Important

Quando você escolhe a classe de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, os objetos permanecem no Amazon S3. Não será possível acessá-los diretamente por meio do serviço separado do Amazon S3 Glacier.

Para saber mais sobre o serviço Amazon S3 Glacier, consulte o [Guia do desenvolvedor do Amazon S3 Glacier](#).

Classe de armazenamento para o Amazon S3 no Outposts

Com o Amazon S3 on Outposts, é possível criar buckets do S3 em seus recursos do AWS Outposts, além de armazenar e recuperar objetos no ambiente on-premises para aplicações que exigem acesso a dados locais, processamento de dados local e residência de dados. É possível usar as mesmas APIs e recursos no AWS Outposts da mesma maneira que no Amazon S3, incluindo políticas de acesso, criptografia e atribuição de tags. Você pode usar o S3 on Outposts por meio do AWS Management Console, da AWS CLI, de AWS SDKs ou da API REST.

O S3 no Outposts fornece uma nova classe de armazenamento, o S3 Outposts (**OUTPOSTS**). A classe de armazenamento do S3 Outposts só está disponível para objetos armazenados em buckets no Outposts. Se você tentar usar essa classe de armazenamento com um bucket do S3 em uma Região da AWS, será gerado o erro `InvalidStorageClass`. Além disso, se você tentar usar outras classes de armazenamento do S3 com objetos armazenados em buckets do S3 on Outposts, ocorrerá o mesmo erro.

Os objetos armazenados na classe de armazenamento do S3 Outposts (**OUTPOSTS**) são sempre criptografados usando criptografia do lado do servidor com chaves de criptografia gerenciadas do Amazon S3 (SSE-S3). Para obter mais informações, consulte [Proteção de dados usando criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 \(SSE-S3\) \(p. 378\)](#).

Você também pode optar explicitamente por criptografar objetos armazenados na classe de armazenamento do S3 Outposts usando criptografia do lado do servidor com chaves de criptografia fornecidas pelo cliente (SSE-C). Para obter mais informações, consulte [Proteger dados usando a criptografia de servidor com chaves de criptografia fornecidas pelo cliente \(SSE-C\) \(p. 390\)](#).

Note

O S3 on Outposts não é compatível com a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS).

Para obter mais informações sobre o S3 on Outposts, consulte [O que é o Amazon S3 on Outposts? \(p. 1318\)](#).

Comparar as classes de armazenamento do Amazon S3

A tabela a seguir compara as classes de armazenamento, incluindo sua disponibilidade, durabilidade, duração mínima de armazenamento e outras considerações.

Storage Class	Designed for	Durability (designed for)	Availability (designed for)	Availability Zones	Min storage duration	Min billable object size	Other Considerations
STANDARD	Frequently accessed data	99.999999999%	99.99%	>= 3	None	None	None
STANDARD_IA	Long-lived, infrequently accessed data	99.999999999%	99.9%	>= 3	30 days	128 KB	Per GB retrieval fees apply.
INTELLIGENT_TIERING	Long-lived data with changing or unknown access patterns	99.999999999%	99.9%	>= 3	30 days	None	Monitoring and automation fees per object apply. No retrieval fees.
ONEZONE_IA	Long-lived, infrequently accessed, non-critical data	99.999999999%	99.5%	1	30 days	128 KB	Per GB retrieval fees apply. Not resilient to the loss of the Availability Zone.
GLACIER	Long-term data archiving with retrieval times ranging from minutes to hours	99.999999999%	99.99% (after you restore objects)	>= 3	90 days	None	Per GB retrieval fees apply. You must first restore archived objects before you can access them. For more information, see Restoring Archived Objects .
DEEP_ARCHIVE	Archiving rarely accessed data with a default retrieval time of 12 hours	99.999999999%	99.99% (after you restore objects)	>= 3	180 days	None	Per GB retrieval fees apply. You must first restore archived objects before you can access them. For more information, see Restoring Archived Objects .
RRS (Not recommended)	Frequently accessed, non-critical data	99.99%	99.99%	>= 3	None	None	None

Todas as classes de armazenamento, exceto a S3 One Zone-IA, são desenvolvidas para serem resistentes à perda física de uma zona de disponibilidade resultante de desastres.

Considere os custos, além dos requisitos de performance do cenário da sua aplicação. Para obter informações sobre os preços das classes de armazenamento, consulte [Definição de preços do Amazon S3](#).

Configurar a classe de armazenamento de um objeto

Para criar e atualizar classes de armazenamento de objetos, use o console do Amazon S3, os AWS SDKs ou a AWS Command Line Interface (AWS CLI). Cada uma usa as APIs do Amazon S3 para enviar solicitações ao Amazon S3.

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 esse cabeçalho não for adicionado, o Amazon S3 usará a classe de armazenamento padrão, Standard.
- Também é possível alterar a classe de armazenamento de um objeto que já está armazenado no Amazon S3 para qualquer outra classe de armazenamento fazendo uma cópia desse objeto usando a API [PUT Object - Copy](#). No entanto, não é possível usar [PUT Object - Copy](#) para copiar objetos que estão armazenados nas classes de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive. Você também não pode fazer a transição do S3 One Zone-IA para o S3 Glacier Instant Retrieval.

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 o 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 S3 Lifecycle em um bucket. Para obter mais informações, consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).
- Ao definir uma configuração de replicação, você poderá definir a classe de armazenamento para objetos replicados como qualquer outra classe de armazenamento. No entanto, não é possível replicar objetos

que estão armazenados nas classes de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive. Para obter mais informações, consulte [Configuração de replicação \(p. 808\)](#).

Restrição de permissões de política de acesso a uma classe de armazenamento específica

Ao conceder permissões de política de acesso para operações do Amazon S3, é possível usar a chave de condição `s3:x-amz-storage-class` para restringir qual classe de armazenamento a ser usada ao armazenar objetos obtidos por upload. Por exemplo, ao conceder permissão de `s3:PUTObject`, você pode restringir uploads de objetos a uma classe de armazenamento específica. Para ver um exemplo de política, consulte [Exemplo 5: restrição de uploads de objetos com uma classe de armazenamento específica \(p. 452\)](#).

Para obter mais informações sobre como usar condições em políticas e uma lista completa de chaves de condição do Amazon S3, consulte o seguinte:

- [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#)
- [Exemplos de chave de condição do Amazon S3 \(p. 447\)](#)

Amazon S3 Intelligent-Tiering

A classe de armazenamento S3 Intelligent-Tiering foi projetada para otimizar os custos de armazenamento movendo automaticamente os dados para o nível de acesso ao armazenamento de melhor custo-benefício quando os padrões de acesso mudam, sem sobrecarga operacional nem impacto na performance. Por uma cobrança pequena mensal de automação e monitoramento de objetos, o S3 Intelligent-Tiering monitora os padrões de acesso e move automaticamente os objetos que não foram acessados para níveis de acesso de baixo custo.

O S3 Intelligent-Tiering oferece economia automática de custos de armazenamento em três níveis de acesso de baixa latência e alta taxa de transferência. Para dados que podem ser acessados de forma assíncrona, você pode optar por ativar os recursos de arquivamento automático na classe de armazenamento do S3 Intelligent-Tiering. Não há cobranças de recuperação no S3 Intelligent-Tiering. Se um objeto no nível Infrequent Access ou Archive Instant Access for acessado posteriormente, ele será automaticamente movido de volta para o nível Frequent Access. Nenhuma cobrança adicional de nível se aplica quando objetos são movidos entre os níveis de acesso na classe de armazenamento do S3 Intelligent-Tiering.

O S3 Intelligent-Tiering é a classe de armazenamento recomendada para dados com padrões de acesso desconhecidos, alterados ou imprevisíveis, independentemente do tamanho do objeto ou do período de retenção, como data lakes, análise de dados e novas aplicações.

Para obter informações sobre como usar o S3 Intelligent-Tiering, consulte as seguintes seções:

Tópicos

- [Como o S3 Intelligent-Tiering funciona \(p. 737\)](#)
- [Como usar o S3 Intelligent-Tiering \(p. 739\)](#)
- [Como gerenciar o S3 Intelligent-Tiering \(p. 742\)](#)

Como o S3 Intelligent-Tiering funciona

A classe de armazenamento S3 Intelligent-Tiering armazena objetos automaticamente em três níveis de acesso. Um nível é otimizado para acesso frequente, um nível de menor custo é otimizado para acesso

pouco frequente e outro nível de baixo custo é otimizado para dados raramente acessados. Por uma cobrança mensal baixa de automação e monitoramento de objetos, o S3 Intelligent-Tiering monitora os padrões de acesso e move automaticamente para o nível Infrequent Access os objetos que não foram acessados por 30 dias consecutivos. Após 90 dias sem acesso, os objetos são movidos para a camada Archive Instant Access sem causar impacto na performance ou sobrecarga operacional.

Para obter o menor custo de armazenamento em dados que podem ser acessados de minutos a horas, você pode optar por ativar outros recursos de arquivo. Depois que você ativa os recursos de arquivo opcionais, o S3 Intelligent-Tiering move os objetos que não foram acessados durante 90 dias consecutivos para o nível Archive Access. Após 180 dias consecutivos sem acesso, os objetos são movidos para o nível Deep Archive Access.

Não há cobranças de recuperação no S3 Intelligent-Tiering. Se um objeto no nível Infrequent Access ou Archive Instant Access for acessado posteriormente, ele será automaticamente movido de volta para o nível Frequent Access. O acesso será iniciado ao fazer o download ou copiar um objeto por meio do AWS Management Console ou executando [GetObject](#) ou [CopyObject](#). A execução de uma operação HEAD, LIST, GET tags de objetos ou PUT tags de objetos não constituirá acesso.

Você pode configurar o S3 Intelligent-Tiering como sua classe de armazenamento padrão para dados recém-criados especificando INTELLIGENT-TIERING em seu cabeçalho de solicitação da [API de PUT do S3](#). O S3 Intelligent-Tiering foi projetado para oferecer 99,9% de disponibilidade e 99,9999999% de durabilidade.

Note

Se o tamanho do objeto for menor que 128 KB, ele não será monitorado nem qualificado para o nivelamento automático. Objetos menores são sempre armazenados no nível Acesso frequente.

Níveis de acesso do S3 Intelligent-Tiering

Nível de Acesso frequente (automático)

Este é o nível de acesso padrão no qual qualquer objeto criado ou transicionado para o S3 Intelligent-Tiering inicia seu ciclo de vida. Um objeto permanecerá neste nível enquanto estiver sendo acessado. O nível de Acesso frequente oferece baixa latência e alta performance de throughput.

Nível de Acesso infrequente (automático)

Se um objeto não for acessado por 30 dias consecutivos, o objeto se moverá para o nível de Acesso infrequente. O nível Acesso infrequente fornece baixa latência e alta performance de throughput.

Nível Archive Instant Access (automático)

Se o objeto não for acessado por 90 dias consecutivos, ele será movido para o nível Archive Instant Access. O nível Archive Instant Access fornece baixa latência e alta performance de throughput.

Nível de Acesso de arquivamento (opcional)

O S3 Intelligent-Tiering oferece a opção de ativar o nível Archive Access para dados que podem ser acessados de maneira assíncrona. Após a ativação, o nível Archive Access arquiva os objetos que não foram acessados durante um mínimo de 90 dias consecutivos. Você pode estender o último tempo de acesso para arquivamento para um máximo de 730 dias. O nível Archive Access tem a mesma performance que a classe de armazenamento do [S3 Glacier Flexible Retrieval](#). Os tempos de recuperação padrão para esse nível de acesso podem variar de 3 a 5 horas.

Note

Ative apenas o nível Archive Access por 90 dias se você quiser ignorar o nível Archive Instant Access. O nível Archive Access oferece um custo de armazenamento um pouco menor com

tempos de recuperação de minutos a horas. O nível Archive Instant Access oferece acesso de milissegundos e alta performance de throughput.

Nível Acesso de arquivamento profundo (opcional)

O S3 Intelligent-Tiering oferece a opção de ativar o nível Deep Archive Access para dados que podem ser acessados de maneira assíncrona. Após a ativação, o nível Deep Archive Access arquiva os objetos que não foram acessados durante um mínimo de 180 dias consecutivos. Você pode estender o último tempo de acesso para arquivamento para um máximo de 730 dias. O nível de Acesso de arquivamento profundo tem a mesma performance que a classe de armazenamento [S3 Glacier Deep Archive](#). A recuperação padrão de objetos desse nível de acesso ocorre em até 12 horas.

Note

Ative os níveis Acesso de arquivamento e Acesso de arquivamento profundo somente se seus objetos puderem ser acessados de forma assíncrona pela aplicação. Se o objeto que você está recuperando estiver armazenado nos níveis Acesso de arquivamento ou Acesso de arquivamento profundo, restaure o objeto usando `RestoreObject`. Para obter mais informações, consulte [Restaurar objetos dos níveis Archive Access e Deep Archive Access do S3 Intelligent-Tiering \(p. 744\)](#).

Como usar o S3 Intelligent-Tiering

Você pode usar a classe de armazenamento S3 Intelligent-Tiering para otimizar automaticamente os custos de armazenamento. O S3 Intelligent-Tiering oferece economia de custo automática ao mover dados em um nível detalhado de objeto entre níveis de acesso quando há alteração nos padrões de acesso. Para dados que podem ser acessados de forma assíncrona, você pode optar por habilitar o arquivamento automático na classe de armazenamento do S3 Intelligent-Tiering usando o AWS Management Console, a AWS CLI ou a API do Amazon S3.

Como mover dados para o S3 Intelligent-Tiering

Há duas maneiras de mover dados para o S3 Intelligent-Tiering. Você pode inserir dados diretamente no S3 Intelligent-Tiering especificando `INTELLIGENT_TIERING` no cabeçalho do `x-amz-storage-class` ou configurar as políticas do S3 Lifecycle para fazer a transição de objetos do S3 Standard ou S3 Standard-Infrequent Access para o S3 Intelligent-Tiering.

Carregar dados para o S3 Intelligent-Tiering usando o Direct PUT

Quando você carrega um objeto para a classe de armazenamento do S3 Intelligent-Tiering usando a operação de API [PUT](#), você especifica S3 Intelligent-Tiering no [cabeçalho da solicitação de x-amz-storage-class](#).

A seguinte solicitação armazena a imagem, `my-image.jpg`, no bucket `myBucket`. A solicitação usa o cabeçalho `x-amz-storage-class` para solicitar que o objeto seja armazenado usando a classe de armazenamento do S3 Intelligent-Tiering.

Example

```
PUT /my-image.jpg HTTP/1.1
Host: myBucket.s3.<Region>.amazonaws.com (http://amazonaws.com/)
Date: Wed, 1 Sep 2021 17:50:00 GMT
Authorization: authorization string
Content-Type: image/jpeg
Content-Length: 11434
Expect: 100-continue
x-amz-storage-class: INTELLIGENT_TIERING
```

Como fazer transição de dados para o S3 Intelligent-Tiering do S3 Standard ou do S3 Standard-Infrequent Access usando o S3 Lifecycle

É possível adicionar regras a uma configuração do S3 Lifecycle para solicitar que o Amazon S3 faça a transição de objetos de uma classe de armazenamento para outra. Para obter informações sobre transições suportadas e restrições relacionadas, consulte [Como fazer transição de objetos usando o S3 Lifecycle](#).

É possível especificar políticas do S3 Lifecycle no nível do bucket ou do prefixo. Nesta regra de configuração do S3 Lifecycle, o filtro especifica um prefixo das chaves (`documents/`). Portanto, a regra aplica-se a objetos com o prefixo de nome de chave `documents/`, como `documents/doc1.txt` e `documents/doc2.txt`. A regra especifica uma ação de `Transition` direcionando o Amazon S3 para fazer a transição de objetos para a classe de armazenamento do S3 Intelligent-Tiering 0 dia após a criação. Nesse caso, os objetos são elegíveis para transição para o S3 Intelligent-Tiering à meia-noite UTC, após a criação.

Example

```
<LifecycleConfiguration>
  <Rule>
    <ID>ExampleRule</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>INTELLIGENT_TIERING</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Habilitar níveis Archive Access e Deep Archive Access do S3 Intelligent-Tiering

Para obter o custo de armazenamento mais baixo sobre dados que podem ser acessados de minutos a horas, você pode ativar um ou ambos os níveis de acesso de arquivo criando uma configuração no nível de bucket, prefixo ou etiqueta de objeto usando o AWS Management Console, a AWS CLI ou a API do Amazon S3.

Uso do console do S3

Para habilitar o arquivamento automático do S3 Intelligent-Tiering

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista de Buckets, escolha o nome do bucket desejado.
3. Escolha Properties (Propriedades).
4. Navegue até a seção S3 Intelligent-Tiering Archive configurations (Configurações do S3 Intelligent-Tiering) e escolha Create configuration (Criar configuração).
5. Na seção Archive configuration settings (Definições de configuração do arquivamento), especifique um nome de configuração descritiva para a configuração de arquivamento do S3 Intelligent-Tiering.
6. Em Choose a configuration scope (Escolher um escopo de configuração), escolha um escopo de configuração a ser usado. Opcionalmente, você pode limitar o escopo de configuração a objetos especificados em um bucket usando um prefixo compartilhado, uma etiqueta de objeto ou uma combinação dos dois.

- a. Para limitar o escopo da configuração, selecione Limit the scope of this configuration using one or more filters (Limitar o escopo dessa configuração usando um ou mais filtros).
 - b. Para limitar o escopo da configuração usando um único prefixo, insira o prefixo em Prefix (Prefixo).
 - c. Para limitar o escopo da configuração usando etiquetas de objeto, selecione Add tag (Adicionar etiqueta) e insira um valor para Key (Chave).

7. Em Status, selecione Enable (Habilitar).

8. Na seção Archive settings (Configurações de arquivamento), selecione um ou ambos os níveis de Acesso de arquivamento a serem ativados.

9. Escolha Create (Criar OpsItem).

Como usar a AWS CLI

Você pode usar os seguintes comandos da AWS CLI para gerenciar as configurações do S3 Intelligent-Tiering:

- `delete-bucket-intelligent-tiering-configuration`
 - `get-bucket-intelligent-tiering-configuration`
 - `list-bucket-intelligent-tiering-configurations`
 - `put-bucket-intelligent-tiering-configuration`

Para obter instruções de configuração da AWS CLI, consulte [Desenvolvimento com o Amazon S3 usando a AWS CLI](#) (p. 1257).

Ao usar a AWS CLI, você não pode especificar a configuração como um arquivo XML. Você deve especificar o JSON em vez disso. A seguir, está um exemplo de configuração XML do S3 Intelligent-Tiering e JSON equivalente que você pode especificar em um comando da AWS CLI.

O exemplo a seguir coloca uma configuração do S3 Intelligent-Tiering no bucket especificado.

Example `put-bucket-intelligent-tiering-configuration`

JSON

```
{
    "Id": "string",
    "Filter": {
        "Prefix": "string",
        "Tag": {
            "Key": "string",
            "Value": "string"
        },
        "And": {
            "Prefix": "string",
            "Tags": [
                {
                    "Key": "string",
                    "Value": "string"
                }
            ...
        ]
    }
},
    "Status": "Enabled"|"Disabled",
    "Tierings": [
        {
            "Name": "string",
            "Type": "string",
            "Order": 0,
            "Tiers": [
                {
                    "Name": "string",
                    "Type": "string",
                    "Order": 0
                }
            ]
        }
    ]
}
```

```
        "Days": integer,
        "AccessTier": "ARCHIVE_ACCESS" | "DEEP_ARCHIVE_ACCESS"
    }
    ...
]
```

XML

```
PUT /?intelligent-tiering&id=Id HTTP/1.1
Host: Bucket.s3.amazonaws.com
<?xml version="1.0" encoding="UTF-8"?>
<IntelligentTieringConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Id>string</Id>
    <Filter>
        <And>
            <Prefix>string</Prefix>
            <Tag>
                <Key>string</Key>
                <Value>string</Value>
            </Tag>
            ...
        </And>
        <Prefix>string</Prefix>
        <Tag>
            <Key>string</Key>
            <Value>string</Value>
        </Tag>
    </Filter>
    <Status>string</Status>
    <Tiering>
        <AccessTier>string</AccessTier>
        <Days>integer</Days>
    </Tiering>
    ...
</IntelligentTieringConfiguration>
```

Usando a operação PUT da API

Você pode usar a operação [PutBucketIntelligentTieringConfiguration](#) para um bucket especificado e até 1.000 configurações do S3 Intelligent-Tiering por bucket. Você pode definir quais objetos em um bucket são elegíveis para os níveis de Acesso de arquivamento usando um prefixo compartilhado ou uma etiqueta de objeto. Usar um prefixo compartilhado ou etiqueta de objeto permite que você alinhe aplicações comerciais, fluxos de trabalho ou organizações internas específicas. Você também tem a flexibilidade de ativar o nível de Acesso de arquivamento, o nível Acesso de arquivamento profundo ou ambos.

Como gerenciar o S3 Intelligent-Tiering

A classe de armazenamento S3 Intelligent-Tiering oferece redução automática de custos de armazenamento em três níveis de acesso de baixa latência e alto throughput. Também oferece recursos de arquivo opcionais para ajudar você a obter os menores custos de armazenamento na nuvem para dados que podem ser acessados de minutos a horas. A classe de armazenamento S3 Intelligent-Tiering oferece suporte a todos os recursos do Amazon S3, inclusive:

- S3 Inventory, para verificar a camada de acesso de objetos
- S3 Replication, para replicação de dados em qualquer região da AWS
- S3 Storage Lens, para exibição de métricas de uso de armazenamento e atividade
- Criptografia do lado do servidor, para dados de objetos

- Bloqueio de objeto do S3, para evitar exclusão acidental
- AWS PrivateLink, para acessar o Amazon S3 por um endpoint privado em uma VPC

Identificar quais objetos do nível de acesso do S3 Intelligent-Tiering estão armazenados

Você pode usar o [Amazon S3 Inventory](#) para obter uma lista de seus objetos e metadados correspondentes, incluindo o nível de acesso do S3 Intelligent-Tiering. O Amazon S3 Inventory fornece uma saída de arquivo CSV, ORC ou Parquet que lista seus objetos e os metadados correspondentes, diariamente ou semanalmente, para um bucket do Amazon S3 ou um prefixo compartilhado. (O prefixo compartilhado se refere a objetos que tenham nomes que começam com uma string em comum.)

Visualizar o status do arquivo de um objeto no S3 Intelligent-Tiering

Configure esse evento de notificação do Amazon S3 para receber um aviso quando um objeto dentro da classe de armazenamento S3 Intelligent-Tiering for movido para o nível Archive Access ou para o nível Deep Archive Access. Para obter mais informações, consulte [Habilitar notificações de eventos](#).

O Amazon S3 pode publicar notificações de eventos em um tópico do Amazon Simple Notification Service (Amazon SNS), em uma fila do Amazon Simple Queue Service (Amazon SQS) ou em uma função do AWS Lambda. Para mais informações, consulte [Notificações de eventos do Amazon S3 \(p. 1083\)](#).

Veja a seguir um exemplo de mensagem que o Amazon S3 envia para publicar um evento `s3:IntelligentTiering`. Para obter mais informações, consulte [Estrutura de mensagens de evento](#).

```
{  
    "Records": [  
        {  
            "eventVersion": "2.3",  
            "eventSource": "aws:s3:",  
            "awsRegion": "us-west-2",  
            "eventTime": "1970-01-01T00:00:00.000Z",  
            "eventName": "IntelligentTiering",  
            "userIdentity": {  
                "principalId": "s3.amazonaws.com"  
            },  
            "requestParameters": {  
                "sourceIPAddress": "s3.amazonaws.com"  
            },  
            "responseElements": {  
                "x-amz-request-id": "C3D13FE58DE4C810",  
                "x-amz-id-2": "FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/JRWeUWerMUE5JgHvANOjpD"  
            },  
            "s3": {  
                "s3SchemaVersion": "1.0",  
                "configurationId": "testConfigRule",  
                "bucket": {  
                    "name": "mybucket",  
                    "ownerIdentity": {  
                        "principalId": "A3NL1KOZZKExample"  
                    },  
                    "arn": "arn:aws:s3:::mybucket"  
                },  
                "object": {  
                    "key": "HappyFace.jpg",  
                    "size": 1024,  
                    "eTag": "d41d8cd98f00b204e9800998ecf8427e",  
                    "versionId": "12345678901234567890123456789012"  
                }  
            }  
        }  
    ]  
}
```

```
        }
    },
    "intelligentTieringEventData":{
        "destinationAccessTier": "ARCHIVE_ACCESS"
    }
}
}
```

Você também pode usar uma [solicitação do objeto HEAD](#) para exibir o status de arquivamento de um objeto. Se um objeto for armazenado usando a classe de armazenamento do S3 Intelligent-Tiering e estiver atualmente em um dos níveis de arquivo, a resposta do objeto HEAD mostrará o nível de arquivo atual. Ele faz isso usando o cabeçalho [x-amz-archive-status](#).

A seguinte solicitação de objeto HEAD retorna os metadados de um objeto.

Example

```
HEAD /my-image.jpg HTTP/1.1
Host: bucket.s3.<Region>.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:02236Q3V0RonhpaBX5sCYVf1bNRuU=
```

As solicitações de objetos HEAD também podem ser usadas para monitorar o status de uma solicitação de [restore-object](#). Se a restauração do arquivo estiver em andamento, a resposta do objeto HEAD incluirá o cabeçalho [x-amz-restore](#).

Veja a seguir um exemplo de resposta do objeto HEAD mostrando um objeto arquivado usando o S3 Intelligent-Tiering com uma solicitação de restauração em andamento.

Example

```
HTTP/1.1 200 OK
x-amz-id-2: FSVaTMjrmBp3Izs1NnwBZeum7M19ii8UbxBmbi0A8AirHANJBo+hEftBuiESACOMJp
x-amz-request-id: E5CEFCB143EB505A
Date: Fri, 13 Nov 2020 00:28:38 GMT
Last-Modified: Mon, 15 Oct 2012 21:58:07 GMT
ETag: "1acccb31fcf202eba0c0f41fa2f09b4d7"
x-amz-storage-class: 'INTELLIGENT_TIERING'
x-amz-archive-status: 'ARCHIVE_ACCESS'
x-amz-restore: 'ongoing-request="true"'
x-amz-restore-request-date: 'Fri, 13 Nov 2020 00:20:00 GMT'
Accept-Ranges: bytes
Content-Type: binary/octet-stream
Content-Length: 300
Server: AmazonS3
```

Restaurar objetos dos níveis Archive Access e Deep Archive Access do S3 Intelligent-Tiering

Para acessar objetos nos níveis Archive Access e Deep Archive Access do S3 Intelligent-Tiering, você deve iniciar a [solicitação de restauração](#) e esperar até que o objeto seja movido para o nível Frequent Access para acessá-lo. Para obter informações sobre objetos arquivados, consulte [Como trabalhar com objetos arquivados](#).

Quando você restaura dos níveis Acesso de arquivamento ou Acesso de arquivamento profundo, o objeto faz a transição de volta para o nível Acesso frequente. Depois, se o objeto não for acessado durante 30 dias consecutivos, ele se moverá automaticamente para o nível Infrequent Access (Acesso infrequente).

Ele passa para o nível de Acesso de arquivamento após um mínimo de 90 dias consecutivos sem acesso. Ele passa para o nível Acesso de arquivamento profundo após um mínimo de 180 dias consecutivos sem acesso.

Não há cobranças de recuperação no S3 Intelligent-Tiering. As recuperações de dados e solicitações de recuperações de dados e restauração **Padrão e Em massa** são gratuitas para os níveis de Acesso de arquivamento e Acesso de arquivamento profundo. As solicitações de restauração subsequentes chamadas em objetos arquivados que já estão sendo restaurados serão cobradas como uma solicitação GET.

Note

Ao restaurar um objeto nos níveis de acesso de arquivo do S3 Intelligent-Tiering, a solicitação de restauração usará a recuperação Standard como opção de recuperação padrão. Você pode especificar a recuperação Padrão ou Em massa em `GlacierJobParameters`. Você também pode especificar a recuperação Expressa no nível de Acesso de arquivamento, que é cobrada de acordo com a taxa de solicitação e recuperação Expressas.

Você pode restaurar um objeto arquivado usando o console do Amazon S3, a API REST, a AWS Command Line Interface (AWS CLI).

Uso do console do S3

Para restaurar um objeto usando o console do Amazon S3.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Bucket name (Nome do bucket), escolha o nome do bucket que contém os objetos que você deseja restaurar.
3. Na lista de Objects (Objetos), selecione um ou mais objetos que você está restaurando, escolha Actions (Ações) e depois Restore from S3 Intelligent-Tiering Archive Access or Deep Archive Access (Restaurar do Acesso de arquivamento ou do Acesso de arquivamento profundo do S3 Intelligent-Tiering).
4. Escolha Restore.

Note

Ao contrário das solicitações de restauração do S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive, não é necessário escolher o nível para o qual deseja atualizar. Os objetos dos níveis Acesso de arquivamento e Acesso de arquivamento profundo do S3 Intelligent-Tiering são restaurados automaticamente para o nível Acesso frequente.

Uso da API REST

O Amazon S3 fornece uma operação de API para que você inicie uma restauração de arquivos. Para obter mais informações, consulte [RestoreObject](#) na Referência da API do Amazon Simple Storage Service.

Uso do AWS CLI

Usar o comando `restore-object` para restaurar objetos dos níveis Acesso de arquivamento ou Acesso de arquivamento profundo do S3 Intelligent-Tiering.

O exemplo a seguir restaura o objeto `dir1/example.obj` em `awsexamplebucket`.

```
aws s3api restore-object --bucket awsexamplebucket --key dir1/example.obj --restore-request '{}'
```

Você pode usar o seguinte comando para monitorar o status de sua solicitação de `restore-object`.

```
aws s3api head-object --bucket awsexamplebucket --key dir1/example.obj
```

Para obter mais informações, consulte [restore-object](#) na Referência de comandos da AWS CLI.

Note

Ao contrário das classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive, as solicitações de restauração para objetos S3 Intelligent-Tiering não aceitam o valor `days`.

Como verificar o status de restauração de um objeto

Você pode verificar o andamento da restauração do seu objeto na página Object overview (Visão geral do objeto) no console do Amazon S3. Para mais informações, consulte [Exibir uma visão geral do objeto no console do Amazon S3 \(p. 271\)](#). Esta página mostrará que a restauração está In progress (Em andamento). Você pode usar a solicitação para ser notificado da conclusão da restauração do objeto usando `s3:ObjectRestore:Completed` com o recurso [Notificações de eventos do Amazon S3](#).

A tabela a seguir resume as velocidades de recuperação de objetos arquivados.

Note

[Recuperações expressão](#) são recursos premium disponíveis para o nível de Acesso de arquivamento do S3 Intelligent-Tiering e são cobrados de acordo com a taxa de solicitação e recuperação expressas.

Para obter informações sobre como pagar pelo Amazon S3, consulte [Preço do Amazon S3](#).

Gerenciando seu ciclo de vida de armazenamento

Para gerenciar seus objetos de maneira que sejam armazenados de maneira econômica durante todo o ciclo de vida, configure o Amazon S3 Lifecycle. Uma configuração do Amazon S3 Lifecycle 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: estas ações 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 S3 Standard – IA 30 dias após a criação deles ou arquivar objetos para a classe de armazenamento do S3 Glacier Flexible Retrieval um ano após a sua criação. Para obter mais informações, consulte [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#).

Há custos associados a solicitações de transição do ciclo de vida. Para obter informações sobre a definição de preços, consulte [Definição de preços do Amazon S3](#).

- Ações de validade: estas ações definem quando os objetos perdem a validade. O Amazon S3 exclui os objetos expirados em seu nome.

Os custos de validade do ciclo de vida dependem da validade que você escolhe para os objetos. Para mais informações, consulte [Expirando objetos \(p. 753\)](#).

Se houver algum atraso entre quando um objeto se tornar elegível para uma ação de ciclo de vida e quando o Amazon S3 transferir ou expirar seu objeto, as alterações de faturamento serão aplicadas assim que o objeto se tornar qualificado para a ação de ciclo de vida. Por exemplo, se um objeto estiver programado para perder a validade e o Amazon S3 não expirar imediatamente o objeto, você não será

cobrado pelo armazenamento após o fim da validade. A única exceção a esse comportamento é se você tiver uma regra de ciclo de vida para a classe de armazenamento S3 Intelligent-Tiering. Nesse caso, as alterações no faturamento não acontecerão até que o objeto tenha feito a transição para o S3 Intelligent-Tiering.

Para obter mais informações sobre regras do Amazon S3 Lifecycle, consulte [Elementos de configuração do ciclo de vida \(p. 767\)](#).

Gerenciando o ciclo de vida do objeto

Defina regras de configuração do Amazon S3 Lifecycle 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 as regras de configuração do S3 Lifecycle, é possível solicitar que o Amazon S3 faça a transição de objetos para classes de armazenamento menos caras, arquive-os ou exclua-os.

Criando uma configuração de ciclo de vida

Uma configuração do S3 Lifecycle é um arquivo XML que consiste em um conjunto de regras com ações predefinidas que você deseja que o Amazon S3 execute em objetos durante sua vida útil.

Você também pode configurar o ciclo de vida usando o console do Amazon S3, a API REST, os AWS SDKs e a AWS Command Line Interface (AWS CLI). Para mais informações, consulte [Definir a configuração do ciclo de vida em um bucket \(p. 753\)](#).

O Amazon S3 fornece um conjunto de operações de API REST para o gerenciamento da configuração de ciclo de vida em um bucket. O Amazon S3 armazena a configuração como um sub-recurso de ciclo de vida que é anexado ao seu bucket. Para obter detalhes, consulte:

[Ciclo de vida de PUT Bucket](#)

[Ciclo de vida de GET Bucket](#)

[DELETE Bucket lifecycle](#)

Para obter mais informações sobre como criar uma configuração de ciclo de vida, consulte os seguintes tópicos:

Tópicos

- [Transição de objetos usando o Amazon S3 Lifecycle \(p. 748\)](#)
- [Expirando objetos \(p. 753\)](#)
- [Definir a configuração do ciclo de vida em um bucket \(p. 753\)](#)
- [Ciclo de vida e outras configurações de bucket \(p. 765\)](#)

- Configurar notificações de evento de ciclo de vida (p. 766)
- Elementos de configuração do ciclo de vida (p. 767)
- Exemplos de configuração do S3 Lifecycle (p. 775)

Transição de objetos usando o Amazon S3 Lifecycle

É possível adicionar regras a uma configuração de ciclo de vida do S3 para solicitar que o Amazon S3 faça a transição de objetos para outra classe de armazenamento do Amazon S3. Para obter mais informações sobre classes de armazenamento, consulte [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#). Alguns exemplos de quando você pode usar as configurações do ciclo de vida do S3 dessa maneira incluem:

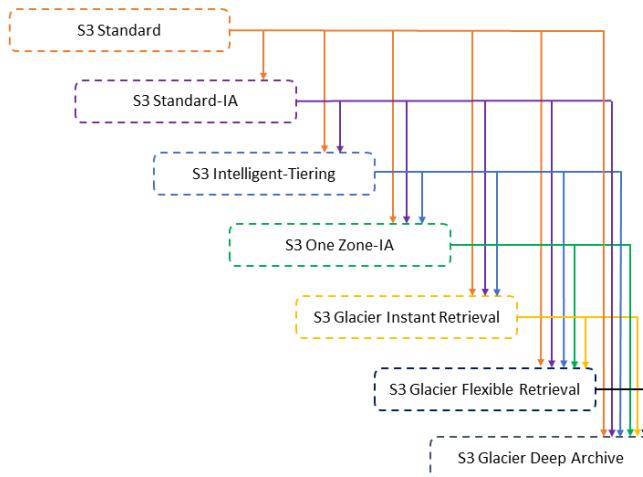
- Quando você sabe que os objetos são acessados com pouca frequência, você pode fazer a transição para a classe de armazenamento S3 Standard – IA.
- Convém arquivar objetos que não precisam de acesso em tempo real à classe de armazenamento S3 Glacier Flexible Retrieval.

As seções a seguir descrevem transições com suporte, limitações relacionadas e a transição para a classe de armazenamento S3 Glacier Flexible Retrieval.

Transições com suporte e limitações relacionadas

Em uma configuração do S3 Lifecycle, 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 você desconhece os padrões de acesso dos objetos ou os padrões de acesso mudam com o passar do tempo, é possível fazer a transição para os objetos para a classe de armazenamento S3 Intelligent-Tiering para economizar automaticamente. Para obter informações sobre classes de armazenamento, consulte [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#).

O Amazon S3 da suporte a um modelo de cacheira para fazer a transição entre classes de armazenamento, conforme mostrado no diagrama a seguir.



Transições de ciclo de vida com suporte

O Amazon S3 oferece suporte às seguintes transições do ciclo de vida entre as classes de armazenamento usando uma configuração do ciclo de vida do S3.

Você pode fazer a transição do seguinte:

- A classe de armazenamento S3 Standard para qualquer outra classe de armazenamento.
- Da classe de armazenamento S3 Standard-IA para S3 Intelligent-Tiering, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive.
- Da classe de armazenamento S3 Intelligent-Tiering para S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive.
- Da classe de armazenamento S3 One Zone-IA para S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive.
- Da classe de armazenamento S3 Glacier Instant Retrieval para S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive.
- A classe de armazenamento S3 Glacier Flexible Retrieval para a classe de armazenamento S3 Glacier Deep Archive.
- Qualquer classe de armazenamento para a S3 Glacier Deep Archive.

Transições de ciclo de vida sem suporte

O Amazon S3 não oferece suporte a nenhuma das seguintes transições do ciclo de vida.

Você não pode fazer a transição do seguinte:

- Nenhuma classe de armazenamento para a classe de armazenamento S3 Standard.
- Qualquer classe de armazenamento para a classe de armazenamento de redundância reduzida (RRS).
- A classe de armazenamento S3 Intelligent-Tiering para a classe de armazenamento S3 Standard – IA.
- A classe de armazenamento S3 One Zone – IA para as classes de armazenamento S3 Intelligent-Tiering, S3 Standard – IA ou S3 Glacier Instant Retrieval.

Restrições

As transições da classe de armazenamento do ciclo de vida têm as seguintes restrições:

Tamanho do objeto e transições de S3 Standard ou S3 Standard – IA para S3 Intelligent-Tiering, S3 Standard – IA ou S3 One Zone – IA

Quando você faz a transição de objetos das classes de armazenamento S3 Standard ou S3 Standard – IA para S3 Intelligent-Tiering, S3 Standard – IA ou S3 One Zone – IA, as seguintes restrições de tamanho de objeto se aplicam:

- Objetos maiores: para as transições a seguir, há um benefício de custo para a transição de objetos maiores:
 - Das classes de armazenamento S3 Standard ou S3 Standard – IA para S3 Intelligent-Tiering.
 - Da classe de armazenamento S3 Standard para S3 Standard – IA ou S3 One Zone – IA.
- Objetos menores que 128 KB: para as transições a seguir, o Amazon S3 não faz transição de objetos menores que 128 KB:
 - Das classes de armazenamento S3 Standard ou S3 Standard – IA para S3 Intelligent-Tiering ou S3 Glacier Instant Retrieval.

- Da classe de armazenamento S3 Standard para S3 Standard – IA ou S3 One Zone – IA.

Note

É possível filtrar regras de ciclo de vida com base no tamanho do objeto.

Dias mínimos para transição de S3 Standard ou S3 Standard – IA para S3 Standard – IA ou S3 One Zone – IA

Antes de fazer a transição de objetos das classes de armazenamento S3 Standard ou S3 Standard – IA para S3 Standard – IA ou S3 One Zone – IA, você deve armazená-los pelo menos 30 dias na classe de armazenamento S3 Standard. Por exemplo, você não pode criar uma regra de ciclo de vida para fazer a transição de objetos para a classe de armazenamento S3 Standard – IA um dia depois de criá-los. O Amazon S3 não faz a transição de objetos nos primeiros 30 dias porque os objetos mais recentes geralmente são acessados com mais frequência ou excluídos mais cedo do que o adequado para armazenamento S3 Standard – IA ou S3 One Zone – IA.

Da mesma forma, se você estiver fazendo a transição de objetos não atuais (em buckets com versões), poderá fazer a transição somente de objetos que não são atuais pelo menos 30 dias para armazenamento S3 Standard – IA ou S3 One Zone – IA.

Taxa mínima de armazenamento de 30 dias para S3 Standard – IA e S3 One Zone – IA

As classes de armazenamento S3 Standard – IA e S3 One Zone – IA têm uma carga mínima de armazenamento de 30 dias. Portanto, você não pode especificar uma única regra de ciclo de vida para uma transição S3 Standard-IA ou S3 One Zone-IA e uma transição S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive quando a transição S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive ocorre menos de 30 dias após a transição S3 Standard-IA ou S3 One Zone-IA.

O mesmo mínimo de 30 dias se aplica ao especificar uma transição do armazenamento S3 Standard – IA para S3 One Zone – IA. 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](#).

Gerenciar o ciclo de vida completo de um objeto

Você pode combinar essas ações do S3 Lifecycle 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, você pode criar uma regra do S3 Lifecycle especificando a ação de transição inicial para armazenamento S3 Intelligent-Tiering, S3 Standard – IA ou S3 One Zone – IA, outra ação de transição para o armazenamento S3 Glacier Flexible Retrieval 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](#).

Transição para as classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive (arquivamento de objetos)

Usando a configuração do S3 Lifecycle, você pode fazer a transição de objetos para as classes de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive para arquivamento. Quando você escolhe a classe de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, os objetos permanecem no Amazon S3. Não será possível acessá-los diretamente por meio do serviço

separado do Amazon S3 Glacier. Para obter mais informações gerais sobre o S3 Glacier, consulte [O que é o Amazon S3 Glacier](#) no Guia do desenvolvedor do Amazon S3 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 armazenados nas classes de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive 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 S3 Glacier Flexible Retrieval.

Você pode restaurar um objeto usando o console do Amazon S3 ou, programaticamente, usando bibliotecas wrapper dos AWS SDKs ou a API REST do Amazon S3 em seu código. Para mais informações, consulte [Restaurar um objeto arquivado \(p. 715\)](#).

- Os objetos armazenados na classe de armazenamento S3 Glacier Flexible Retrieval só podem ser transferidos para a classe de armazenamento S3 Glacier Deep Archive.

Você pode usar uma regra de configuração do S3 Lifecycle somente para converter a classe de armazenamento de um objeto do S3 Glacier Flexible Retrieval para a classe de armazenamento S3 Glacier Deep Archive. Se quiser alterar a classe de armazenamento de um objeto armazenado no S3 Glacier Flexible Retrieval para uma classe de armazenamento diferente do S3 Glacier Deep Archive, use a operação de restauração para fazer uma cópia temporária do objeto primeiro. Depois, use a operação de cópia para substituir o objeto especificando S3 Standard, S3 Intelligent-Tiering, S3 Standard – IA, S3 One Zone – IA ou Reduced Redundancy como a classe de armazenamento.

- A transição de objetos para a classe de armazenamento S3 Glacier Deep Archive é apenas unidirecional.

Não é possível usar uma regra de configuração do S3 Lifecycle para converter a classe de armazenamento de um objeto do S3 Glacier Deep Archive para 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. Depois, use a operação de cópia para substituir o objeto especificando S3 Standard, S3 Intelligent-Tiering, S3 Standard – IA, S3 One Zone – IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval ou armazenamento de redundância reduzida como a classe de armazenamento.

- Os objetos armazenados nas classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive são visíveis e ficam disponíveis somente por meio do Amazon S3. Eles não estão disponíveis por meio do serviço separado do Amazon S3 Glacier.

Esses são objetos do Amazon S3 e você só pode acessá-los usando o console do Amazon S3 ou a API do Amazon S3. Você não pode acessar os objetos arquivados por meio do console separado do Amazon S3 Glacier ou da API do Amazon S3 Glacier.

Considerações sobre custos

Se você estiver planejando arquivar dados acessados com pouca frequência por um período de meses ou anos, as classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive podem reduzir seus custos de armazenamento. No entanto, para garantir que a classe de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive seja apropriada para você, considere o seguinte:

- Cobranças extras de armazenamento: quando você faz a transição de objetos para a classe de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, uma quantidade fixa de armazenamento é adicionada a cada objeto para acomodar metadados para gerenciar o objeto.
 - Para cada objeto arquivado no S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, o Amazon S3 usa 8 KB de armazenamento para o nome do objeto e outros metadados. O Amazon S3 armazena esses metadados de modo que você possa obter uma lista em tempo real de seus objetos arquivados usando a API do Amazon S3. Para obter mais informações, consulte [GET bucket \(listar objetos\)](#). Você é cobrado conforme as taxas do S3 Standard nesse armazenamento adicional.
 - Para cada objeto arquivado no S3 Glacier Flexible Retrieval ou no S3 Glacier Deep Archive, o Amazon S3 adiciona 32 KB de armazenamento para indexação e metadados relacionados. Esses dados extras são necessários para identificar e recuperar seu objeto. As taxas do S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive são cobradas por esse armazenamento adicional.

Se você estiver arquivando objetos pequenos, considere esses encargos de armazenamento. Considere também a possibilidade de agrregar vários objetos pequenos em um número menor de objetos grandes para reduzir os gastos adicionais.

- Número de dias que você pretende manter objetos arquivados: o S3 Glacier Flexible Retrieval e o S3 Glacier Deep Archive são soluções de arquivamento de longo prazo. O período mínimo de duração de armazenamento é de 90 dias para a classe de armazenamento S3 Glacier Flexible Retrieval e 180 dias para o S3 Glacier Deep Archive. A exclusão de dados arquivados no Amazon S3 Glacier não incorrerá em cobrança se os objetos que você excluir estiverem arquivados por mais do que o período mínimo de duração do armazenamento. Se você excluir ou substituir um objeto arquivado dentro do período mínimo de duração do arquivamento, o Amazon S3 cobrará uma taxa pro rata pela exclusão antecipada. Para obter informações sobre a taxa de exclusão antecipada, consulte a pergunta “Como é a cobrança pela exclusão de objetos do Amazon S3 Glacier com menos de 90 dias? de arquivamento?” nas [Perguntas frequentes sobre o Amazon S3](#).
- Cobranças de solicitações de transição do S3 Glacier Flexible Retrieval e do S3 Glacier Deep Archive: cada objeto que você transitar para a classe de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive constitui uma solicitação de transição. 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. Se você estiver arquivando objetos pequenos, considere a possibilidade de agrregar vários objetos pequenos em um número menor de objetos grandes para reduzir o custo da solicitação de transição.
- Cobranças de restauração de dados do S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive: o S3 Glacier Flexible Retrieval e o S3 Glacier Deep Archive foram projetados para arquivamento de dados de longo prazo que você acessa com pouca frequência. Para obter informações sobre cobranças de restauração de dados, consulte a pergunta “Quanto custa recuperar dados do Amazon S3 Glacier?”. nas [Perguntas frequentes sobre o Amazon S3](#). Para obter informações sobre como restaurar dados do Amazon S3 Glacier, consulte [Restaurar um objeto arquivado \(p. 715\)](#).

Quando você arquiva objetos no Amazon S3 Glacier usando o gerenciamento do S3 Lifecycle, o Amazon S3 faz a transição desses objetos de forma assíncrona. Pode haver um atraso entre a data de transição na regra de configuração do S3 Lifecycle e a data de transição física. Você é cobrado pelos preços do Amazon S3 Glacier com base na data de transição especificada na regra. Para obter mais informações, consulte a seção Amazon S3 Glacier das [Perguntas frequentes sobre o Amazon S3](#).

A página de detalhes de produto do Amazon S3 fornece informações sobre preços 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 do armazenamento de objetos do Amazon S3 arquivados no Amazon S3 Glacier?” nas [Perguntas frequentes sobre o Amazon S3](#).
- “Como é feita a cobrança pela exclusão de objetos do Amazon S3 Glacier com menos de 90 dias de arquivamento?” nas [Perguntas frequentes sobre o Amazon S3](#).
- “Quanto custa a recuperação de dados do Amazon S3 Glacier?” nas [Perguntas frequentes sobre o Amazon S3](#).

- [Definição de preço do Amazon S3](#) para custos de armazenamento das diferentes classes de armazenamento.

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 S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive. (Uma solicitação de operação [HEAD Object](#) ou [GET Object](#) da API retornará o S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive como a classe de armazenamento.)

Note

Ao restaurar um arquivo, você paga pelo arquivo (taxa do S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive) e por uma cópia que restaurou temporariamente (taxa de armazenamento de redundância reduzida). Para obter mais informações sobre definição de preços, consulte [Definição de preços do Amazon S3](#).

Você pode restaurar uma cópia do objeto programaticamente ou usando o console do Amazon S3. O Amazon S3 só processa uma solicitação de restauração por vez por objeto. Para obter mais informações, consulte [Restaurar um objeto arquivado \(p. 715\)](#).

Expirando objetos

Quando um objeto atinge o fim de seu ciclo de vida, com base em sua política de ciclo de vida, o Amazon S3 o coloca 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 pela validade ou tempo de armazenamento associado a um objeto que expirou.

Ao lidar com buckets que têm o versionamento habilitado, observe que será necessário criar uma transição para versões não atuais, bem como para evitar ser cobrados por objetos com versões não atuais. Para obter mais informações sobre transições de versões não atuais, consulte [Elementos de configuração do ciclo de vida \(p. 767\)](#).

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 você criar uma regra de expiração do S3 Lifecycle que fará com que os objetos que estiveram no armazenamento S3 Standard – IA ou S3 One Zone – IA por menos de 30 dias expirem, você será cobrado por 30 dias. Se criar uma regra de validade do Lifecycle que resulte na expiração de objetos armazenados na classe S3 Glacier Flexible Retrieval por pelo menos 90 dias, você será cobrado por 90 dias. Se você criar uma regra de expiração do ciclo de vida que faça com que os objetos que estiveram no armazenamento do S3 Glacier Deep Archive por menos de 180 dias expirem, será cobrado por 180 dias. Para obter mais informações, consulte a [Definição de preço do Amazon S3](#) e [Uso do console do S3 \(p. 754\)](#).

Definir a configuração do ciclo de vida em um bucket

Esta seção explica como você pode definir uma configuração do S3 Lifecycle em um bucket usando AWS SDKs, a AWS CLI ou o console do Amazon S3. Para obter informações sobre a configuração do S3 Lifecycle, consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

É possível usar regras de ciclo de vida para definir ações que você deseja que o Amazon S3 realize durante o ciclo de vida de um objeto (por exemplo, fazer a transição de objetos para outra classe de armazenamento, arquivá-los, ou excluí-los após um período especificado).

Antes de definir uma configuração de ciclo de vida, observe o seguinte:

Atraso de propagação

Quando você adiciona uma configuração do S3 Lifecycle 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 entre totalmente em vigor. Esse atraso também pode ocorrer quando você exclui uma configuração do S3 Lifecycle.

Desabilitando ou excluindo regras de ciclo de vida

Quando você desabilita ou exclui regras 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 foram agendados não são agendados e não são excluídos ou transicionados.

Objetos existentes e novos

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.

Alterações no faturamento

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.

Por exemplo, você não será cobrado pelo armazenamento após o tempo de expiração do objeto, mesmo que o objeto não seja excluído imediatamente. Outro exemplo é a cobrança das taxas de armazenamento do S3 Glacier Flexible Retrieval assim que o tempo de transição do objeto termina, mesmo que não seja feita a transição do objeto para a classe de armazenamento S3 Glacier Flexible Retrieval imediatamente. As transições do ciclo de vida para a classe de armazenamento S3 Intelligent-Tiering são a exceção. As alterações no faturamento não acontecem até que o objeto tenha transitado para a classe de armazenamento S3 Intelligent-Tiering.

Uso do console do S3

É possível definir uma regra de ciclo de vida para todos os objetos ou para um subconjunto de objetos no bucket usando um prefixo compartilhado (nomes de objetos que começam com uma string comum) ou uma tag. Usando uma regra de ciclo de vida, é possível definir ações específicas para versões atuais e anteriores do objeto. Para obter mais informações, consulte:

- [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#)
- [Usando o versionamento em buckets do S3 \(p. 678\)](#)

Como criar uma regra de ciclo de vida

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket para o qual você deseja criar uma regra de ciclo de vida.
3. Escolha a guia Management (Gerenciamento) e escolha Create lifecycle rule (Criar regra de ciclo de vida).

4. Em Lifecycle rule name (Nome da regra do ciclo de vida), insira um nome para sua regra.

O nome deve ser exclusivo dentro do bucket.

5. Escolha o escopo da regra do ciclo de vida:

- Para aplicar essa regra de ciclo de vida a todos os objetos com um prefixo ou uma tag específica, escolha Limitar o escopo a prefixos ou tags específicos.
 - Para limitar o escopo por prefixo, em Prefix (Prefixo), insira o prefixo.
 - Para limitar o escopo por tag, escolha Add tag (Adicionar tag) e insira a chave e o valor da tag.

Para obter mais informações sobre prefixes de nome de objeto, consulte [Criar nomes de chave de objeto \(p. 162\)](#). Para obter mais informações sobre tags de objeto, consulte [Categorizando seu armazenamento usando tags \(p. 878\)](#).

- Para aplicar essa regra de ciclo de vida a todos os objetos no bucket, escolha This rule applies to all objects in the bucket e escolha I acknowledge that this rule applies to all objects in the bucket (Eu reconheço que essa regra se aplica a todos os objetos no bucket).

6. Para filtrar uma regra por tamanho de objeto, você pode marcar Specify minimum object size (Especificar o tamanho mínimo de objeto), Specify maximum object size (Especificar o tamanho máximo do objeto) ou as duas opções.

- Quando você especificar um tamanho mínimo do objeto ou tamanho máximo do objeto, o valor deverá ser maior que 0 byte e até 5 TB. É possível especificar esse valor em bytes, KB, MB ou GB.
- Quando você especificar ambos, o tamanho máximo do objeto deverá ser maior que o tamanho mínimo do objeto.

7. Em Lifecycle rule actions (Ações de regra do ciclo de vida), escolha as ações que você deseja que sua regra de ciclo de vida execute:

- Transição de versões atuais de objetos entre classes de armazenamento
- Transição de versões anteriores de objetos entre classes de armazenamento
- Expirar as versões atuais dos objetos
- Excluir permanentemente versões anteriores de objetos
- Excluir marcadores de exclusão expirados ou multipart uploads incompletos

Dependendo das ações que você escolher, diferentes opções serão exibidas.

8. Para fazer a transição de versões atuais de objetos entre classes de armazenamento, em Transition current versions of objects between storage classes (Transição de versões atuais de objetos entre classes de armazenamento):

- a. Em Storage class transitions (Transições de classe de armazenamento), escolha a classe de armazenamento para a qual fazer a transição:
 - Standard-IA
 - Intelligent-Tiering
 - One Zone-IA
 - S3 Glacier Flexible Retrieval
 - Glacier Deep Archive
- b. Em Days after object creation (Dias após a criação do objeto), insira o número de dias após a criação para fazer a transição do objeto.

Para obter mais informações sobre classes de armazenamento, consulte [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#). É possível definir transições para a versão atual ou as versões anteriores do objeto, [versões antigas do S3](#). O armazenamento permite que você mantenha várias

versões de um objeto em um bucket. Para obter mais informações sobre versionamento, consulte [Uso do console do S3 \(p. 684\)](#).

Important

Quando você escolhe a classe de armazenamento S3 Glacier Flexible Retrieval ou Glacier Deep Archive, os objetos permanecem no Amazon S3. Não será possível acessá-los diretamente por meio do serviço separado do Amazon S3 Glacier. Para obter mais informações, consulte [Transição de objetos usando o Amazon S3 Lifecycle \(p. 748\)](#).

9. Para fazer a transição de versões não atuais de objetos entre classes de armazenamento, em Transition non-current versions of objects between storage classes (Transição de versões não atuais de objetos entre classes de armazenamento):
 - a. Em Storage class transitions (Transições de classe de armazenamento), escolha a classe de armazenamento para a qual fazer a transição:
 - Standard-IA
 - Intelligent-Tiering
 - One Zone-IA
 - S3 Glacier Flexible Retrieval
 - Glacier Deep Archive
 - b. Em Days after object becomes non-current (Dias após o objeto se tornar não atual), insira o número de dias após a criação para fazer a transição do objeto.
10. Para expirar as versões atuais dos objetos, em Expire previous versions of objects (Expirar versões anteriores de objetos), em Number of days after object creation (Número de dias após a criação do objeto), insira o número de dias.

Important

Em um bucket sem versionamento, a ação de expiração faz com que o Amazon S3 remova permanentemente o objeto. Para obter mais informações sobre ações de ciclo de vida, consulte [Elementos para descrever ações de ciclo de vida \(p. 771\)](#).

11. Para excluir permanentemente versões anteriores de objetos, em Permanently delete previous versions of objects (Excluir permanentemente versões desatualizadas de objetos), em Days after objects become noncurrent (Dias após os objetos ficarem desatualizados), insira o número de dias. Opcionalmente, você pode especificar o número de versões mais recentes a serem retidas inserindo um valor em Number of newer versions to retain (Número de versões mais novas a serem mantidas).
12. Em Delete expired delete markers or incomplete multipart uploads (Excluir marcadores de exclusão expirados ou multipart uploads incompletos), escolha Delete expired object delete markers (Excluir marcadores de exclusão de objetos expirados) e Delete incomplete multipart uploads (Excluir multipart uploads incompletos). Depois, insira o número de dias após o início do multipart upload que você deseja encerrar e limpar multipart uploads incompletos.

Para obter mais informações sobre multipart uploads, consulte [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).

13. Selecione Criar regra.

Se a regra não contiver erros, o Amazon S3 habilitará e você poderá vê-la na guia Management (Gerenciamento) em Lifecycle rules (Regras de ciclo de vida).

Para obter informações sobre modelos e exemplos do CloudFormation, consulte [Como trabalhar com modelos do AWS CloudFormation](#) e [AWS::S3::Bucket](#) no Guia do usuário do AWS CloudFormation.

Usar a AWS CLI

Use os comandos da AWS CLI a seguir para gerenciar as configurações do S3 Lifecycle:

- `put-bucket-lifecycle-configuration`
- `get-bucket-lifecycle-configuration`
- `delete-bucket-lifecycle`

Para obter instruções de configuração da AWS CLI, consulte [Desenvolvimento com o Amazon S3 usando a AWS CLI \(p. 1257\)](#).

A configuração do ciclo de vida do Amazon S3 é um arquivo XML. Mas ao usar a AWS CLI, você não pode especificar o XML. Você deve especificar o JSON em vez disso. Veja a seguir exemplos de configurações do ciclo de vida em XML e o JSON equivalente que pode ser especificado em um comando da AWS CLI.

Considere o seguinte exemplo de configuração do S3 Lifecycle.

Example Exemplo 1

JSON

```
{  
    "Rules": [  
        {  
            "Filter": {  
                "Prefix": "documents/"  
            },  
            "Status": "Enabled",  
            "Transitions": [  
                {  
                    "Days": 365,  
                    "StorageClass": "GLACIER"  
                }  
            ],  
            "Expiration": {  
                "Days": 3650  
            },  
            "ID": "ExampleRule"  
        }  
    ]  
}
```

XML

```
<LifecycleConfiguration>  
    <Rule>  
        <ID>ExampleRule</ID>  
        <Filter>  
            <Prefix>documents/</Prefix>  
        </Filter>  
        <Status>Enabled</Status>  
        <Transition>  
            <Days>365</Days>  
            <StorageClass>GLACIER</StorageClass>  
        </Transition>  
        <Expiration>  
            <Days>3650</Days>  
        </Expiration>  
    </Rule>  
</LifecycleConfiguration>
```

Example Exemplo 2

JSON

```
{  
    "Rules": [  
        {  
            "ID": "id-1",  
            "Filter": {  
                "And": {  
                    "Prefix": "myprefix",  
                    "Tags": [  
                        {  
                            "Value": "mytagvalue1",  
                            "Key": "mytagkey1"  
                        },  
                        {  
                            "Value": "mytagvalue2",  
                            "Key": "mytagkey2"  
                        }  
                    ]  
                }  
            },  
            "Status": "Enabled",  
            "Expiration": {  
                "Days": 1  
            }  
        }  
    ]  
}
```

XML

```
<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">  
    <Rule>  
        <ID>id-1</ID>  
        <Expiration>  
            <Days>1</Days>  
        </Expiration>  
        <Filter>  
            <And>  
                <Prefix>myprefix</Prefix>  
                <Tag>  
                    <Key>mytagkey1</Key>  
                    <Value>mytagvalue1</Value>  
                </Tag>  
                <Tag>  
                    <Key>mytagkey2</Key>  
                    <Value>mytagvalue2</Value>  
                </Tag>  
            </And>  
        </Filter>  
        <Status>Enabled</Status>  
    </Rule>  
</LifecycleConfiguration>
```

Teste o comando `put-bucket-lifecycle-configuration` da seguinte forma.

Como testar a configuração

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 S3 Lifecycle usando o comando `get-bucket-lifecycle-configuration` da AWS CLI da seguinte maneira:

```
$ aws s3api get-bucket-lifecycle-configuration \
--bucket bucketname
```

4. Para excluir a configuração do S3 Lifecycle, use o comando `delete-bucket-lifecycle` da AWS CLI da seguinte maneira:

```
aws s3api delete-bucket-lifecycle \
--bucket bucketname
```

Uso da SDKs AWS

Java

Use o AWS SDK for Java para gerenciar a configuração do S3 Lifecycle de um bucket. Para obter mais informações sobre o gerenciamento da configuração do ciclo de vida do S3, consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

Note

Quando você adiciona uma configuração do S3 Lifecycle 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 adicione a configuração revisada ao bucket.

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 modificada ao bucket. O Amazon S3 substitui a configuração 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 exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketLifecycleConfiguration;
import com.amazonaws.services.s3.model.BucketLifecycleConfiguration.Transition;
import com.amazonaws.services.s3.model.StorageClass;
import com.amazonaws.services.s3.model.Tag;
```

```
import com.amazonaws.services.s3.model.lifecycle.LifecycleAndOperator;
import com.amazonaws.services.s3.model.lifecycle.LifecycleFilter;
import com.amazonaws.services.s3.model.lifecycle.LifecyclePrefixPredicate;
import com.amazonaws.services.s3.model.lifecycle.LifecycleTagPredicate;

import java.io.IOException;
import java.util.Arrays;

public class LifecycleConfiguration {

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

        // Create a rule to archive objects with the "glacierobjects/" prefix to
        // Glacier immediately.
        BucketLifecycleConfiguration.Rule rule1 = new
        BucketLifecycleConfiguration.Rule()
            .withId("Archive immediately rule")
            .withFilter(new LifecycleFilter(new
        LifecyclePrefixPredicate("glacierobjects/")))
            .addTransition(new
        Transition().withDays(0).withStorageClass(StorageClass.Glacier))
            .withStatus(BucketLifecycleConfiguration.ENABLED);

        // Create a rule to transition objects to the Standard-Infrequent Access
        storage class
        // after 30 days, then to Glacier after 365 days. Amazon S3 will delete the
        objects after 3650 days.
        // The rule applies to all objects with the tag "archive" set to "true".
        BucketLifecycleConfiguration.Rule rule2 = new
        BucketLifecycleConfiguration.Rule()
            .withId("Archive and then delete rule")
            .withFilter(new LifecycleFilter(new LifecycleTagPredicate(new
        Tag("archive", "true"))))
            .addTransition(new
        Transition().withDays(30).withStorageClass(StorageClass.StandardInfrequentAccess))
            .addTransition(new
        Transition().withDays(365).withStorageClass(StorageClass.Glacier))
            .withExpirationInDays(3650)
            .withStatus(BucketLifecycleConfiguration.ENABLED);

        // Add the rules to a new BucketLifecycleConfiguration.
        BucketLifecycleConfiguration configuration = new BucketLifecycleConfiguration()
            .withRules(Arrays.asList(rule1, rule2));

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

            // Save the configuration.
            s3Client.setBucketLifecycleConfiguration(bucketName, configuration);

            // Retrieve the configuration.
            configuration = s3Client.getBucketLifecycleConfiguration(bucketName);

            // Add a new rule with both a prefix predicate and a tag predicate.
            configuration.getRules().add(new
            BucketLifecycleConfiguration.Rule().withId("NewRule")
                .withFilter(new LifecycleFilter(new LifecycleAndOperator(
                    Arrays.asList(new
                    LifecyclePrefixPredicate("YearlyDocuments/"),
                    new LifecycleTagPredicate(new Tag("expire_after",
                    "ten_years")))))))

        } catch (AmazonServiceException e) {
            System.out.println("Caught an AmazonServiceException, which means your request failed");
            System.out.println("HTTP Status Code: " + e.getStatusCode());
            System.out.println("Error Code: " + e.getErrorCode());
            System.out.println("Error Message: " + e.getMessage());
            System.out.println("Request ID: " + e.getRequestId());
        }
    }
}
```

```
.withExpirationInDays(3650)
.withStatus(BucketLifecycleConfiguration.ENABLED));

// Save the configuration.
s3Client.setBucketLifecycleConfiguration(bucketName, configuration);

// Retrieve the configuration.
configuration = s3Client.getBucketLifecycleConfiguration(bucketName);

// Verify that the configuration now has three rules.
configuration = s3Client.getBucketLifecycleConfiguration(bucketName);
System.out.println("Expected # of rules = 3; found: " +
configuration.getRules().size());

// Delete the configuration.
s3Client.deleteBucketLifecycleConfiguration(bucketName);

// Verify that the configuration has been deleted by attempting to retrieve
it.
configuration = s3Client.getBucketLifecycleConfiguration(bucketName);
String s = (configuration == null) ? "No configuration found." :
"Configuration found.";
System.out.println(s);
} catch (AmazonServiceException e) {
// The call was transmitted successfully, but Amazon S3 couldn't process
// it, so it returned an error response.
e.printStackTrace();
} catch (SdkClientException e) {
// Amazon S3 couldn't be contacted for a response, or the client
// couldn't parse the response from Amazon S3.
e.printStackTrace();
}
}
}
```

.NET

Use o AWS SDK for .NET para gerenciar a configuração do S3 Lifecycle em um bucket. Para obter mais informações sobre o gerenciamento da configuração do ciclo de vida, consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

Note

Quando você adiciona uma configuração do ciclo de vida, o Amazon S3 substitui a configuração existente no bucket especificado. Para atualizar uma configuração, recupere a configuração do ciclo de vida, faça as alterações e, depois, adicione a configuração revisada ao bucket.

O exemplo a seguir mostra como usar o AWS SDK for .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 modificada 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 Lifecycle e verifica a exclusão.

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

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

        private static async Task AddUpdateDeleteLifecycleConfigAsync()
        {
            try
            {
                var lifeCycleConfiguration = new LifecycleConfiguration()
                {
                    Rules = new List<LifecycleRule>
                    {
                        new LifecycleRule
                        {
                            Id = "Archive immedately 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);
}
}
```

Ruby

Você pode usar o AWS SDK for Ruby para gerenciar a configuração do S3 Lifecycle em um bucket com a classe [AWS::S3::BucketLifecycleConfiguration](#). Para obter mais informações sobre como usar o AWS SDK for Ruby com o Amazon S3, consulte [Usar o AWS SDK for Ruby - versão 3 \(p. 1269\)](#). Para obter mais informações sobre o gerenciamento da configuração do ciclo de vida, consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

Uso dos REST API

As seções a seguir na Referência da API do Amazon Simple Storage Service descrevem a API REST relacionada à configuração do S3 Lifecycle.

- [Ciclo de vida de PUT Bucket](#)
- [Ciclo de vida de GET Bucket](#)
- [DELETE Bucket lifecycle](#)

Ciclo de vida e outras configurações de bucket

Além de configurações do S3 Lifecycle, você pode associar outras configurações a seu bucket. Esta seção explica como a configuração do S3 Lifecycle está relacionada a outras configurações de bucket.

Ciclo de vida e versionamento

Você pode adicionar configurações de ciclo de vida do S3 a buckets com e sem versionamento. Para obter mais informações, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

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 \(p. 767\)](#).

Configuração do ciclo de vida em buckets com MFA habilitado

A configuração do ciclo de vida em buckets habilitados para autenticação multifator (MFA) não é suportada.

Ciclo de vida e registro em log

As ações de ciclo de vida do Amazon S3 não são capturadas pelo log no nível do objeto do AWS CloudTrail. O CloudTrail captura solicitações de API feitas para endpoints externos do Amazon S3, enquanto as ações de ciclo de vida do S3 são executadas usando endpoints internos do Amazon S3. Os logs de acesso ao servidor do Amazon S3 podem ser habilitados em um bucket do S3 para capturar ações relacionadas ao ciclo de vida do S3, como transição de objetos para outra classe de armazenamento e expiração de objeto, resultando em exclusão permanente ou exclusão lógica. Para obter mais informações, consulte [the section called “Registrando acesso ao servidor” \(p. 1041\)](#).

Se o registro em logs estiver ativado em seu bucket, os logs de acesso do servidor do Amazon S3 relatarão os resultados das seguintes operações.

Log de operação	Descrição
S3.EXPIRE.OBJECT	O Amazon S3 exclui permanentemente o objeto devido à ação de expiração de ciclo de vida.
S3.CREATE.DELETEMARKER	O Amazon S3 exclui logicamente a versão atual e adiciona o marcador de exclusão em um bucket com habilitação para versionamento.
S3.TRANSITION_SIA.OBJECT	O Amazon S3 faz a transição do objeto para a classe de armazenamento S3 Standard – IA.
S3.TRANSITION_ZIA.OBJECT	O Amazon S3 faz a transição do objeto para a classe de armazenamento S3 One Zone – IA.
S3.TRANSITION_INT.OBJECT	O Amazon S3 faz a transição do objeto para a classe de armazenamento S3 Intelligent-Tiering.
S3.TRANSITION_GIR.OBJECT	O Amazon S3 inicia a transição do objeto para a classe de armazenamento S3 Glacier Instant Retrieval.

Log de operação	Descrição
S3.TRANSITION.OBJECT	O Amazon S3 inicia a transição do objeto para a classe de armazenamento S3 Glacier Flexible Retrieval.
S3.TRANSITION_GDA.OBJECT	O Amazon S3 inicia a transição do objeto para a classe de armazenamento S3 Glacier Deep Archive.
S3.DELETE.UPLOAD	O Amazon S3 aborta o multipart upload incompleto.

Note

Os registros em log de acesso ao servidor do Amazon S3 geralmente são entregues com base no melhor esforço, não podendo ser usados para uma contabilização completa de todas as solicitações do Amazon S3.

Mais informações

- [Elementos de configuração do ciclo de vida \(p. 767\)](#)
- [Transição para as classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive \(arquivamento de objetos\) \(p. 750\)](#)
- [Definir a configuração do ciclo de vida em um bucket \(p. 753\)](#)

Configurar notificações de evento de ciclo de vida

É possível configurar uma notificação de evento do Amazon S3 para receber aviso quando o Amazon S3 excluir um objeto ou fizer a transição para outra classe de armazenamento do Amazon S3 seguindo uma regra do S3 Lifecycle.

Ao usar os tipos de evento LifecycleExpiration, você poderá receber uma notificação sempre que o Amazon S3 excluir um objeto com base na configuração do S3 Lifecycle. O tipo de evento s3:LifecycleExpiration:Delete notifica você quando um objeto em um bucket sem versionamento é excluído. Ele também notifica quando uma versão de objeto é excluída permanentemente por uma configuração do S3 Lifecycle. O tipo de evento s3:LifecycleExpiration:DeleteMarkerCreated notifica você quando o S3 Lifecycle cria um marcador de exclusão quando uma versão atual de um objeto no bucket com versionamento é excluída. Para obter mais informações, consulte [Versão do objeto Delete](#).

Ao usar o tipo de evento s3:LifecycleTransition, você poderá receber essa notificação quando um objeto for transferido para outra classe de armazenamento do Amazon S3 por uma configuração do S3 Lifecycle.

O Amazon S3 pode publicar notificações de eventos em um tópico do Amazon Simple Notification Service (Amazon SNS), em uma fila do Amazon Simple Queue Service (Amazon SQS) ou em uma função do AWS Lambda. Para obter mais informações, consulte [Notificações de eventos do Amazon S3 \(p. 1083\)](#).

Para obter instruções sobre como configurar as notificações de evento do Amazon S3, consulte [Habilitar notificações de eventos](#).

Veja a seguir um exemplo de mensagem que o Amazon S3 envia para publicar um evento s3:LifecycleExpiration:Delete. Para obter mais informações, consulte [Estrutura de mensagens de evento](#).

```
{  
    "Records": [  
        {  
            "eventVersion": "2.3",  
            "userIdentity": {  
                "principal": "arn:aws:iam::123456789012:root",  
                "type": "AWSIdentity"  
            },  
            "approximateCreationDate": 1516229190000,  
            "s3": {  
                "objectKey": "image1.jpg",  
                "bucket": "mybucket",  
                "versionId": "V1234567890123456789012345678901",  
                "size": 1024000,  
                "region": "us-east-1",  
                "account": "123456789012",  
                "s3SchemaVersion": "1.0",  
                "storageClass": "STANDARD",  
                "encryptionType": "NONE",  
                "sseAlgorithm": null,  
                "sequencer": "00000000000000000001",  
                "configurationId": "myconfig",  
                "objectLockEnabled": false,  
                "objectLockLegalHoldStatus": null  
            }  
        }  
    ]  
}
```

```
"eventSource": "aws:s3",
"awsRegion": "us-west-2",
"eventTime": "1970-01-01T00:00:00.000Z",
"eventName": "LifecycleExpiration:Delete",
"userIdentity": {
    "principalId": "s3.amazonaws.com"
},
"requestParameters": {
    "sourceIPAddress": "s3.amazonaws.com"
},
"responseElements": {
    "x-amz-request-id": "C3D13FE58DE4C810",
    "x-amz-id-2": "FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/JRWeUWerMUE5JgHvANOjpD"
},
"s3": {
    "s3SchemaVersion": "1.0",
    "configurationId": "testConfigRule",
    "bucket": {
        "name": "mybucket",
        "ownerIdentity": {
            "principalId": "A3NL1KOZZKExample"
        },
        "arn": "arn:aws:s3:::mybucket"
    },
    "object": {
        "key": "expiration/delete",
        "sequencer": "0055AED6DCD90281E5"
    }
}
]
```

As mensagens que o Amazon S3 envia para publicar um evento s3:LifecycleTransition também incluem as informações a seguir.

```
"lifecycleEventData": {
    "transitionEventData": {
        "destinationStorageClass": the destination storage class for the object
    }
}
```

Elementos de configuração do ciclo de vida

Tópicos

- [Elemento ID \(p. 768\)](#)
- [Elemento Status \(p. 768\)](#)
- [Elemento Filter \(p. 768\)](#)
- [Elementos para descrever ações de ciclo de vida \(p. 771\)](#)

Especifique uma configuração de ciclo de vida do S3 como XML, consistindo em uma ou mais regras de ciclo de vida.

```
<LifecycleConfiguration>
    <Rule>
        ...
    </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 das chaves de objeto, uma ou mais etiquetas de objeto ou uma combinação de filtros.
- 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 do S3. Para obter configurações de exemplo, consulte [Exemplos de configuração do S3 Lifecycle \(p. 775\)](#).

Elemento ID

Uma configuração do ciclo de vida do S3 pode ter até 1.000 regras. Este limite não é ajustável. 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 das chaves, por tag de objeto ou por uma combinação dos dois (neste último caso, o Amazon S3 usa uma operação AND lógica 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 do S3 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 [Criar nomes de chave de objeto \(p. 162\)](#).

- Especificação um filtro com base em tags de objeto: no exemplo a seguir, a regra de ciclo de vida especifica um filtro com base em uma tag (**chave**) e 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 fazer uma quebra de etiquetas 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 ambas as tags especificadas. O Amazon S3 realiza uma operação AND lógica. Observe o seguinte:

- Cada tag deve corresponder exatamente à chave e ao valor.

- A regra se aplica a um subconjunto de objetos que tem todas as tags especificadas na regra. Se um objeto tiver tags adicionais especificadas, a regra ainda será aplicada.

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 quebrar 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 uma operação AND lógica. 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 aplicará a todos os objetos do bucket.

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
    </Filter>
    <Status>Enabled</Status>
    transition/expiration actions.
  </Rule>
</LifecycleConfiguration>
```

- Para filtrar uma regra por tamanho do objeto, você pode especificar um tamanho mínimo (ObjectSizeGreaterThan), um tamanho máximo (ObjectSizeLessThan) ou um intervalo de tamanhos de objeto.

Os valores de tamanho de objeto estão em bytes. O tamanho máximo do filtro é de 5 TB. Algumas classes de armazenamento têm limitações mínimas de tamanho de objeto. Para obter mais informações, consulte [Comparar as classes de armazenamento do Amazon S3 \(p. 735\)](#).

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
      <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
    </Filter>
    <Status>Enabled</Status>
    transition/expiration actions.
  </Rule>
</LifecycleConfiguration>
```

```
</Rule>  
</LifecycleConfiguration>
```

Se você estiver especificando um intervalo de tamanho de objeto, o número inteiro `ObjectSizeGreater Than` deve ser menor que o valor de `ObjectSizeLess Than`. Ao usar mais de um filtro, é necessário fazer a quebra dos filtros em um elemento `<And>`. O exemplo a seguir mostra como especificar objetos em um intervalo entre 500 e 64000 bytes.

```
<LifecycleConfiguration>  
  <Rule>  
    <Filter>  
      <And>  
        <Prefix>key-prefix</Prefix>  
        <ObjectSizeGreater Than>500</ObjectSizeGreater Than>  
        <ObjectSizeLess Than>64000</ObjectSizeLess Than>  
      </And>  
    </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 do S3. 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 \(p. 748\)](#). 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 desatualizadas, o Amazon S3 define a ação `NoncurrentVersionTransition` (descrita posteriormente neste tópico).

- Elemento de ação de expiração: a ação `Expiration` expira 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 [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#). O Amazon S3 deixa 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 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.

- 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 [Usando o versionamento em buckets do S3 \(p. 678\)](#). 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 do 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 do objeto expirado. Para ver um exemplo, consulte [Exemplo 7: Remover marcadores de exclusão de objetos expirados \(p. 783\)](#).

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 esta ação para especificar quando fazer com que o Amazon S3 execute a transição de objetos para a classe de armazenamento especificada. Essa expiração pode se basear em um determinado número de dias desde o momento em que os objetos se tornam não atuais. Além do número de dias, você também pode fornecer um número máximo de versões não atuais para reter. Você também precisa fornecer um elemento `Filter` para especificar o número máximo de versões não atuais. Se não especificar um elemento `Filter`, o Simple Storage Service (Amazon S3) gerará um erro `InvalidRequest` quando você fornecer um número máximo de versões não atuais.

Para obter mais informações sobre transição de objetos, consulte [Transições com suporte e limitações relacionadas \(p. 748\)](#). Para obter detalhes sobre como o Amazon S3 calcula a data para especificar o número de dias na ação `NoncurrentVersionTransition`, consulte [Regras de ciclo de vida: com base na idade de um objeto \(p. 774\)](#).

- Elemento de ação NoncurrentVersionExpiration: use esta ação para especificar quando o Amazon S3 deve remover permanentemente objetos desatualizados. Esses objetos excluídos não poderão ser recuperados. Essa expiração pode se basear em um determinado número de dias desde o momento em que os objetos se tornam não atuais. Além do número de dias, você também pode fornecer um número máximo de versões não atuais para reter. Você também precisa fornecer um elemento `Filter` para especificar o número máximo de versões não atuais. Se não especificar um elemento `Filter`, o Simple Storage Service (Amazon S3) gerará um erro `InvalidRequest` quando você fornecer um número máximo de versões não atuais.

A remoção atrasada de objetos não atuais pode ser útil quando é necessário 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 denominado `photo.gif` (ID 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.

Para obter detalhes sobre como o Amazon S3 calcula a data para especificar o número de dias em uma ação `NoncurrentVersionExpiration`, consulte [Regras de ciclo de vida: com base na idade de um objeto \(p. 774\)](#).

Important

As políticas de ciclo de vida de validade do objeto não removem carregamentos fracionados 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 parar 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 parará os multipart uploads incompletos. Para obter mais informações, consulte [Abortar um multipart upload \(p. 206\)](#).

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: Remover marcadores de exclusão de objetos expirados \(p. 783\)](#).

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 habilitado, é possível ter várias versões de um objeto. Há sempre uma versão atual e zero 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 que um objeto fica desatualizado, o Amazon S3 observa quando o sucessor foi criado. O Amazon S3 usa o número de dias desde que o sucessor foi criado como o número de dias que um objeto fica desatualizado.

Restaurar versões anteriores de um objeto ao usar configurações do S3 Lifecycle

Como explicado em detalhes no tópico [Restaurar versões anteriores \(p. 698\)](#), 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.

Quando você estiver usando regras de configuração do ciclo de vida do S3 com buckets habilitados para versionamento, recomendamos como prática recomendada que você use o primeiro método.

O S3 Lifecycle opera sob um modelo de consistência final. Uma versão atual que você excluiu permanentemente pode não desaparecer até que as alterações sejam propagadas (talvez o Amazon S3 não esteja ciente dessa exclusão). Entretanto, a regra de ciclo de vida configurada para a validade de objetos desatualizados pode remover permanentemente objetos desatualizados, inclusive 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 do S3, 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:

- É o número de dias a partir do momento em que a versão do objeto se torna desatualizada (ou seja, quando o objeto é sobreescrito ou excluído) que o Amazon S3 executará a ação no objeto ou objetos especificados.
- 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 do S3, é possível especificar uma data em que deseja que o Amazon S3 realize a ação. Quando a data especificada chegar, o Amazon S3 aplicará a ação a todos os objetos qualificados (com base nos critérios de filtro).

Se você especificar uma ação de ciclo de vida do S3 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 Amazon S3 continuará aplicando a ação com base em data mesmo após a data ter passado, contanto que o status da regra seja `Enabled`.

Por exemplo, imagine que você especifique uma ação de `Expiration` com base em data para excluir todos os objetos (supondo que nenhum filtro seja especificado na regra). Na data especificada, o Amazon 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, desativar ou excluir essas regras.

Exemplos de configuração do S3 Lifecycle

Esta seção fornece exemplos de configuração do ciclo de vida do S3. Cada exemplo mostra como você pode especificar o XML em cada um dos cenários de exemplo.

Tópicos

- [Exemplo 1: Especificar um filtro \(p. 775\)](#)
- [Exemplo 2: desabilitar uma regra do Lifecycle \(p. 777\)](#)
- [Exemplo 3: Rebaixar uma classe de armazenamento pela duração do ciclo de vida de um objeto \(p. 778\)](#)
- [Exemplo 4: Especificar várias regras \(p. 778\)](#)
- [Exemplo 5: sobreposição de filtros, ações de ciclo de vida conflitantes e o que o Amazon S3 faz com buckets sem versionamento \(p. 779\)](#)
- [Exemplo 6: Especificar uma regra de ciclo de vida para um bucket com versionamento habilitado \(p. 782\)](#)
- [Exemplo 7: Remover marcadores de exclusão de objetos expirados \(p. 783\)](#)
- [Exemplo 8: Configuração de ciclo de vida para anular multipart uploads \(p. 785\)](#)
- [Exemplo 9: configuração do ciclo de vida usando regras baseadas em tamanho \(p. 786\)](#)

Exemplo 1: Especificar um filtro

Cada regra do S3 Lifecycle inclui um filtro que pode ser usado para identificar um subconjunto de objetos em seu bucket ao qual a regra do S3 Lifecycle se aplica. As configurações de ciclo de vida do S3 a seguir mostram exemplos de como você pode especificar um filtro.

- Nesta regra de configuração do S3 Lifecycle, o filtro especifica um prefixo das chaves (`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 S3 Glacier Flexible Retrieval 365 dias (um ano) após a criação.
- Exclua objetos (a ação `Expiration`) 3.650 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>S3 Glacier Flexible Retrieval</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 do S3 Lifecycle se aplique a todos os objetos no bucket, especifique um prefixo vazio. Na configuração a seguir, a regra especifica uma ação Transition que direciona o Amazon S3 para fazer a transição de objetos para a classe de armazenamento S3 Glacier Flexible Retrieval 0 dia após a criação. Essa regra significa que os objetos estão qualificados para arquivamento no Amazon S3 Glacier à meia-noite (UTC) após a criação. Para obter mais informações sobre restrições de ciclo de vida, consulte [Restrições \(p. 749\)](#).

```
<LifecycleConfiguration>
  <Rule>
    <ID>Archive all object same-day upon creation</ID>
    <Filter>
      <Prefix></Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>S3 Glacier Flexible Retrieval</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 do S3 Lifecycle a um subconjunto de objetos com prefixo das chaves tax/ e a objetos que têm duas etiquetas com uma chave e um valor específicos. Ao especificar mais de um filtro, você deve incluir o elemento <And> conforme mostrado (o Amazon S3 aplica um elemento 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 do S3 Lifecycle a seguir aplica-se a objetos que tenham duas etiquetas especificadas (não especifica nenhum prefixo).

```
...
```

```
<Filter>
  <And>
    <Tag>
      <Key>key1</Key>
      <Value>value1</Value>
    </Tag>
    <Tag>
      <Key>key2</Key>
      <Value>value2</Value>
    </Tag>
  </And>
</Filter>
...
...
```

Important

Quando você tem várias regras em uma configuração do S3 Lifecycle, um objeto pode se tornar qualificado para várias ações do S3 Lifecycle. Nesses casos, o Amazon S3 segue estas regras gerais:

- 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 S3 Glacier Flexible Retrieval e S3 Standard – IA (ou S3 One Zone – IA), o Amazon S3 escolhe a transição do S3 Glacier Flexible Retrieval.

Para ver exemplos, consulte [Exemplo 5: sobreposição de filtros, ações de ciclo de vida conflitantes e o que o Amazon S3 faz com buckets sem versionamento \(p. 779\)](#).

Exemplo 2: desabilitar uma regra do Lifecycle

Você pode desabilitar temporariamente uma regra do S3 Lifecycle. A configuração do S3 Lifecycle 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 S3 Glacier Flexible Retrieval logo após a criação.
- Na regra 2, o Amazon S3 faz a transição de objetos com o prefixo documents/ para a classe de armazenamento S3 Glacier Flexible Retrieval logo após a criação.

Na política, a regra 1 é habilitada e a regra 2 é desabilitada. O Amazon S3 ignora regras desabilitadas.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule1</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>S3 Glacier Flexible Retrieval</StorageClass>
    </Transition>
  </Rule>
  <Rule>
```

```
<ID>Rule2</ID>
<Filter>
  <Prefix>documents/</Prefix>
</Filter>
<Status>Disabled</Status>
<Transition>
  <Days>0</Days>
  <StorageClass>S3 Glacier Flexible Retrieval</StorageClass>
</Transition>
</Rule>
</LifecycleConfiguration>
```

Exemplo 3: Rebaixar uma classe de armazenamento pela duração do ciclo de vida de um objeto

Neste exemplo, você usa a configuração do S3 Lifecycle 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 a definição de preços, consulte [Definição de preços do Amazon S3](#).

A configuração do S3 Lifecycle 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:
 - Transição de objetos para a classe de armazenamento S3 Standard – IA 30 dias após a criação.
 - Faça a transição de objetos para a classe de armazenamento S3 Glacier Flexible Retrieval 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 do S3 Lifecycle 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: Especificar várias regras

Você pode especificar várias regras se quiser diferentes ações do S3 Lifecycle de diferentes objetos. A configuração do S3 Lifecycle 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 S3 Glacier Flexible Retrieval um ano após a criação e expira esses objetos dez anos após a criação.
- A regra 2 se aplica a objetos com prefixo de nome de chave `classB/`. Ela direciona o Amazon S3 para fazer a transição de objetos para a classe de armazenamento S3 Standard – IA 90 dias após a criação e exclui-los 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 com buckets sem versionamento

Você pode especificar uma configuração de ciclo de vida do S3 em que especifica prefixos ou ações sobrepostos.

Geralmente, o S3 Lifecycle otimiza o custo. Por exemplo, se duas políticas de expiração se sobreponerem, a mais curta será honrada para que os dados não sejam armazenados por mais tempo do que o esperado. Do mesmo modo, se duas políticas de transição se sobreponerem, o ciclo de vida do S3 irá fazer a transição dos seus objetos para a classe de armazenamento de menor custo.

Nos dois casos, o ciclo de vida do S3 tenta escolher o caminho com menos custo para você. Uma exceção a essa regra geral é com a classe de armazenamento S3 Intelligent-Tiering. O S3 Intelligent-Tiering é favorecido pelo ciclo de vida do S3 em qualquer classe de armazenamento, além das classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive.

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. A Regra 2 solicita que o Amazon S3 faça a transição de um subconjunto de objetos para a classe de armazenamento do S3 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>
```

Como não há conflito nesse caso, o Amazon S3 fará a transição dos objetos com o prefixo `logs/` para a classe de armazenamento S3 Standard – IA 30 dias após a criação. Ao atingir um ano após a criação, qualquer objeto será excluído.

Example 2: ações de ciclo de vida conflitantes

Nesta 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 S3 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 percam a validade (sejam removidos), não faz sentido mudar a classe de armazenamento, então o Amazon S3 simplesmente escolhe a ação de validade desses objetos.

Example 3: prefixos sobrepostos que resultam em ações de ciclo de vida conflitantes

Neste exemplo, a configuração tem duas regras, que especificam prefixos 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 do S3 Lifecycle em ambas as regras se aplicam. Uma regra direciona o Amazon S3 a fazer a transição de objetos dez dias após a criação, e outra regra direciona o Amazon S3 a fazer 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 do S3 que tem duas regras, cada uma especificando um filtro de tag:

- A regra 1 especifica um filtro baseado em tag (tag1/value1). Essa regra direciona o Amazon S3 para fazer a transição de objetos para a classe de armazenamento S3 Glacier Flexible Retrieval 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 do S3 Lifecycle é apresentada como a seguir.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
      <Tag>
        <Key>tag1</Key>
        <Value>value1</Value>
      </Tag>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>GLACIER<StorageClass>
      <Days>365</Days>
    </Transition>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Tag>
        <Key>tag2</Key>
        <Value>value2</Value>
      </Tag>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>14</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

Se um objeto tiver as duas etiquetas, o Amazon S3 precisará decidir qual regra seguir. Nesse caso, o Amazon S3 expira o objeto 14 dias após a criação. O objeto é removido e, portanto, a ação de transição não é aplicada.

Exemplo 6: Especificar uma regra de ciclo de vida para um bucket com versionamento habilitado

Suponha que você tenha um bucket com versionamento habilitado, ou seja, para cada objeto, há uma versão atual e zero ou mais versões desatualizadas. (Para obter mais informações sobre o versionamento do S3, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).) Neste exemplo, você deseja manter o equivalente a um ano de histórico e excluir as versões desatualizadas. As configurações do S3 Lifecycle permitem manter de 1 a 100 versões de qualquer objeto.

Para economizar em custos de armazenamento, você deseja migrar as versões desatualizadas para o S3 Glacier Flexible Retrieval 30 dias depois de se tornarem desatualizadas (supondo que sejam dados frios,

que você não precisa acessar em tempo real). Além disso, você espera que a frequência de acesso das versões atuais diminua 90 dias após a criação, podendo optar por mover esses objetos para a classe de armazenamento S3 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>S3 Glacier Flexible Retrieval</StorageClass>
    </NoncurrentVersionTransition>
    <NoncurrentVersionExpiration>
      <NewerNoncurrentVersions>5</NewerNoncurrentVersions>
      <NoncurrentDays>365</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

Exemplo 7: Remover 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 do S3 para que o Amazon S3 remova os marcadores de exclusão de objetos expirados.

Suponha que você elabore uma política do S3 Lifecycle que use a ação `NoncurrentVersionExpiration` para remover as versões desatualizadas 30 dias após se tornarem desatualizadas e retenha até dez versões desatualizadas, como mostrado a seguir.

```
<LifecycleConfiguration>
  <Rule>
    ...
    <NoncurrentVersionExpiration>
      <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
      <NoncurrentDays>30</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

A ação `NoncurrentVersionExpiration` não se aplica às versões de objeto atuais. Ela remove apenas versões desatualizadas.

Para versões atuais do objeto, existem as opções abaixo para gerenciar o ciclo de vida. Tudo vai depender de as versões atuais seguirem 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 do S3 Lifecycle com a ação `Expiration` para que o Amazon S3 remova as versões atuais, conforme exibido no exemplo a seguir.

```
<LifecycleConfiguration>
  <Rule>
    ...
    <Expiration>
      <Days>60</Days>
    </Expiration>
    <NoncurrentVersionExpiration>
      <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
      <NoncurrentDays>30</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

Neste exemplo, 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. Esse processo torna a versão atual do objeto desatualizada, e o marcador de exclusão torna-se a versão atual. Para mais informações, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

Note

Não é possível especificar uma etiqueta `Days` e uma `ExpiredObjectDeleteMarker` na mesma regra. Com a especificação da etiqueta `Days`, o Amazon S3 executará automaticamente a limpeza `ExpiredObjectDeleteMarker` assim que os marcadores de exclusão tiverem idade suficiente para satisfazer os critérios de idade. Para limpar marcadores de exclusão assim que eles se tornarem a única versão, crie uma regra separada contendo apenas a etiqueta `ExpiredObjectDeleteMarker`.

A ação `NoncurrentVersionExpiration` na mesma configuração do S3 Lifecycle remove os objetos desatualizados 30 dias após se tornarem desatualizados. Assim, neste exemplo, todas as versões de objeto são permanentemente removidas 90 dias após a criação do objeto. Embora marcadores de exclusão de objetos expirados sejam criados durante esse o processo, o Amazon S3 detecta e remove os marcadores de exclusão de objetos expirados para você.

- As versões atuais do objeto não têm um ciclo de vida bem definido.

Neste caso, convém remover os objetos manualmente quando não forem mais necessários criando um marcador de exclusão com uma ou mais versões desatualizadas. Se a configuração do S3 Lifecycle com a ação `NoncurrentVersionExpiration` remover todas as versões desatualizadas, agora você terá marcadores de exclusão de objetos expirados.

Especificamente para este cenário, a configuração do S3 Lifecycle fornece uma ação `Expiration` que você pode usar para remover 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>
    <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
    <NoncurrentDays>30</NoncurrentDays>
</NoncurrentVersionExpiration>
</Rule>
</LifecycleConfiguration>
```

Ao definir o elemento `ExpiredObjectDeleteMarker` como `true` na ação `Expiration`, você faz o Amazon S3 remover os marcadores de exclusão de objetos expirados.

Note

Ao usar a ação `ExpiredObjectDeleteMarker` do S3 Lifecycle, a regra não pode especificar um filtro baseado em etiqueta.

Exemplo 8: Configuração de ciclo de vida para anular multipart uploads

Você pode usar as operações de API REST de carregamento fracionado do Amazon S3 para carregar objetos grandes em partes. Para obter mais informações sobre multipart uploads, consulte [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).

Usando a configuração do S3 Lifecycle, você pode levar o Amazon S3 a interromper carregamentos fracionados incompletos (identificados pelo prefixo de nome de chave especificado na regra) se eles não forem concluídos em um número especificado de dias após a inicialização. Ao interromper um carregamento fracionado, o Amazon S3 exclui todas as partes associadas ao carregamento fracionado. Esse processo ajuda a controlar os custos de armazenamento, garantindo que você não tenha carregamentos fracionados incompletos com partes que estão armazenadas no Amazon S3.

Note

Ao usar a ação `AbortIncompleteMultipartUpload` do S3 Lifecycle, a regra não pode especificar um filtro baseado em etiqueta.

Veja a seguir um exemplo de configuração de ciclo de vida do S3 que especifica uma regra com a ação `AbortIncompleteMultipartUpload`. Essa ação solicita que o Amazon S3 interrompa carregamentos fracionados 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>
```

Exemplo 9: configuração do ciclo de vida usando regras baseadas em tamanho

É possível criar regras que fazem a transição de objetos com base somente no tamanho. Você pode especificar um tamanho mínimo (`ObjectSizeGreater Than`), um tamanho máximo (`ObjectSizeLess Than`) ou um intervalo de tamanhos de objeto em bytes. Ao usar mais de um filtro, como prefixo e regra de tamanho, é necessário quebrar os filtros em um elemento `<And>`.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Transition with a prefix and based on size</ID>
    <Filter>
      <And>
        <Prefix>tax/</Prefix>
        <ObjectSizeGreater Than>500</ObjectSizeGreater Than>
      </And>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>S3 Glacier Flexible Retrieval</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Se você especificar um intervalo usando os elementos `ObjectSizeGreater Than` e `ObjectSizeLess Than`, o tamanho máximo do objeto deverá ser maior que o tamanho mínimo do objeto. Ao usar mais de um filtro, é necessário fazer a quebra dos filtros em um elemento `<And>`. O exemplo a seguir mostra como especificar objetos em um intervalo entre 500 e 64000 bytes.

```
<LifecycleConfiguration>
  <Rule>
    ...
    <And>
      <ObjectSizeGreater Than>500</ObjectSizeGreater Than>
      <ObjectSizeLess Than>64000</ObjectSizeLess Than>
    </And>
  </Rule>
</LifecycleConfiguration>
```

Inventário do Amazon S3

O Amazon S3 Inventory 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. Também é possível simplificar e acelerar os fluxos de trabalho de negócios e os trabalhos de big data usando o Amazon S3 Inventory, que fornece uma alternativa programada para a operação síncrona da API `List` do Amazon S3. O inventário do Amazon S3 não usa a API `List` para auditar seus objetos e não afeta a taxa de solicitação do bucket.

O Amazon S3 Inventory fornece arquivos de saída nos formatos CSV (valores separados por vírgulas), [ORC \(colunar de linhas otimizado do Apache\)](#) ou [Apache Parquet](#) que listam seus objetos e os metadados correspondentes, diária ou semanalmente, para um bucket do S3 ou prefixo compartilhado (ou seja, objetos que tenham nomes que começem com uma string comum). Caso isso ocorra semanalmente, um relatório será gerado todos os domingos (UTC) após o relatório inicial. Para obter mais informações sobre preços do Amazon S3 Inventory, consulte [Preços 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.

É possível consultar o Amazon S3 Inventory usando o SQL padrão, o [Amazon Athena](#), o Amazon Redshift Spectrum e outras ferramentas, como [Presto](#), [Apache Hive](#) e [Apache Spark](#). Você pode usar o Athena para executar consultas em seus arquivos de inventário. É possível usar as consultas do Amazon S3 Inventory em todas as regiões onde o Athena está disponível.

Buckets de origem e destino

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 do Amazon S3 Inventory 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 [Manifesto de inventário \(p. 795\)](#).
- Deve haver uma política de 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 do bucket de origem.
- Pode ser igual ao bucket de origem.
- Pode pertencer a uma Conta da AWS diferente da conta que é proprietária do bucket de origem.

Lista do Amazon S3 Inventory

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 compactado com Snappy. Os objetos são classificados em ordem crescente com base nos nomes das chaves.

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.
- Key name (Nome da chave): o 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.
- Version ID (ID da versão): o ID de versão do objeto. 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 [Usando o versionamento em buckets do S3 \(p. 678\)](#). (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.)
- Size (Tamanho): o tamanho do objeto em bytes.
- Last modified date (Data da última modificação) – a 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 ser um resumo MD5 dos dados do objeto. Tudo depende de como o objeto foi criado e de como está criptografado.
- Storage class (Classe de armazenamento) – a classe de armazenamento usada para armazenar o objeto. Para obter mais informações, consulte [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#).
- Sinalizador de multipart upload: definido como `True` se o objeto foi carregado como um multipart upload. Para obter mais informações, consulte [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).
- Delete marker (Marcador de exclusão): definido como `True` se o objeto for um marcador de exclusão. Para obter mais informações, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#). (Este campo é adicionado automaticamente ao seu relatório se você configurou o relatório para incluir todas as versões dos seus objetos).
- Replication status (Status da replicação): defina como `PENDING`, `COMPLETED`, `FAILED`, ou `REPLICA`. Para obter mais informações, consulte [Obtenção de informações sobre o status da replicação \(p. 873\)](#).
- 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 \(p. 358\)](#).
- Manutenção do bloqueio do objeto do S3 até a data: a data até a qual o objeto bloqueado não pode ser excluído. Para obter mais informações, consulte [Usar o bloqueio de objetos do S3 \(p. 722\)](#).
- Modo de bloqueio do objeto do S3: definido como `Governance` ou `Compliance` para objetos que estejam bloqueados. Para obter mais informações, consulte [Usar o bloqueio de objetos do S3 \(p. 722\)](#).
- S3 Object Lock Legal hold status (Status de retenção legal de bloqueio do objeto do S3): definido como `On` caso uma retenção legal tenha sido aplicada a um objeto. Caso contrário, ele será definido como `Off`. Para obter mais informações, consulte [Usar o bloqueio de objetos do S3 \(p. 722\)](#).
- S3 Intelligent-Tiering access tier (Nível de acesso do S3 Intelligent-Tiering): nível de acesso (frequente ou infrequente) do objeto se armazenado no S3 Intelligent-Tiering. Para obter mais informações, consulte [Classe de armazenamento para otimizar automaticamente dados com padrões de acesso alterados ou desconhecidos \(p. 732\)](#).
- S3 Bucket Key Status (Status da chave do bucket do S3): defina como `ENABLED` ou `DISABLED`. Indica se o objeto usa a chave do bucket do S3 para criptografia no lado do servidor. Para obter mais informações, consulte [Uso de chaves de bucket do Amazon S3 \(p. 369\)](#).
- Checksum Algorithm (Algoritmo de soma de verificação): indica o algoritmo usado para criar a soma de verificação do objeto.

Recomendamos que você crie uma política de ciclo de vida que exclua listas de inventário antigas. Para obter mais informações, consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

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 AWS SDKs. Para obter mais informações, consulte [Objeto HEAD](#) na Referência de APIs do Amazon Simple Storage Service.

Para obter mais informações sobre como trabalhar com o Amazon S3 Inventory, consulte os tópicos a seguir.

Tópicos

- [Configurar o inventário do Amazon S3 \(p. 789\)](#)
- [Configuração de notificações de eventos do Amazon S3 para conclusão de inventário \(p. 794\)](#)
- [Localização de lista de inventário \(p. 794\)](#)
- [Consulta do Amazon S3 Inventory com o Amazon Athena \(p. 797\)](#)
- [Converter strings de ID de versão vazias nos relatórios do Amazon S3 Inventory em cadeias de caracteres null \(p. 799\)](#)

Configurar o inventário do Amazon S3

O Amazon S3 Inventory fornece uma lista de arquivos simples dos seus objetos e metadados, que é uma alternativa agendada para a operação da API `List` síncrona do Amazon S3. O Amazon S3 Inventory fornece arquivos de saída no formato de valores separados por vírgulas (CSV), [formato colunar de linhas otimizado \(ORC\) do Apache](#) ou [Apache Parquet \(Parquet\)](#) que listam seus objetos e os metadados correspondentes, com frequência diária ou semanal, para um bucket do S3 ou para os objetos que compartilham um prefixo (objetos com nomes iniciados pela mesma string). Para obter mais informações, consulte [Inventário do Amazon S3 \(p. 786\)](#).

Esta seção descreve como configurar um inventário, incluindo detalhes sobre seus buckets de origem e destino do inventário.

Tópicos

- [Visão geral \(p. 789\)](#)
- [Criação de uma política de bucket de destino \(p. 790\)](#)
- [Conceder permissão para que o Amazon S3 use a AWS KMS key para criptografia \(p. 791\)](#)
- [Configuração de um inventário usando o console do S3 \(p. 792\)](#)

Visão geral

O Amazon S3 Inventory ajuda você a gerenciar seu armazenamento, criando listas dos objetos em um bucket do S3 em uma programação definida. 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 AWS Management Console, 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.

Para configurar o Amazon S3 inventory para um bucket do S3

1. Adicionar uma política do bucket para o bucket de destino.

É necessário criar uma política de 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 Amazon S3 Inventory e a análise do Simple Storage Service \(Amazon S3\) \(p. 526\)](#).

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.

É possível especificar que o arquivo da lista de inventário seja criptografado usando uma chave gerenciada do Amazon S3 (SSE-S3) ou uma chave gerenciada pelo cliente do AWS Key Management Service (AWS KMS). Para obter mais informações sobre SSE-S3 e SSE-KMS, consulte [Proteção de dados usando criptografia no lado do servidor \(p. 359\)](#). Se você pretende usar a criptografia de SSE-KMS, consulte a Etapa 3.

- Para obter informações sobre como usar o console para configurar uma lista de inventário, consulte [Configurar o inventário do Amazon S3 \(p. 789\)](#).
 - Para usar a API do Amazon S3 para configurar uma lista de inventários, use a API REST da [configuração de inventário de PUT Bucket](#) ou o equivalente da AWS CLI ou dos AWS SDKs.
3. Para criptografar o arquivo da lista de inventários com o SSE-KMS, conceda permissão para que o Amazon S3 use a AWS KMS key.

Você pode configurar a criptografia do arquivo da lista de inventários usando o AWS Management Console, a API REST, a AWS CLI ou os AWS SDKs. Independentemente do que você escolher, é necessário conceder permissão para que o Amazon S3 use a chave gerenciada pelo cliente para criptografar o arquivo de inventário. Conceda permissão ao Amazon S3 modificando a política de chave da chave gerenciada pelo cliente que você deseja usar para criptografar o arquivo de inventário. Para obter mais informações, consulte a próxima sessão, [Conceder permissão para que o Amazon S3 use a AWS KMS key para criptografia \(p. 791\)](#).

4. Para configurar o inventário, consulte [Configuração de um inventário usando o console do S3 \(p. 792\)](#).

Se você usar criptografia para operações entre contas da configuração de inventário do Amazon S3 no bucket de destino, deverá usar o ARN de chave do KMS totalmente qualificado. Para obter mais informações, consulte [Usar criptografia para operações entre contas \(p. 143\)](#) e [ServerSideEncryptionByDefault](#).

Criação de uma política de bucket de destino

Se você criar a configuração de inventário via console S3, o Amazon S3 criará automaticamente uma política de bucket no bucket de destino que concede ao Amazon S3 permissão para gravação. Se você criar a configuração de inventário via AWS CLI, SDKs ou a API REST, será necessário adicionar manualmente uma política de bucket no bucket de destino. Para obter mais informações, consulte [Conceder permissões para o Amazon S3 Inventory e a análise do Simple Storage Service \(Amazon S3\) \(p. 526\)](#). A política do bucket de destino do Amazon S3 Inventory permite que o Amazon S3 grave dados para os relatórios de inventário no bucket.

Se ocorrer um erro quando você tentar criar a política do bucket, você receberá instruções sobre como corrigi-lo. Por exemplo, se você escolher um bucket de destino em outra Conta da AWS e não tiver as permissões para ler e gravar na política do bucket, você verá uma mensagem de erro.

Nesse caso, o proprietário do bucket de destino deve adicionar a política do bucket exibida ao bucket de destino. Se a política não for adicionada ao bucket de destino, você não terá um relatório de inventário, pois o Amazon S3 não tem permissão para gravar no bucket de destino. Se o bucket de origem pertencer a uma conta diferente da conta do usuário atual, o ID de conta correto do bucket de origem deverá ser substituído na política.

Conceder permissão para que o Amazon S3 use a AWS KMS key para criptografia

Para conceder permissão para que o Amazon S3 criptografe usando uma chave do AWS Key Management Service (AWS KMS) gerenciada pelo cliente, é necessário usar uma política de chave. Para atualizar sua política de chaves para que você possa usar uma chave gerenciada pelo cliente, siga as etapas abaixo.

Para conceder permissões de criptografia usando a chave do KMS

1. Usando a conta da AWS que é proprietária da chave gerenciada pelo cliente, faça login no AWS Management Console.
2. Abra o console do AWS KMS em <https://console.aws.amazon.com/kms>.
3. Para alterar a Região da AWS, use o seletor de regiões no canto superior direito da página.
4. No painel de navegação esquerdo, escolha Customer managed keys (Chaves gerenciadas pelo cliente).
5. Em Customer managed keys (Chaves gerenciadas pelo cliente), escolha a chave gerenciada pelo cliente que você deseja usar para criptografar o arquivo de inventário.
6. Em Key policy (Política de chave), escolha Switch to policy view (Alternar para visualização de política).
7. Para atualizar a política de chave, escolha Edit (Editar).
8. Em Edit key policy (Editar política de chave), adicione a seguinte política de chave à existente.

```
{  
    "Sid": "Allow Amazon S3 use of the KMS key",  
    "Effect": "Allow",  
    "Principal": {  
        "Service": "s3.amazonaws.com"  
    },  
    "Action": [  
        "kms:GenerateDataKey"  
    ],  
    "Resource": "*",  
    "Condition":{  
        "StringEquals":{  
            "aws:SourceAccount":"source-account-id"  
        },  
        "ArnLike":{  
            "aws:SourceARN": "arn:aws:s3:::source-bucket-name"  
        }  
    }  
}
```

9. Selecione Save changes.

Para obter mais informações sobre como criar chaves gerenciadas pelo cliente e usar políticas de chaves, consulte os links a seguir no Guia do desenvolvedor do AWS Key Management Service:

- [Conceitos básicos](#)
- [Usar políticas de chaves no AWS KMS](#)

Configuração de um inventário usando o console do S3

Use estas instruções para configurar o inventário usando o console do S3.

Note

O primeiro relatório pode demorar até 48 horas para ser entregue.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket para o qual deseja configurar o Amazon S3 Inventory.
3. Escolha Management.
4. Em Inventory configurations (Configurações de inventário), escolha Create inventory configuration (Criar configuração de inventário).
5. Em Inventory configuration name (Nome da configuração do inventário), insira um nome.
6. Defina o Inventory scope (Escopo do inventário):
 - Insira um prefixo opcional.
 - Escolha versões de objeto: Current versions only (Somente versões atuais) ou Include all versions (Incluir todas as versões).
7. Em Report details (Detalhes do relatório), escolha o local da Conta da AWS em que deseja salvar os relatórios: This account (Esta conta) ou A different account (Uma conta diferente).
8. Em Destination (Destino), selecione o bucket de destino no qual deseja salvar os relatórios.

O bucket de destino deve estar na mesma Região da AWS que o bucket para o qual você está configurando o inventário. O bucket de destino pode estar em uma Conta da AWS diferente. No campo de bucket Destination (Destino), você verá Destination bucket permission (Permissão do bucket de destino) que é adicionada à política de bucket de destino para permitir que o Amazon S3 coloque dados nesse bucket. Para obter mais informações, consulte [Criação de uma política de bucket de destino \(p. 790\)](#).

9. Em Frequency (Frequência), escolha a frequência com que o relatório será gerado: Daily (Diário) ou Weekly (Semanal).
10. Escolha o Output format (Formato de saída) para o relatório:
 - CSV
 - Apache ORC
 - Apache Parquet
11. Em Status, escolha Enable (Ativar) ou Disable (Desabilitar).
12. Para usar a criptografia do lado do servidor, em Server-side encryption (Criptografia do lado do servidor), siga estas etapas:
 - a. Escolha Habilitar.
 - b. Em Encryption key type (Tipo de chave de criptografia), escolha Amazon S3 key (SSE-S3) [Chave do Amazon S3 (SSE-S3)] ou AWS Key Management Service key (SSE-KMS) [Chave do AWS Key Management Service (SSE-KMS)].

A criptografia do lado do servidor do Amazon S3 usa o Advanced Encryption Standard de 256 bits (AES-256). Para obter mais informações, consulte [Proteção de dados usando criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 \(SSE-S3\) \(p. 378\)](#). Para obter mais informações sobre SSE-KMS, consulte [Proteção de dados usando criptografia do servidor com AWS Key Management Service \(SSE-KMS\) \(p. 360\)](#).

- c. Para usar uma AWS KMS key, escolha uma das seguintes opções:
 - Choose from your AWS KMS keys (Escolher de suas chaves do KMS), e escolha sua chave do KMS.

- Enter AWS KMS key ARN (Inserir ARN da chave do KMS) e insira o seu ARN da chave do AWS KMS.

Note

Para criptografar o arquivo da lista de inventários com SSE-KMS, você deve conceder uma permissão ao Amazon S3 para usar a AWS KMS key. Portanto, você só pode usar uma [chave gerenciada pelo cliente](#) e não a [chave gerenciada pela AWS \(aws/s3\)](#). Para obter instruções, consulte [Conceder permissão para que o Amazon S3 use a AWS KMS key para criptografia \(p. 791\)](#).

13. Em Additional fields (Campos adicionais), selecione um ou mais dos seguintes campos para adicionar ao relatório de inventário:

- Size (Tamanho): o tamanho do objeto em bytes.
- Last modified date (Data da última modificação) – a data de criação do objeto ou data da última modificação, a mais atual entre as duas.
- Storage class (Classe de armazenamento) – a classe de armazenamento usada para armazenar o objeto.
- ETag: a tag de entidade é um hash do objeto. A ETag reflete as alterações somente nos conteúdos de um objeto e 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.
- Multipart upload – especifica se o objeto foi carregado como um multipart upload. Para obter mais informações, consulte [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).
- Replication status (Status da replicação) – o status de replicação do objeto. Para obter mais informações, consulte [Uso do console do S3 \(p. 823\)](#).
- Encryption status (Status da criptografia) – a criptografia do lado do servidor usada para criptografar o objeto. Para obter mais informações, consulte [Proteção de dados usando criptografia no lado do servidor \(p. 359\)](#).
- S3 Object lock configurations (Configurações do bloqueio de objeto do S3): o status do bloqueio de objetos, incluindo as seguintes configurações:
 - Retention mode (Modo de retenção) – O nível de proteção aplicado ao objeto, Governance (Governança) ou Compliance (Conformidade).
 - Retain until date (Reter até uma data específica) – a data até a qual o objeto bloqueado não pode ser excluído.
 - Legal hold status (Status da retenção legal) – o status de retenção legal do objeto bloqueado.

Para obter mais informações sobre bloqueio de objetos do S3, consulte [Como o bloqueio de objetos do S3 funciona \(p. 723\)](#).

- (Intelligent-Tiering access tier (Nível de acesso do Intelligent-Tiering): indica o nível de acesso (frequente ou infrequente) do objeto se armazenado no Intelligent-Tiering. Para obter mais informações, consulte [Classe de armazenamento para otimizar automaticamente dados com padrões de acesso alterados ou desconhecidos \(p. 732\)](#).
- S3 Bucket Key Status (Status da chave do bucket do S3): indica se uma chave no nível do bucket gerada pelo AWS KMS se aplica ao objeto. Para obter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3 \(p. 369\)](#).
- Checksum Algorithm (Algoritmo de soma de verificação): indica o algoritmo usado para criar a soma de verificação do objeto.

Para obter mais informações sobre o conteúdo de um relatório de inventário, consulte [Lista do Amazon S3 Inventory \(p. 787\)](#).

14. Escolha Create (Criar).

Configuração de notificações de eventos do Amazon S3 para conclusão de inventário

É possível configurar uma notificação de evento do Amazon S3 para receber um 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), em uma fila do Amazon Simple Queue Service (Amazon SQS) ou em uma função do AWS Lambda. Para obter mais informações, consulte [Notificações de eventos do Amazon S3 \(p. 1083\)](#).

A configuração de notificação a seguir define que todos os arquivos `manifest.checksum` recém-adicionados ao bucket de destino são processados pela AWS Lambda do `cloud-function-list-write`.

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>1</Id>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>prefix</Name>
          <Value>destination-prefix/source-bucket</Value>
        </FilterRule>
        <FilterRule>
          <Name>suffix</Name>
          <Value>checksum</Value>
        </FilterRule>
      </S3Key>
    </Filter>
    <Cloudcode>arn:aws:lambda:us-west-2:222233334444:cloud-function-list-write</Cloudcode>
    <Event>s3:ObjectCreated:*</Event>
  </QueueConfiguration>
</NotificationConfiguration>
```

Para obter mais informações, consulte [Uso do AWS Lambda com o Amazon S3](#) no Guia do desenvolvedor do AWS Lambda.

Localização de lista de inventário

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. O *config-ID* vem da configuração do relatório de inventário e é o nome do relatório definido na configuração.
- *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.
- *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/data/example-file-name.csv.gz
...
destination-prefix/source-bucket/config-ID/data/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*.

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. Esses arquivos podem se sobrepor. Em buckets habilitados para versionamento, o Amazon S3 cria versões dos arquivos de manifesto.

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*.

Example Manifesto de inventário em um arquivo *manifest.json*

Os exemplos a seguir mostram um manifesto de inventário em um arquivo *manifest.json* para inventários formatados CSV, ORC e Parquet.

CSV

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, IntelligentTieringAccessTier, BucketKeyStatus,  
ChecksumAlgorithm",  
    "files": [  
        {  
            "key": "Inventory/example-source-bucket/2016-11-06T21-32Z/  
files/939c6d46-85a9-4ba8-87bd-9db705a579ce.csv.gz",  
            "size": 2147483647,  
            "MD5checksum": "f11166069f1990abeb9c97ace9cdfabc"  
        }  
    ]  
}
```

ORC

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:  
    "files": [  
        {  
            "key": "inventory/example-source-bucket/data/  
d794c570-95bb-4271-9128-26023c8b4900.orc",  
            "size": 56291,  
            "MD5checksum": "5925f4e78e1695c2d020b9f6eexample"  
        }  
    ]  
}
```

Parquet

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); optional int64
object_lock_retain_until_date (TIMESTAMP_MILLIS); optional binary object_lock_mode
(UTF8); optional binary object_lock_legal_hold_status (UTF8); optional binary
intelligent_tiering_access_tier (UTF8); optional binary bucket_key_status (UTF8);
optional binary checksum_algorithm (UTF8); }",
    "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 Athena e o Amazon Redshift Spectrum. Ele também funciona com aplicativos compatíveis com o Hive, como [Presto](#), [Apache Hive](#), [Apache Spark](#) e muitos outros.

Important

O `symlink.txt` arquivo manifesto compatível com Apache Hive atualmente não funciona com o AWS Glue.

A leitura do `symlink.txt` com [Apache Hive](#) e [Apache Spark](#) não é compatível com arquivos de inventário nos formatos ORC e Parquet.

Consulta do Amazon S3 Inventory com o Amazon Athena

É possível consultar o Amazon S3 Inventory usando o SQL padrão ao usar o Amazon Athena em todas as regiões onde o Athena esteja disponível. Para verificar a disponibilidade da Região da AWS, consulte a [Tabela de Região da AWS](#).

O Athena pode consultar arquivos do Amazon S3 Inventory 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 uma performance de consulta mais rápida 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 o Amazon S3 Inventory estão disponíveis em todas as Regiões da AWS.

Como começar a usar o Athena para consultar o Amazon S3 Inventory

1. Crie uma tabela do Athena. Para obter informações sobre como criar uma tabela, 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, você deve usar o nome e a localização do seu bucket no caminho de destino do inventário. Substitua o nome do bucket a seguir e o local do inventário conforme apropriado para sua configuração: `s3://destination-prefix/DOC-EXAMPLE-BUCKET/config-ID/hive/`. Você também deve substituir a data inicial em `projection.dt.range` pelo primeiro dia que tiver dados.

```
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 bigint,  
    e_tag string,  
    storage_class string,  
    is_multipart_uploaded boolean,  
    replication_status string,  
    encryption_status string,  
    object_lock_retain_until_date bigint,  
    object_lock_mode string,  
    object_lock_legal_hold_status string,  
    intelligent_tiering_access_tier string,  
    bucket_key_status string,  
    checksum_algorithm string  
) 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/'  
TBLPROPERTIES (  
    "projection.enabled" = "true",  
    "projection.dt.type" = "date",  
    "projection.dt.format" = "yyyy-MM-dd-HH-mm",  
    "projection.dt.range" = "2022-01-01-00-00,NOW",  
    "projection.dt.interval" = "1",  
    "projection.dt.interval.unit" = "DAYS"  
)
```

Ao usar o 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'
```

Ao usar o Athena para consultar um relatório de inventário no formato CSV, use o modelo a seguir.

```
CREATE EXTERNAL TABLE your_table_name(  
    bucket string,  
    key string,  
    version_id string,  
    is_latest boolean,  
    is_delete_marker boolean,  
    size string,  
    last_modified_date string,  
    e_tag string,  
    storage_class string,  
    is_multipart_uploaded boolean,  
    replication_status string,  
    encryption_status string,  
    object_lock_retain_until_date string,  
    object_lock_mode string,  
    object_lock_legal_hold_status string,  
    intelligent_tiering_access_tier string,  
    bucket_key_status string,  
    checksum_algorithm string
```

```
) PARTITIONED BY (
    dt string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
    STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
    OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat'
    LOCATION 's3://destination-prefix/source-bucket/config-ID/hive/'
TBLPROPERTIES (
    "projection.enabled" = "true",
    "projection.dt.type" = "date",
    "projection.dt.format" = "yyyy-MM-dd-HH-mm",
    "projection.dt.range" = "2022-01-01-00-00,NOW",
    "projection.dt.interval" = "1",
    "projection.dt.interval.unit" = "DAYS"
);
```

- Depois de executar essa etapa, você pode executar consultas ad hoc no inventário, como mostrado no exemplo a seguir.

```
# Get list of latest inventory report dates available
SELECT DISTINCT dt FROM your_table_name ORDER BY 1 DESC limit 10;

# Get encryption status for a provided report date.
SELECT encryption_status, count(*) FROM your_table_name WHERE dt = 'YYYY-MM-DD-HH-MM'
GROUP BY encryption_status;

# Get encryption status for report dates in the provided range.
SELECT dt, encryption_status, count(*) FROM your_table_name
WHERE dt > 'YYYY-MM-DD-HH-MM' AND dt < 'YYYY-MM-DD-HH-MM' GROUP BY dt,
encryption_status;
```

Para obter mais informações sobre como usar o Athena, consulte o [Guia do usuário do Amazon Athena](#).

Veja a seguir as operações REST usadas para o Amazon S3 Inventory.

- [Inventário do bucket DELETE](#)
- [Inventário do bucket GET](#)
- [Inventário do bucket de lista](#)
- [Inventário do PUT Bucket](#)

Converter strings de ID de versão vazias nos relatórios do Amazon S3 Inventory em cadeias de caracteres null

Note

O procedimento a seguir se aplica somente aos relatórios do Amazon S3 Inventory que incluem todas as versões e somente se os relatórios “todas as versões” forem usados como manifestos para o S3 Batch Operations em buckets que têm o versionamento do S3 habilitado. Não é necessário converter strings para relatórios do S3 Inventory que especificam somente a versão atual.

Você pode usar relatórios do S3 Inventory como manifestos para o S3 Batch Operations. Porém, quando o versionamento S3 está habilitado em um bucket, os relatórios do S3 Inventory que incluem todas as versões marcam qualquer objeto com versionamento nulo com strings vazias no campo de ID da versão.

Quando um relatório de inventário inclui todas as IDs de versão do objeto, o Batch Operations reconhece strings `null` como IDs de versão, mas não strings vazias.

Quando um trabalho do S3 Batch Operations usa um relatório “all versions” (todas as versões) do S3 Inventory como manifesto, ele falha em todas as tarefas em objetos que têm uma string vazia no campo de ID da versão. Para converter cadeias de caracteres vazias no campo ID da versão do relatório do S3 Inventory para strings `null` do Batch Operations, use o procedimento a seguir.

Atualize um relatório do Amazon S3 Inventory para uso com o Batch Operations

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Navegue até o relatório do S3 Inventory. O relatório de inventário está localizado no bucket de destino especificado durante a configuração do relatório de inventário. Para obter mais informações sobre a localização de relatórios de inventário, consulte [Localização de lista de inventário \(p. 794\)](#).
 - a. Escolha o bucket de destino.
 - b. Escolha a pasta . A pasta tem o nome do bucket de origem original.
 - c. Escolha a pasta com o nome da configuração do inventário.
 - d. Marque a caixa de seleção ao lado da pasta chamada `hive`. Na parte superior da página, escolha `Copy S3 URI` (Copiar URI do S3) para copiar a URI do S3 para a pasta.
3. Abra o console do Amazon Athena em <https://console.aws.amazon.com/athena/>.
4. No editor de consultas, escolha `Settings` (Configurações) e escolha `Manage` (Gerenciar). Na página `Manage settings` (Gerenciar configurações), em `Location of query result` (Local do resultado da consulta), escolha um bucket do S3 para armazenar os resultados da consulta.
5. No editor de consultas, crie uma tabela do Athena para manter os dados no relatório de inventário usando o comando a seguir. Substitua `table_name` pelo nome de sua preferência, e na cláusula `LOCATION`, insira a URI do S3 que você copiou anteriormente. Em seguida, escolha `Run` (Executar) para executar a consulta.

```
CREATE EXTERNAL TABLE table_name(bucket string, key string,  
version_id string) PARTITIONED BY (dt string)ROW FORMAT SERDE  
'org.apache.hadoop.hive.serde2.OpenCSVSerde' STORED AS INPUTFORMAT  
'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat' OUTPUTFORMAT  
'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat' LOCATION 'Copied S3 URI';
```

6. Para limpar o editor de consultas, escolha `Clear` (Limpar). Então carregue o relatório de inventário na tabela usando o comando a seguir. Substitua `table_name` pelo que você escolheu na etapa anterior. Em seguida, escolha `Run` (Executar) para executar a consulta.

```
MSCK REPAIR TABLE table_name;
```

7. Para limpar o editor de consultas, escolha `Clear` (Limpar). Execute a seguinte consulta `SELECT` para recuperar todas as entradas no relatório de inventário original e substituir quaisquer IDs de versão vazias por strings `null`. Substitua `table_name` pelo que você escolheu anteriormente e substitua `YYYY-MM-DD-HH-MM` na cláusula `WHERE` pela data do relatório de inventário em que você deseja que essa ferramenta seja executada. Em seguida, escolha `Run` (Executar) para executar a consulta.

```
SELECT bucket as Bucket, key as Key, CASE WHEN version_id = '' THEN 'null' ELSE  
version_id END as VersionId FROM table_name WHERE dt = 'YYYY-MM-DD-HH-MM';
```

8. Retorne ao console do Amazon S3 (<https://console.aws.amazon.com/s3/>) e navegue até o bucket do S3 que você escolheu para `Location of query result` (Local do resultado da consulta) anteriormente. Dentro dele, deve haver uma série de pastas terminando com a data.

Por exemplo, você verá algo parecido com s3://**DOC-EXAMPLE-BUCKET**/query-result-location/Unsaved/2021/10/07/. Haverá arquivos .csv contendo os resultados da consulta SELECT que você executou.

Escolha o arquivo CSV com a data de modificação mais recente. Baixe esse arquivo na máquina local para a próxima etapa.

9. O arquivo CSV gerado contém uma linha de cabeçalho. Para usar esse arquivo CSV como entrada para um trabalho do S3 Batch Operations, é necessário remover a linha de cabeçalho, pois o Batch Operations não é compatível com linhas de cabeçalho em manifestos CSV.

Para remover a linha do cabeçalho, execute um dos comandos a seguir no arquivo. Substitua **file.csv** pelo nome de seu arquivo CSV.

Para máquinas macOS e Linux, execute o comando tail em uma janela Terminal.

```
tail -n +2 file.csv > tmp.csv && mv tmp.csv file.csv
```

Para máquinas Windows, execute o script a seguir em uma janela do Windows PowerShell. Substitua **File-location** pelo caminho do arquivo e **file.csv** pelo nome do arquivo.

```
$ins = New-Object System.IO.StreamReader File-location\file.csv
$out = New-Object System.IO.StreamWriter File-location\temp.csv
try {
    $skip = 0
    while ( !$ins.EndOfStream ) {
        $line = $ins.ReadLine();
        if ( $skip -ne 0 ) {
            $out.WriteLine($line);
        } else {
            $skip = 1
        }
    }
} finally {
    $out.Close();
    $ins.Close();
}
Move-Item File-location\temp.csv File-location\file.csv -Force
```

10. Depois de remover a linha de cabeçalho do arquivo CSV, você estará pronto para usá-lo como manifesto em um trabalho do S3 Batch Operations. Carregue o arquivo CSV para um bucket do S3 ou local de sua preferência e crie um trabalho do Batch Operations usando o arquivo CSV como manifesto.

Para obter mais informações sobre como criar um trabalho do Batch Operations, consulte [Criar um trabalho de operações em lote do S3 \(p. 946\)](#).

Replicação de objetos

O Replication permite a cópia automática e assíncrona de objetos nos buckets do Amazon S3. Os buckets configurados para replicação de objetos podem pertencer à mesma conta da Conta da AWS ou a contas diferentes. Os objetos podem ser replicados em um ou em vários buckets de destino. Os buckets de destino podem estar em Regiões da AWS diferentes ou na mesma região que o bucket de origem.

Para replicar automaticamente novos objetos à medida que forem gravados no bucket, use replicação em tempo real, como a Same-Region Replication (SRR – Replicação na Mesma Região) e a Cross-Region Replication (CRR – Replicação entre Regiões). Para replicar os objetos existentes em um bucket

diferente sob demanda, use S3 Batch Replication. Para obter mais informações sobre replicação de objetos existentes, consulte [Quando usar S3 Batch Replication \(p. 803\)](#).

Para habilitar a SRR ou a CRR, adicione uma configuração de replicação ao bucket de origem. A configuração mínima deve fornecer o seguinte:

- O bucket de destino ou buckets nos quais você quer que o Amazon S3 replique os objetos
- Uma função AWS Identity and Access Management do (IAM) que o Amazon S3 pode assumir para replicar objetos em seu nome

Estão disponíveis opções de configuração adicionais. Para obter mais informações, consulte [Configurações de replicação adicionais \(p. 857\)](#).

Tópicos

- [Por que usar a replicação \(p. 802\)](#)
- [Quando usar a replicação entre regiões \(p. 803\)](#)
- [Quando usar Replicação na mesma região \(p. 803\)](#)
- [Quando usar S3 Batch Replication \(p. 803\)](#)
- [Requisitos para replicação \(p. 804\)](#)
- [O que o Amazon S3 replica? \(p. 804\)](#)
- [Configuração da replicação \(p. 807\)](#)
- [Replicar objetos existentes com o S3 Batch Replication \(p. 849\)](#)
- [Configurações de replicação adicionais \(p. 857\)](#)
- [Obtenção de informações sobre o status da replicação \(p. 873\)](#)
- [Solução de problemas de replicação \(p. 875\)](#)
- [Considerações adicionais \(p. 877\)](#)

Por que usar a replicação

A replicação pode ajudar você a fazer o seguinte:

- Replicar objetos retendo os metadados: você pode usar a replicação para fazer cópias dos objetos retendo todos os metadados, como as horas de criação e os IDs de versão dos objetos originais. Esse recurso é importante se você precisar garantir que a réplica seja idêntica ao objeto original.
- Replicar objetos em diferentes classes de armazenamento: você pode usar a replicação para colocar objetos diretamente nas classes de armazenamento S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive ou em outra classe de armazenamento no bucket de destino. Você também pode replicar seus dados para a mesma classe de armazenamento e usar políticas de ciclo de vida no bucket de destino para mover os objetos para uma classe de armazenamento "mais fria" à medida que eles forem ficando velhos.
- Manter cópias de objetos com proprietários diferentes: independentemente de quem seja o proprietário do objeto original, você pode instruir o Simple Storage Service (Amazon S3) a alterar o proprietário da réplica para a Conta da AWS que é proprietária do bucket de destino. Isso se chama opção de substituição do proprietário. Você pode usar esta opção para restringir acesso às réplicas do objeto.
- Manter objetos armazenados em várias Regiões da AWS: para garantir diferenças geográficas em onde seus dados são mantidos, você pode definir vários buckets de destino em diferentes Regiões da AWS. Esse recurso pode ajudar a atender a determinados requisitos de conformidade.
- Replicar objetos dentro de 15 minutos: para replicar dados na mesma Região da AWS ou entre diferentes regiões dentro de um período previsível, você pode usar S3 Replication Time Control (S3 RTC - Controle de Horário de Replicação do S3). O S3 RTC replica 99,99 por cento dos novos objetos

armazenados no Simple Storage Service (Amazon S3) em 15 minutos (baseado em um acordo de nível de serviço). Para obter mais informações, consulte [the section called “Uso do Controle do tempo de replicação do S3” \(p. 860\)](#).

- Sincronizar buckets, replicar objetos existentes e replicar objetos cuja replicação falhou anteriormente: para sincronizar buckets e replicar objetos existentes, use a replicação em lote como uma ação de replicação sob demanda. Para obter mais informações sobre quando usar a replicação em lote, consulte [Quando usar S3 Batch Replication \(p. 803\)](#).

Note

O S3 RTC não se aplica à replicação em lote. A replicação em lote é um trabalho de replicação sob demanda e pode ser rastreada com o recurso S3 Batch Operations. Para obter mais informações, consulte [Rastreamento de relatórios de status e conclusão \(p. 982\)](#).

Quando usar a replicação entre regiões

A replicação entre regiões (CRR) do S3 é usada para copiar objetos entre buckets do Amazon S3 em Regiões da AWS diferentes. As tags podem ajudar a fazer o seguinte:

- Atender aos requisitos de conformidade: embora o Simple Storage Service (Amazon S3), por padrão, armazene seus dados em diversas zonas de disponibilidade geograficamente distantes, requisitos de conformidade podem exigir que você armazene os dados a distâncias ainda maiores. Para atender a esses requisitos, use a Cross-Region Replication para replicar dados entre Regiões da AWS distantes.
- Minimizar a latência: se seus clientes estiverem em duas localizações geográficas diferentes, é possível minimizar a latência no acesso a objetos mantendo cópias dos objetos nas Regiões da AWS geograficamente mais próximas dos usuários.
- Aumentar a eficiência operacional: se tiver clusters de computação em duas Regiões da AWS diferentes que analisam o mesmo conjunto de objetos, talvez você prefira manter cópias dos objetos nessas regiões.

Quando usar Replicação na mesma região

A replicação para a mesma região (SRR) é usada para copiar objetos entre buckets do Amazon S3 na mesma Região da AWS. A SRR pode ajudar a fazer o seguinte:

- Agregar logs em um único bucket: se você armazenar os logs em vários buckets ou em várias contas, será possível replicar facilmente os logs em um único bucket na região. Isso permite processar os logs em um único local, o que é mais simples.
- Configurar replicação em tempo real entre contas de produção e teste: se você ou seus clientes tiverem contas de produção e de teste que usam os mesmos dados, será possível replicar objetos entre essas várias contas, mantendo os metadados dos objetos.
- Cumprir as leis de soberania de dados: pode ser necessário armazenar várias cópias dos dados em Contas da AWS separadas em uma determinada região. A replicação na mesma região pode ajudar a replicar automaticamente dados vitais quando os regulamentos de conformidade não permitirem que os dados saiam do país.

Quando usar S3 Batch Replication

A replicação em lote replica objetos existentes em diferentes buckets como uma opção sob demanda. Ao contrário da replicação em tempo real, esses trabalhos podem ser executados conforme a necessidade. A replicação em lote pode ajudar você a fazer o seguinte:

- Replicar objetos existentes: você pode usar a replicação em lote para replicar objetos que foram adicionados ao bucket antes que a Same-Region Replication ou a Cross-Region Replication fossem configuradas.
- Replicar objetos cuja replicação falhou anteriormente: você pode filtrar um trabalho de replicação em lote para tentar replicar objetos com um status de replicação de FAILED (FALHOU).
- Replicar objetos que já foram replicados: pode ser necessário armazenar várias cópias dos dados em Contas da AWS ou Regiões da AWS separadas. A replicação em lote pode replicar objetos existentes em destinos recém-adicionados.
- Replicar réplicas de objetos que foram criadas segundo uma regra de replicação: as configurações de replicação criam réplicas de objetos em buckets de destino. As réplicas de objetos só podem ser replicadas com a replicação em lote.

Requisitos para replicação

A replicação exige o seguinte:

- O proprietário do bucket de origem deve ter as Regiões da AWS de origem e de destino habilitadas para a conta. O proprietário do bucket de destino deve ter a região de destino habilitada para a conta.
Para obter mais informações sobre como ativar e desativar uma Região da AWS, consulte [Managing Regiões da AWS](#) (Gerenciar Regiões da AWS) na Referência geral da AWS.
- Tanto o bucket de origem quanto o de destino devem ter o versionamento habilitado. Para obter mais informações sobre versionamento, consulte [Usando o versionamento em buckets do S3](#) (p. 678).
- O Amazon S3 deve ter permissões para replicar objetos do bucket de origem para o bucket de destino ou buckets em seu nome. Para obter mais informações sobre essas permissões, consulte [Configuração de permissões](#) (p. 818).
- 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 lista de controle de acesso (ACL) do objeto. Para obter mais informações, consulte [Visão geral da lista de controle de acesso \(ACL\)](#) (p. 587).
- Se o bucket de origem tiver o Bloqueio de objetos do S3 habilitado, o bucket de destino também deverá ter o Bloqueio de objetos do S3 habilitado.

Para obter mais informações, consulte [Usar o bloqueio de objetos do S3](#) (p. 722). Para habilitar a replicação em um bucket que tenha o bloqueio de objetos habilitado, entre em contato com o [AWS Support](#).

Para obter mais informações, consulte [Configuração da replicação](#) (p. 807).

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), o seguinte requisito adicional se aplicará:

- O proprietário dos buckets de destino precisa conceder ao proprietário do bucket de origem permissões para replicar objetos com uma política de bucket. Para obter mais informações, consulte [Conceder permissões quando os buckets de origem e de destino pertencerem a Contas da AWS diferentes](#) (p. 820).
- O bucket de destino não pode ser configurado como um bucket de Pagamento pelo solicitante. Para obter mais informações, consulte [Configuração de buckets de Pagamento pelo solicitante para transferências de armazenamento e uso](#) (p. 155).

O que o Amazon S3 replica?

O Amazon S3 replica somente itens específicos nos buckets que estão configurados para replicação.

Tópicos

- [O que é replicado com as configurações de replicação? \(p. 805\)](#)
- [O que não é replicado com as configurações de replicação? \(p. 806\)](#)

O que é replicado com as configurações de replicação?

Por padrão, o Simple Storage Service (Amazon S3) replica o seguinte:

- Objetos criados depois de adicionar uma configuração de replicação.
- Objetos não criptografados.
- Objetos criptografados que estão em repouso e têm uma chave gerenciada pelo Simple Storage Service (Amazon S3) (SSE-S3) ou uma chave do KMS armazenada no AWS Key Management Service (SSE-KMS).

Para replicar objetos criptografados com uma chave do KMS armazenada no AWS KMS, é necessário habilitar explicitamente a opção. 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 \(p. 359\)](#).

- Metadados de objeto dos objetos de origem para as réplicas. Para obter informações sobre como replicar metadados das réplicas para os objetos de origem, consulte [Replicação de alterações de metadados com sincronização de modificação de réplica do Amazon S3 \(p. 863\)](#).
- 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 [Propriedade de bucket e objeto do Amazon S3 \(p. 420\)](#).

- 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.

Para obter mais informações, consulte [Alterar o proprietário da réplica \(p. 864\)](#).

Pode levar um tempo até o Amazon S3 sincronizar as duas ACLs. Essa alteração de proprietário aplica-se apenas aos objetos criados depois que você adiciona uma configuração de replicação ao bucket.

- Tags de objeto, se houver.
- Informações de retenção de bloqueio de objetos do S3, se houver.

Quando o Amazon S3 replica objetos que tenham retenção aplicadas, ele aplica esse mesmo controle de retenção às suas réplicas, substituindo o período de retenção padrão configurado nos buckets de destino. Se você não tiver controles de retenção aplicados aos objetos em seu bucket de origem e replicá-los em buckets de destino que tenham um período de retenção definido, o período de retenção padrão do bucket de destino será aplicado às réplicas de objeto. Para obter mais informações, consulte [Usar o bloqueio de objetos do S3 \(p. 722\)](#).

Como a exclusão de operações afeta a replicação

Se você excluir um objeto do bucket de origem, as seguintes ações ocorrerão por padrão:

- Se você fizer uma solicitação DELETE sem especificar um ID de versão de objeto, o Amazon S3 adicionará um marcador de exclusão. O Amazon S3 lida com o marcador de exclusão da seguinte forma:
 - 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 replicará o

marcador de exclusão, por padrão. Porém, você pode adicionar replicação de marcador de exclusão a regras não baseadas em etiquetas. Para obter mais informações, consulte [Replicação de marcadores de exclusão entre intervalos \(p. 862\)](#).

- Se você não especificar o elemento `Filter`, o Amazon S3 assumirá que a configuração de replicação é a versão V1 e replicará marcadores de exclusão resultantes de ações do usuário. No entanto, se o Amazon S3 excluir um objeto devido a uma ação de ciclo de vida, o marcador de exclusão não será replicado para os buckets de destino.
- 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. Porém, ele não replica a exclusão no bucket de destino. Em outras palavras: ele não exclui a mesma versão do objeto dos buckets de destino. Isso protege os dados contra exclusões mal-intencionadas.

O que não é replicado com as configurações de replicação?

Por padrão, o Simple Storage Service (Amazon S3) não replica o seguinte:

- Objetos no bucket de origem que são réplicas criadas por outra regra de replicação. Por exemplo, se você configurar uma replicação em que o bucket A é a origem e o bucket B é o destino. Agora, suponha que você adicione outra configuração da replicação 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.

Para replicar objetos que são réplicas, use a replicação em lote. Saiba mais sobre como configurar a replicação em lote em [Replicar objetos existentes \(p. 849\)](#).

- Objetos no bucket de origem que já foram replicados para um destino diferente. Por exemplo, se você alterar o bucket de destino em uma configuração de replicação existente, o Simple Storage Service (Amazon S3) não replicará os objetos novamente.

Para replicar objetos que já foram replicados anteriormente, use a replicação em lote. Saiba mais sobre como configurar a replicação em lote em [Replicar objetos existentes \(p. 849\)](#).

- A replicação em lote não é compatível com a nova replicação de objetos que foram excluídos com o ID de versão do objeto do bucket de destino. Para replicar esses objetos novamente, você pode copiar os objetos de origem no local com um trabalho de cópia em lote. Copiar esses objetos no local criará versões do objeto no bucket de origem e iniciará a replicação automaticamente para o destino. Para obter mais informações sobre como usar a cópia em lote, consulte [Exemplos que usam o Batch Operations para copiar objetos \(p. 954\)](#).
- Objetos criados com criptografia do lado do servidor usando as chaves de criptografia fornecidas pelo cliente (SSE-C).
- Por padrão, ao replicar de uma Conta da AWS diferente, os marcadores de exclusão adicionados ao bucket de origem não são replicados.

Para obter mais informações sobre como os marcadores de exclusão funcionam, consulte [Replicação de marcadores de exclusão entre intervalos \(p. 862\)](#).

- Objetos armazenados na classe de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive.

Para saber mais sobre o serviço Amazon S3 Glacier, consulte o [Guia do desenvolvedor do Amazon S3 Glacier](#).

- Objetos no bucket de origem que o proprietário do bucket não tem permissões suficientes para replicar.

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 \(p. 526\)](#).

- 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. Esse recurso torna possível ter configurações diferentes no 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. Se você quiser que a mesma configuração de ciclo de vida seja aplicada aos buckets 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 [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

Configuração da replicação

Note

Objetos que já existiam antes de você configurar a replicação não são replicados automaticamente. Em outras palavras: o Amazon S3 não replica os objetos retroativamente. Para replicar objetos que foram criados antes da configuração de replicação, use S3 Batch Replication. Saiba mais sobre como configurar a replicação em lote em [Replicar objetos existentes \(p. 849\)](#).

Para habilitar a Same-Region Replication (SRR) ou a Cross-Region Replication (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:

- Os buckets de destino: o bucket ou buckets onde você quer que o Simple Storage Service (Amazon S3) replique os objetos.
- Os objetos que você deseja replicar: você pode replicar todos os objetos do bucket de origem ou um subgrupo deles. 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 uma regra de replicação para replicar somente objetos com o prefixo de nome da chave `Tax/`, o Amazon S3 replicará objetos com chaves como `Tax/doc1` ou `Tax/doc2`. Porém, ele não replica objetos com a chave `Legal1/doc3`. Se você especificar um prefixo e uma ou mais etiquetas, o Simple Storage Service (Amazon S3) replicará somente os objetos com o prefixo das chaves e as etiquetas específicas.

Além desses requisitos mínimos, você pode escolher as seguintes opções:

- Classe de armazenamento de réplicas: por padrão, o Simple Storage Service (Amazon S3) armazena réplicas de objetos usando a mesma classe de armazenamento do objeto de origem. Você pode especificar uma classe de armazenamento diferente para as réplicas.
- Propriedade da réplica: o Simple Storage Service (Amazon S3) pressupõe que a réplica de um objeto continue pertencendo ao proprietário do objeto de origem. Portanto, quando ele replica objetos, também replica a lista de controle de acesso (ACL) a objetos correspondente ou a configuração de propriedade de objetos do S3. Se os buckets de origem e destino pertencerem a Contas da AWS diferentes, você poderá configurar a replicação para alterar o proprietário de uma réplica para a Conta da AWS que é proprietária do bucket de destino.

Você pode configurar a replicação usando a API REST, os AWS SDKs, a AWS Command Line Interface (AWS CLI) ou o console do Amazon S3.

O Simple Storage Service (Amazon S3) também fornece operações de API ser compatível com a configuração de regras de replicação. Para obter mais informações, consulte os tópicos a seguir na Referência da API do Amazon Simple Storage Service:

- Replicação do PUT Bucket
- Replicação do GET Bucket
- DELETE replicação de bucket

Tópicos

- [Configuração de replicação \(p. 808\)](#)
- [Configuração de permissões \(p. 818\)](#)
- [Demonstrações: exemplos para configurar a replicação \(p. 822\)](#)

Configuração de replicação

O Amazon S3 armazena uma configuração de replicação como XML. No arquivo XML de configuração da replicação, você especifica uma função do 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 de replicação. O Amazon S3 assume a função do IAM para replicar objetos em seu nome. Primeiro você precisa fornecer as permissões necessárias à função do IAM. Para obter mais informações sobre como gerenciar permissões, consulte [Configuração de permissões \(p. 818\)](#).

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 adicionará várias regras a uma configuração de replicação, se desejar selecionar um subgrupo diferente de objetos. Em cada regra, você especifica um filtro que seleciona um subgrupo diferente de objetos. Por exemplo, talvez você queira replicar objetos com os prefixos de chaves `tax/` ou `document/`. Para fazer isso, você adiciona duas regras, uma que especifica o filtro de prefixo das chaves `tax/` e outro que especifica o prefixo das chaves `document/`.

As seções a seguir fornecem informações adicionais.

Tópicos

- [Configuração da regra básica \(p. 809\)](#)
- [Opcional: Especificação de um filtro \(p. 809\)](#)
- [Configurações adicionais de destino \(p. 811\)](#)
- [Exemplo de configurações de replicação \(p. 814\)](#)

- [Compatibilidade retroativa \(p. 817\)](#)

Configuração da regra básica

Cada regra deve incluir o status e a prioridade da regra. A regra também precisa indicar se os marcadores de exclusão devem ser replicados.

- Status indica se a regra está habilitada ou desabilitada usando os valores Enabled ou Disabled. Se uma regra estiver desabilitada, o Amazon S3 não executará as ações especificadas nela.
- Priority indica qual regra tem precedência sempre que duas ou mais regras de replicação entram em conflito. O Simple Storage Service (Amazon S3) tenta replicar objetos de acordo com todas as regras de replicação. Porém, se houver duas ou mais regras com o mesmo bucket de destino, os objetos serão replicados de acordo com a regra com a prioridade mais alta. Quanto maior o número, maior a prioridade.
- DeleteMarkerReplication indica se os marcadores de exclusão devem ser replicados usando os valores Enabled ou Disabled.

Na configuração de destino, você deve fornecer o nome do bucket ou buckets onde quer que o Amazon S3 replique objetos.

O exemplo a seguir mostra os requisitos mínimos para uma regra da V2. Para compatibilidade com versões anteriores, o Amazon S3 continua a oferecer compatibilidade com o formato XML V1. Para obter mais informações, consulte [Compatibilidade retroativa \(p. 817\)](#).

```
...
<Rule>
    <ID>Rule-1</ID>
    <Status>Enabled-or-Disabled</Status>
    <Filter>
        <Prefix></Prefix>
    </Filter>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
        <Status>Enabled-or-Disabled</Status>
    </DeleteMarkerReplication>
    <Destination>
        <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET</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 pelo prefixo das chaves do objeto, as etiquetas do objeto ou uma combinação de ambos. Se você filtrar tanto pelo prefixo das chaves quanto pelas etiquetas do 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 etiquetas específicas.

Filtro baseado no prefixo da chave de objeto

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>
...
```

Filtrar com base em tags de objeto

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>
...
```

Filtrar com um prefixo de chave e tags de objeto

Para especificar um filtro de regra com uma combinação de um prefixo de chaves e tags de objeto, use o código a seguir. Você envolve esses filtros em um elemento pai `And`. O Amazon S3 executa uma operação lógica AND para combinar esses filtros. Em outras palavras, a regra se aplica ao subgrupo de objetos com as etiquetas e o prefixo de chave específicos.

```
<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>
...
```

Note

Se você especificar uma regra com uma etiqueta de filtro vazia, a regra será aplicada a todos os objetos do bucket.

Configurações adicionais de destino

Na configuração de destino, especifique o bucket onde você quer que o Amazon S3 replique objetos. Você pode definir configurações para replicar objetos de um bucket de origem para um bucket de destino.

```
...  
<Destination>  
    <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET</Bucket>  
</Destination>  
...
```

Você pode adicionar as seguintes opções no elemento `<Destination>`.

Tópicos

- [Especificar classe de armazenamento \(p. 811\)](#)
- [Adicionar vários buckets de destino \(p. 811\)](#)
- [Especificar parâmetros diferentes para cada regra de replicação com vários buckets de destino \(p. 812\)](#)
- [Alterar a propriedade da réplica \(p. 812\)](#)
- [Ativar o controle do tempo de replicação do S3 \(p. 813\)](#)
- [Replicar objetos criados com criptografia do lado do servidor com o uso do AWS KMS. \(p. 813\)](#)

Especificar classe de armazenamento

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.

```
...  
<Destination>  
    <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET</Bucket>  
    <StorageClass>storage-class</StorageClass>  
</Destination>  
...
```

Adicionar vários buckets de destino

Você pode adicionar vários buckets de destino em uma única configuração de replicação, como segue.

```
...  
<Rule>  
    <ID>Rule-1</ID>  
    <Status>Enabled-or-Disabled</Status>  
    <Priority>integer</Priority>  
    <DeleteMarkerReplication>  
        <Status>Enabled-or-Disabled</Status>  
    </DeleteMarkerReplication>  
    <Destination>  
        <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>  
    </Destination>  
</Rule>  
<Rule>  
    <ID>Rule-2</ID>  
    <Status>Enabled-or-Disabled</Status>
```

```
<Priority>integer</Priority>
<DeleteMarkerReplication>
    <Status>Enabled-or-Disabled</Status>
</DeleteMarkerReplication>
<Destination>
    <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET2</Bucket>
</Destination>
</Rule>
...
...
```

Especificar parâmetros diferentes para cada regra de replicação com vários buckets de destino

Ao adicionar vários buckets de destino em uma única configuração de replicação, você pode especificar parâmetros diferentes para cada regra de replicação, da seguinte forma.

```
...
<Rule>
    <ID>Rule-1</ID>
    <Status>Enabled-or-Disabled</Status>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
        <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Metrics>
        <Status>Enabled</Status>
        <EventThreshold>
            <Minutes>15</Minutes>
        </EventThreshold>
    </Metrics>
    <Destination>
        <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>
    </Destination>
</Rule>
<Rule>
    <ID>Rule-2</ID>
    <Status>Enabled-or-Disabled</Status>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
        <Status>Enabled</Status>
    </DeleteMarkerReplication>
    <Metrics>
        <Status>Enabled</Status>
        <EventThreshold>
            <Minutes>15</Minutes>
        </EventThreshold>
    </Metrics>
    <ReplicationTime>
        <Status>Enabled</Status>
        <Time>
            <Minutes>15</Minutes>
        </Time>
    </ReplicationTime>
    <Destination>
        <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET2</Bucket>
    </Destination>
</Rule>
...
...
```

Alterar a propriedade da réplica

Quando os buckets de origem e de destino não pertencem às mesmas contas, você pode alterar a propriedade da réplica para a Conta da AWS que for a proprietária do bucket de destino. Para fazer isso, adicione o elemento `AccessControlTranslation`. Esse elemento assume o valor de `Destination`.

```
...
<Destination>
  <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET</Bucket>
  <Account>destination-bucket-owner-account-id</Account>
  <AccessControlTranslation>
    <Owner>Destination</Owner>
  </AccessControlTranslation>
</Destination>
...
...
```

Se você não adicionar o elemento `AccessControlTranslation` à configuração de replicação, as réplicas pertencerão à mesma Conta da AWS que for a proprietária do objeto de origem. Para obter mais informações, consulte [Alterar o proprietário da réplica \(p. 864\)](#).

Ativar o controle do tempo de replicação do S3

Você pode habilitar o Controle do tempo de replicação do S3 (S3 RTC) na configuração de replicação. O S3 RTC replica a maioria dos objetos em segundos e 99,99 por cento dos objetos em 15 minutos (baseado em um acordo de nível de serviço).

Note

Somente um valor de `<Minutes>15</Minutes>` é aceito para `EventThreshold` e `Time`.

```
...
<Destination>
  <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET</Bucket>
  <Metrics>
    <Status>Enabled</Status>
    <EventThreshold>
      <Minutes>15</Minutes>
    </EventThreshold>
  </Metrics>
  <ReplicationTime>
    <Status>Enabled</Status>
    <Time>
      <Minutes>15</Minutes>
    </Time>
  </ReplicationTime>
</Destination>
...
...
```

Para obter mais informações, consulte [Atendimento aos requisitos de conformidade usando o Controle do tempo de replicação do S3 \(S3 RTC\) \(p. 860\)](#). Para obter exemplos de API, consulte `PutBucketReplication` na Referência da API do Amazon Simple Storage Service.

Replicar objetos criados com criptografia do lado do servidor com o uso do AWS KMS.

Seu bucket de origem pode conter objetos que foram criados com criptografia no servidor usando as chaves do AWS Key Management Service (AWS KMS) (SSE-KMS). Por padrão, o Amazon S3 não replica esses objetos. Opcionalmente, você pode direcionar o Amazon S3 para replicar esses objetos. Para fazer isso, primeiro opte explicitamente por esse recurso adicionando o elemento `SourceSelectionCriteria`. Depois, forneça a AWS KMS key (para a Região da AWS do bucket de destino) a ser usada para criptografar réplicas de objetos. Os exemplo a seguir mostra como especificar esses elementos.

```
...
<SourceSelectionCriteria>
  <SseKmsEncryptedObjects>
```

```
<Status>Enabled</Status>
</SseKmsEncryptedObjects>
</SourceSelectionCriteria>
<Destination>
  <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET</Bucket>
  <EncryptionConfiguration>
    <ReplicaKmsKeyID>AWS KMS key ID to use for encrypting object replicas</ReplicaKmsKeyID>
  </EncryptionConfiguration>
</Destination>
...
```

Para obter mais informações, consulte [Replicação de objetos criados com criptografia no lado do servidor \(SSE\) usando chaves do KMS \(p. 867\)](#).

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 transmitir uma função a um AWS service \(Serviço da AWS\)](#) 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 único para réplicas de objetos. O valor de `Status` de `Enabled` 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::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>

    <Destination><Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET</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::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
      <Status>string</Status>
    </DeleteMarkerReplication>
```

```
<Filter>
    <Prefix>Tax/</Prefix>
</Filter>

<Destination><Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET</Bucket></Destination>

</Rule>
</ReplicationConfiguration>
```

Se você especificar o elemento **Filter**, inclua também os elementos **Priority** e **DeleteMarkerReplication**. Nesse exemplo, **Priority** é irrelevante porque só há 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. Especificamente, ela se aplica ao objeto com o prefixo **Tax/** nos nomes das chaves e às duas etiquetas de objeto especificadas. A **Priority** não se aplica porque há apenas uma regra.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Role>arn:aws:iam::account-id:role/role-name</Role>
    <Rule>
        <Status>Enabled</Status>
        <Priority>1</Priority>
        <DeleteMarkerReplication>
            <Status>string</Status>
        </DeleteMarkerReplication>

        <Filter>
            <And>
                <Prefix>Tax/</Prefix>
                <Tag>
                    <Tag>
                        <Key>tagA</Key>
                        <Value>valueA</Value>
                    </Tag>
                </Tag>
                <Tag>
                    <Tag>
                        <Key>tagB</Key>
                        <Value>valueB</Value>
                    </Tag>
                </Tag>
            </And>
        </Filter>

        <Destination><Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET</Bucket></Destination>
    </Rule>
</ReplicationConfiguration>
```

Você pode especificar uma classe de armazenamento para as réplicas conforme a seguir.

```
<?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:::DOC-EXAMPLE-BUCKET</Bucket>
            <StorageClass>storage-class</StorageClass>
        </Destination>
    </Rule>
</ReplicationConfiguration>
```

```
</Destination>
</Rule>
</ReplicationConfiguration>
```

Você pode especificar uma classe de armazenamento à qual o Amazon S3 ofereça suporte.

Example 2: Configuração de replicação com duas regras

Example

Na seguinte configuração de replicação:

- Cada regra filtra em um prefixo de chave diferente para que cada regra se aplique a um subconjunto distinto de objetos. Nesse exemplo, o Simple Storage Service (Amazon S3) replica objetos com os nomes de chave *Tax/doc1.pdf* e *Project/project1.txt*, mas não replica objetos com o nome da 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 S3 Standard – IA para réplicas de objetos. O Amazon S3 usa a classe de armazenamento especificada para essas réplicas de objetos.

```
<?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:::DOC-EXAMPLE-BUCKET1</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:::DOC-EXAMPLE-BUCKET1</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 os prefixos das chaves sobrepostos, **star** e **starship**. As duas regras se aplicam aos objetos com o nome de chave **starship-x**. Nesse caso, o Amazon S3 usa a prioridade da regra para determinar qual regra deve ser aplicada. Quanto maior o número, maior a prioridade.

```
<ReplicationConfiguration>
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
      <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
      <Prefix>star</Prefix>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>
    </Destination>
  </Rule>
  <Rule>
    <Status>Enabled</Status>
    <Priority>2</Priority>
    <DeleteMarkerReplication>
      <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
      <Prefix>starship</Prefix>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Example 4: Demonstrações de exemplo

Para obter exemplos de demonstrações, consulte [Demonstrações: exemplos para configurar a replicação \(p. 822\)](#).

Para obter mais informações sobre a estrutura XML da configuração de replicação, consulte [PutBucketReplication](#) na Referência da API do Amazon Simple Storage Service.

Compatibilidade retroativa

A versão mais recente do XML de configuração da replicação é V2. As configurações de replicação XML V2 são aquelas que contêm o elemento `Filter` para regras e regras que especificam o Controle do tempo de replicação do S3 (S3 RTC).

Para ver a versão da configuração de replicação, você pode usar a operação de API `GetBucketReplication`. Para obter mais informações, consulte [GetBucketReplication](#) na Amazon Simple Storage Service API Reference (Referência da API do Amazon Simple Storage Service).

Para compatibilidade com versões anteriores, o Amazon S3 continua a oferecer suporte à configuração de replicação XML V1. Se você usou a configuração da replicação XML V1, 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 de 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 configuração de replicação do XML V1 oferecia compatibilidade com a filtragem com base somente no prefixo da chave. Nesse caso, você adiciona o `Prefix` diretamente como elemento-filho do elemento `Rule`, conforme o exemplo a seguir.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Prefix>key-prefix</Prefix>
    <Destination><Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET</Bucket></Destination>

  </Rule>
</ReplicationConfiguration>
```

Para compatibilidade com versões anteriores, o Amazon S3 continua a oferecer compatibilidade com a 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 Simple Storage Service (Amazon S3) replicará os marcadores de exclusão que resultaram de ações do usuário. Em outras palavras: o Amazon S3 só replica o marcador de exclusão se um usuário excluir um objeto. Se um objeto expirado for removido pelo Amazon S3 (como parte de uma ação de ciclo de vida), ele não replicará o marcador de exclusão.

Nas configurações de replicação de V2, você pode habilitar a replicação de marcadores de exclusão para regras com base em tags. Para obter mais informações, consulte [Replicação de marcadores de exclusão entre intervalos \(p. 862\)](#).

Configuração de permissões

Ao configurar a replicação, é preciso adquirir as permissões necessárias do seguinte modo:

- O Simple Storage Service (Amazon S3) precisa de permissões para replicar objetos em seu nome. Você concede essas permissões criando uma função do IAM e especificando essa função na configuração da replicação.
- Quando os buckets de origem e de destino não pertencem à mesma conta, o proprietário do bucket de destino deve conceder ao proprietário do bucket de origem permissões para armazenar réplicas.

Tópicos

- [Criar um perfil do IAM \(p. 818\)](#)
- [Conceder permissões quando os buckets de origem e de destino pertencerem a Contas da AWS diferentes \(p. 820\)](#)
- [Alterar a propriedade da réplica \(p. 822\)](#)
- [Habilitar o recebimento de objetos replicados de um bucket de origem \(p. 822\)](#)

Criar um perfil do IAM

Por padrão, todos os recursos do Simple Storage Service (Amazon S3): buckets, objetos e sub-recursos relacionados, são privados, e somente o proprietário do recurso pode acessá-lo. O Amazon S3 precisa

de permissões de leitura e replicação de objetos a partir do bucket de origem. Você concede essas permissões criando uma função do IAM e especificando a função na configuração da replicação.

Esta seção explica a política de confiança e a política de permissão mínima necessária. As demonstrações de exemplo fornecem instruções passo a passo para criar uma função do IAM. Para obter mais informações, consulte [Demonstrações: exemplos para configurar a replicação \(p. 822\)](#).

- O exemplo a seguir mostra uma política de confiança na qual você identifica o Simple Storage Service (Amazon S3) como a entidade principal do 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.

- O exemplo a seguir mostra uma política de acesso em que você concede à função permissões para realizar as tarefas de replicação em seu nome. Quando o Amazon S3 assumir a função, ele terá as permissões que você especificar nessa política. Nesta política, **DOC-EXAMPLE-BUCKET1** é o bucket de origem e **DOC-EXAMPLE-BUCKET2** é o bucket de destino.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetReplicationConfiguration",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetObjectVersionForReplication",  
                "s3:GetObjectVersionAcl",  
                "s3:GetObjectVersionTagging"  
            ],  
            "Resource": [  
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:ReplicateObject",  
                "s3:ReplicateDelete",  
                "s3:ReplicateTags"  
            ],  
        }  
    ]  
}
```

```
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET2/*"
    }
}
```

A política de acesso concede permissões às seguintes ações:

- **s3:GetReplicationConfiguration** e **s3>ListBucket**: as permissões para essas ações no bucket **DOC-EXAMPLE-BUCKET1** (o bucket de origem) permitem que o Amazon S3 recupere a configuração de replicação e liste o conteúdo do bucket. (O modelo de permissões atual exige a permissão **s3>ListBucket** para acessar marcadores de exclusão.)
- **s3GetObjectVersionForReplication** e **s3GetObjectVersionAcl**: as permissões para essas ações são concedidas em todos os objetos para permitir que o Simple Storage Service (Amazon S3) obtenha uma versão específica dos objetos e as listas de controle de acesso (ACL) associadas aos objetos.
- **s3:ReplicateObject** e **s3:ReplicateDelete**: as permissões para essas ações em todos os objetos no bucket **DOC-EXAMPLE-BUCKET2** (o bucket de destino) permitem que o Amazon S3 replique objetos ou marcadores de exclusão no bucket de destino. Para obter mais informações sobre marcadores de exclusão, consulte [Como a exclusão de operações afeta a replicação \(p. 805\)](#).

Note

As permissões para a ação **s3:ReplicateObject** no bucket **DOC-EXAMPLE-BUCKET2** (o bucket de destino) também permitem a replicação de etiquetas de objeto. Por isso, você não precisa conceder permissões explicitamente à ação **s3:ReplicateTags**.

- **s3GetObjectVersionTagging**: as permissões para essa ação em objetos no bucket **DOC-EXAMPLE-BUCKET1** permitem que o Amazon S3 leia etiquetas de objeto para replicação. Para obter mais informações, consulte [Categorizando seu armazenamento usando tags \(p. 878\)](#). Se o Amazon S3 não tiver essas permissões, ele replicará os objetos, mas não as tags do objeto.

Para obter uma lista de ações do Amazon S3, consulte [Ações do Amazon S3 \(p. 441\)](#).

Important

Mais especificamente, a Conta da AWS proprietária da função do IAM precisa ter permissões para as ações que ela concede à função do IAM.

Por exemplo, suponha que o bucket de origem contenha objetos pertencentes a uma outra Conta da AWS. O proprietário dos objetos deverá conceder explicitamente as permissões exigidas à Conta da AWS proprietária da função do IAM por meio da ACL do objeto. Caso contrário, o Amazon S3 não conseguirá acessar os objetos e haverá falha na replicação dos objetos. Para obter informações sobre as permissões da ACL, consulte [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#).

As permissões descritas aqui são relativas à configuração mínima da replicação. Se optar por adicionar configurações de replicação opcionais, será necessário conceder permissões adicionais ao Amazon S3. Para obter mais informações, consulte [Configurações de replicação adicionais \(p. 857\)](#).

Conceder permissões quando os buckets de origem e de destino pertencerem a Contas da AWS diferentes

Quando os buckets de origem e de destino não pertencem à mesma conta, o proprietário do bucket de destino também deve adicionar uma política de bucket para conceder ao proprietário do bucket de origem permissões para realizar as ações de replicação, como se segue. Nesta política, **DOC-EXAMPLE-BUCKET2** é o bucket de destino.

```
{
    "Version": "2012-10-17",
```

```
"Id":"PolicyForDestinationBucket",
"Statement": [
    {
        "Sid":"Permissions on objects",
        "Effect":"Allow",
        "Principal":{
            "AWS":"arn:aws:iam::SourceBucket-account-ID:role/service-role/source-account-
IAM-role"
        },
        "Action":[
            "s3:ReplicateDelete",
            "s3:ReplicateObject"
        ],
        "Resource":"arn:aws:s3:::DOC-EXAMPLE-BUCKET2/*"
    },
    {
        "Sid":"Permissions on bucket",
        "Effect":"Allow",
        "Principal":{
            "AWS":"arn:aws:iam::SourceBucket-account-ID:role/service-role/source-account-
IAM-role"
        },
        "Action": [
            "s3>List*",
            "s3:GetBucketVersioning",
            "s3:PutBucketVersioning"
        ],
        "Resource":"arn:aws:s3:::DOC-EXAMPLE-BUCKET2"
    }
]
```

Para ver um exemplo, consulte [Configurar a replicação quando os buckets de origem e destino pertencerem a contas diferentes \(p. 835\)](#).

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 (pela função do IAM), o Amazon S3 replicará as tags com os objetos. Para obter informações sobre a função do IAM, consulte [Criar um perfil do IAM \(p. 818\)](#).
- Se o proprietário do bucket de destino não quiser replicar as tags, ele poderá adicionar a seguinte instrução à política do bucket de destino para negar explicitamente a permissão para a ação `s3:ReplicateTags`. Nesta política, `DOC-EXAMPLE-BUCKET2` é o bucket de destino.

```
...
"Statement": [
    {
        "Effect":"Deny",
        "Principal":{
            "AWS":"arn:aws:iam::SourceBucket-account-id:role/service-role/source-account-
IAM-role"
        },
        "Action":"s3:ReplicateTags",
        "Resource":"arn:aws:s3:::DOC-EXAMPLE-BUCKET2/*"
    }
]
...
```

Alterar a propriedade da réplica

Quando diferentes Contas da AWS são proprietárias dos buckets de origem e de destino, é possível instruir o Amazon S3 para alterar a propriedade da réplica para a Conta da AWS proprietária do bucket de destino. Para obter mais informações sobre a substituição do proprietário, consulte [Alterar o proprietário da réplica \(p. 864\)](#).

Habilitar o recebimento de objetos replicados de um bucket de origem

Você pode gerar rapidamente as políticas necessárias para permitir o recebimento de objetos replicados de um bucket de origem no AWS Management Console.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, escolha o bucket que você deseja usar como bucket de destino.
4. Escolha Management (Gerenciamento) e role para baixo até Replication rules (Regras de replicação).
5. Em Actions (Ações), escolha Receive replicated objects (Receber objetos replicados).

- Siga as instruções e insira o ID da Conta da AWS do bucket de origem e escolha Generate policies (Gerar políticas). Serão geradas uma política de bucket do Amazon S3 e uma política de chave do KMS.
6. Para adicionar essa política à política de bucket existente, escolha Apply settings (Aplicar configurações) ou Copy (Copiar) para copiar manualmente as alterações.
 7. (Opcional) Copie a política AWS KMS em sua política de chave do KMS desejada no console do AWS Key Management Service.

Demonstrações: exemplos para configurar a replicação

Os exemplos a seguir mostram como configurar a replicação em tempo real para casos de uso comuns. Os exemplos demonstram a configuração de replicação usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI) e os AWS SDKs (exemplos de Java e .NET SDK são mostrados). Para obter informações sobre como instalar e configurar a AWS CLI, consulte os tópicos a seguir no Guia do usuário da AWS Command Line Interface

Note

Replicação em tempo real refere-se à Same-Region Replication (SRR – Replicação na Mesma Região) e à Cross-Region Replication (CRR – Replicação entre Regiões). Para uma ação de replicação sob demanda para sincronizar buckets e replicar objetos existentes, consulte [Replicar objetos existentes \(p. 849\)](#).

- [Instalar o AWS Command Line Interface](#)
- [Configurar a AWS CLI](#): é necessário configurar pelo menos um perfil. Se você estiver explorando cenários entre contas, configure dois perfis.

Para obter informações sobre AWS SDKs, consulte [AWS SDK para Java](#) e [AWS SDK for .NET](#).

Tópicos

- [Configuração da replicação para buckets de origem e de destino pertencentes à mesma conta \(p. 823\)](#)
- [Configurar a replicação quando os buckets de origem e destino pertencerem a contas diferentes \(p. 835\)](#)

- Alterar o proprietário da réplica quando os buckets de origem e de destino forem de propriedade de contas diferentes (p. 836)
- Replicar objetos criptografados (p. 841)
- Replicação de objetos com o Controle de Tempo de Replicação do S3 (S3 RTC) (p. 846)
- Gerenciamento de regras de replicação usando o console do Amazon S3 (p. 848)

Configuração da replicação para buckets de origem e de destino pertencentes à mesma conta

A replicação é a cópia assíncrona automática de objetos em buckets na mesma região ou em diferentes Regiões da AWS. A replicação copia os objetos recém-criados e as atualizações de objeto de um bucket de origem para um bucket de destino. Para obter mais informações, consulte [Replicação de objetos \(p. 801\)](#).

Ao configurar a replicação, você adiciona regras de replicação ao bucket de origem. As regras de replicação definem quais objetos do bucket de origem devem ser replicados e o bucket de destino ou buckets nos quais os objetos replicados são armazenados. Você pode criar uma regra para replicar todos os objetos dentro de um bucket ou um subgrupo de objetos com um prefixo específico de nome de chaves, uma ou mais tags de objetos ou ambos. Um bucket de destino pode estar na mesma Conta da AWS que o bucket de origem ou pode estar em uma conta diferente.

Se você especificar um ID da versão do objeto a ser excluído, o Amazon S3 excluirá essa versão do objeto no bucket de origem. No entanto, ele não replica 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.

Quando você adiciona uma regra de replicação a um bucket, ela fica ativada por padrão, portanto, começa a funcionar assim que é salva.

Neste exemplo, você configura a replicação para os buckets de origem e destino que pertencem à mesma Conta da AWS. São apresentados exemplos de uso do console do Amazon S3, da AWS Command Line Interface (AWS CLI), do AWS SDK for Java e do AWS SDK for .NET.

Uso do console do S3

Siga essas etapas para configurar uma replicação quando o bucket de destino estiver na mesma Conta da AWS que o bucket de origem.

Se o bucket de destino estiver em uma conta diferente do bucket de origem, você deverá adicionar uma política ao bucket de destino. Assim, será possível conceder ao proprietário da conta do bucket de origem permissão para replicar objetos no bucket de destino. Para obter mais informações, consulte [Conceder permissões quando os buckets de origem e de destino pertencerem a Contas da AWS diferentes \(p. 820\)](#).

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket desejado.
3. Escolha Management (Gerenciamento), role para baixo até Replication rules (Regras de replicação) e escolha Create replication rule (Criar regra de replicação).
4. Em Rule name (Nome da regra), insira um nome para a regra, ajudando a identificá-la posteriormente. O nome é obrigatório e precisa ser exclusivo dentro do bucket.
5. Configure uma função do AWS Identity and Access Management (IAM) que o Amazon S3 possa assumir para replicar objetos em seu nome.

Para configurar uma função do IAM, na seção Replication rule configuration (Configuração da regra de replicação), em IAM role (Função do IAM), siga um destes procedimentos:

- Recomendamos que você escolha Create new role (Criar nova função) para que o Amazon S3 crie uma nova função do IAM para você. Quando você salva a regra, uma nova política é gerada para a função do IAM que coincide com os buckets de origem e de destino que você escolher.
- Você pode usar uma função do IAM existente. Se fizer isso, escolha uma função que conceda ao Amazon S3 as permissões necessárias para a replicação. A replicação falhará se essa função não conceder ao Amazon S3 permissões suficientes para seguir sua regra de replicação.

Important

Quando você adiciona uma regra de replicação a um bucket, você deve ter a permissão `iam:PassRole` para poder transmitir a função do IAM que concede permissões de replicação do Amazon S3. Para obter mais informações, consulte [Conceder permissões ao usuário para passar uma função a um serviço da](#) no Guia do usuário do IAM.

6. Em Status, Enabled (Habilitado) é selecionado por padrão. Uma regra ativada começa a funcionar assim que você a salva. Se você quiser habilitar a regra posteriormente, selecione Disabled (Desabilitado).
7. Se o bucket tiver regras de replicação existentes, você será instruído a definir uma prioridade para a regra. Defina uma prioridade para a regra, de maneira a evitar conflitos causados pelos objetos incluídos no escopo de mais de uma regra. No caso de sobreposição de regras, o Amazon S3 usa a prioridade da regra para determinar qual regra aplicar. Quanto maior o número, maior a prioridade. Para obter mais informações sobre prioridade de regra, consulte [Configuração de replicação \(p. 808\)](#).
8. Na Replication rule configuration (Configuração da regra de replicação), em Source bucket (Bucket de origem), você tem as seguintes opções para definir a origem da replicação:
 - Para replicar todo o bucket, escolha This rule applies to all objects in the bucket (Esta regra se aplica a todos os objetos no bucket).
 - Para replicar todos os objetos que tenham o mesmo prefixo, escolha Limit the scope of this rule using one or more filters (Limitar o escopo desta regra usando um ou mais filtros). Isso limita a replicação a todos os objetos que tenham nomes que começam com a string (por exemplo, `pictures`). Insira um prefixo na caixa.

Note

Se você inserir um prefixo que seja o nome de uma pasta, é preciso usar / (barra) como o último caractere (por exemplo, `pictures/`).

- Para replicar todos os objetos com uma ou mais tags de objeto, selecione Add tag (Adicionar tag) e insira o par de valores de chave nas caixas. Repita o procedimento para adicionar outra tag. Você pode combinar um prefixo e tags. Para obter mais informações sobre etiquetas de objeto, consulte [Categorizando seu armazenamento usando tags \(p. 878\)](#).

O novo esquema é compatível com os filtros de prefixo e tags e com a priorização das regras. Para obter mais informações sobre o novo esquema, consulte [Compatibilidade retroativa \(p. 817\)](#).

Para obter mais informações sobre o XML usado com a API do Amazon S3 que funciona atrás da interface do usuário, consulte [Configuração de replicação \(p. 808\)](#). O novo esquema é descrito como configuração de replicação XML V2.

9. Em Destination (Destino), selecione o bucket no qual você deseja que o Amazon S3 replique objetos.

Note

O número de buckets de destino é limitado ao número de Regiões da AWS em determinada partição. Uma partição é um agrupamento de regiões. A AWS atualmente tem três partições: aws (regiões Standard), aws-cn (regiões da China) e aws-us-gov (regiões GovCloud [EUA])

da AWS). Você pode usar [cotas de serviço](#) para solicitar um aumento no limite de bucket de destino.

- Para replicar para um bucket ou buckets em sua conta, selecione Choose a bucket in this account (Escolher um bucket nesta conta) e digite ou procure o bucket de destino.
- Para replicar para um ou mais buckets em uma Conta da AWS diferente, selecione Choose a bucket in another account (Escolher um bucket em outra conta) e insira o ID e o nome da conta do bucket de destino.

Se o destino estiver em uma conta diferente do bucket de origem, você deverá adicionar uma política de bucket aos buckets de destino para conceder ao proprietário da conta do bucket de origem permissão para replicar objetos. Para obter mais informações, consulte [Conceder permissões quando os buckets de origem e de destino pertencerem a Contas da AWS diferentes](#) (p. 820).

Note

Se o versionamento não estiver ativado no bucket de destino, você receberá uma advertência que contém um botão Enable versioning (Habilitar versionamento). Escolha esse botão para habilitar o versionamento no bucket.

10. Você tem as seguintes opções adicionais ao definir o Destination (Destino):

- Se você quiser habilitar a Object Ownership (Propriedade do objeto) para ajudar a padronizar a propriedade de novos objetos no bucket de destino, escolha Change object ownership to the destination bucket owner (Alterar propriedade do objeto para o proprietário do bucket de destino). Para obter mais informações sobre essa opção, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#) (p. 638).
- Se você quiser replicar seus dados a uma classe de armazenamento específica no destino, escolha Change the storage class for the replicated objects (Alterar a classe de armazenamento dos objetos replicados). Em seguida, escolha a classe de armazenamento que você deseja usar para os objetos replicados no destino. Se você não selecionar essa opção, a classe de armazenamento para objetos replicados será a mesma classe dos objetos originais.
- Se quiser habilitar a replicação de marcadores de exclusão na configuração de replicação, selecione Delete marker replication (Excluir replicação de marcador). Para obter mais informações, consulte [Replicação de marcadores de exclusão entre intervalos](#) (p. 862).
- Se você quiser habilitar a sincronização de modificação de réplica do Amazon S3 em sua configuração de replicação, selecione Replica modification sync (Sincronização de modificação de réplica). Para obter mais informações, consulte [Replicação de alterações de metadados com sincronização de modificação de réplica do Amazon S3](#) (p. 863).
- Se você quiser habilitar métricas de replicação do S3 na configuração de replicação, selecione Replication metrics and events (Métricas e eventos de replicação). Para obter mais informações, consulte [Monitoramento do progresso com métricas de replicação e notificações de eventos do Amazon S3](#) (p. 857).
- Se você quiser habilitar o Controle do tempo de replicação do S3 (S3 RTC) na configuração de replicação, selecione S3 Replication Time Control (Controle do tempo de replicação do S3). Para obter mais informações sobre essa opção, consulte [Atendimento aos requisitos de conformidade usando o Controle do tempo de replicação do S3 \(S3 RTC\)](#) (p. 860).

Note

Quando você usa métricas de replicação S3 RTC ou S3, aplicam-se taxas adicionais.

11. Para replicar objetos no bucket de origem que sejam criptografados com o AWS Key Management Service (AWS KMS), em Replication criteria (Critérios de replicação), selecione Replicate objects encrypted with AWS KMS (Replicar objetos criptografados com o AWS KMS). Em chave de AWS KMS

Versão da API 2006-03-01

para criptografar objetos de destino estão as chaves de origem que você permite que a replicação use. Todas as chaves de origem do KMS são incluídas por padrão. Você pode optar por refinar a seleção de chaves do KMS.

Os objetos criptografados pelas AWS KMS keys que você não seleciona não são replicados. A chave do KMS ou um grupo de chaves do KMS é escolhido para você, mas você pode escolher as chaves do KMS, se desejar. Para obter informações sobre como usar o AWS KMS com replicação, consulte [Replicação de objetos criados com criptografia no lado do servidor \(SSE\) usando chaves do KMS \(p. 867\)](#).

Important

Ao replicar objetos que estão criptografados com AWS KMS, a taxa de solicitações do AWS KMS sobra na Região de origem e aumenta na Região de destino pelo mesmo valor. Essas taxas de chamada aprimoradas para AWS KMS se devem à forma com que os dados são recriptografados usando a chave do KMS que você define na região de destino de replicação. O AWS KMS tem um limite de taxas de solicitação por conta de chamada por região. Para obter informações sobre os padrões de limite, consulte [Limites do AWS KMS – solicitações por segundo: varia](#) no Guia do desenvolvedor do AWS Key Management Service.

Se a taxa de solicitações do objeto PUT do Amazon S3 durante a replicação for maior que a metade do limite da taxa padrão do AWS KMS para sua conta, recomendamos que você solicite um aumento do limite de taxa de solicitação do AWS KMS. Para solicitar um aumento, crie um caso no AWS Support Center em [Entre em contato conosco](#). Por exemplo, suponha que a sua taxa de solicitação do objeto PUT seja 1.000 solicitações por segundo e você use o AWS KMS para criptografar seus objetos. Nesse caso, recomendamos solicitar que o AWS Support aumente o limite da taxa do AWS KMS para 2,5 mil solicitações por segundo nas regiões de origem e de destino (se forem diferentes) para garantir que não haja nenhum controle de utilização por parte do AWS KMS.

Para ver a taxa de solicitação de objeto PUT no bucket de origem, visualize `PutRequests` nas métricas de solicitação do Amazon CloudWatch para o Amazon S3. Para obter informações sobre como visualizar métricas do CloudWatch, consulte [Uso do console do S3 \(p. 1079\)](#)

Se você optar por replicar objetos criptografados com o AWS KMS, insira o nome de recurso da Amazon (ARN) da AWS KMS key para criptografar réplicas no bucket de destino. Você pode encontrar o ARN da chave do KMS no console do IAM em Encryption keys (Chaves de criptografia). Outra possibilidade é escolher o nome da chave do KMS na lista suspensa.

Para obter mais informações sobre como criar uma AWS KMS key, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

Important

O console do Amazon S3 lista apenas 100 chaves do KMS por região da AWS. Se você tiver mais de 100 chaves do KMS na mesma região, será possível ver somente as primeiras 100 chaves do KMS no console do S3. Para usar uma chave do KMS que não esteja listada no console, escolha Custom KMS ARN (Personalizar o ARN do KMS) e insira o ARN da chave do KMS.

12. Para terminar, escolha Save (Salvar).
13. Depois de salvar sua regra, você pode editar, ativar, desativar ou excluir sua regra selecionando sua regra e escolhendo Edit rule (Editar regra).

Usar a AWS CLI

Para configurar a AWS CLI para configurar a replicação quando os buckets de origem e de destino pertencem à mesma Conta da AWS, faça o seguinte:

- Crie os buckets de origem e de destino

- Habilite o versionamento nos buckets
- Crie uma função do IAM que dê ao Simple Storage Service (Amazon S3) permissão para replicar objetos
- Adicione uma configuração de replicação ao bucket

Para verificar sua configuração, teste-a.

Para configurar a replicação quando os buckets de origem e destino pertencem à 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 passar uma função a um produto da AWS](#) no Guia do usuário do IAM. Se você usar credenciais de administrador para criar um perfil nomeado, poderá executar todas as tarefas.

2. Crie um bucket de `source` e habilite nele o versionamento. O código a seguir cria um bucket de `source` 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
```

3. Crie um bucket de `destination` e habilite nele o versionamento. O código a seguir cria um bucket de `destination` na região Oeste dos EUA (Oregon) (us-west-2).

Note

Para fazer a configuração da replicação quando os buckets de origem e de destino estiverem na mesma Conta da AWS, use o mesmo perfil. Este exemplo usa `acctA`. Para testar a configuração da replicação quando os buckets pertencerem a Contas da AWS distintas, especifique diferentes perfis para cada um. Este exemplo usa 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
```

4. Crie uma função do IAM. Você especifica essa função na configuração da replicação que adiciona ao bucket de *origem* posteriormente. O Amazon S3 assume essa função para replicar objetos em seu nome. A função do IAM é criada em duas etapas:

- Crie uma função.
- Anexar uma política de permissões à função.

- a. Crie a função do IAM.

- i. Copie a política de confiança a seguir e salve-a em um arquivo chamado `s3-role-trust-policy.json` no diretório atual do computador local. Essa política concede ao principal do serviço da 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 comando a seguir para criar uma função.

```
$ aws iam create-role \  
--role-name replicationRole \  
--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 computador local. Essa política concede permissões para várias ações de bucket e objetos do Amazon S3.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetObjectVersionForReplication",  
                "s3:GetObjectVersionAcl",  
                "s3:GetObjectVersionTagging"  
            ],  
            "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"
    ],
    "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 replicationRole \
--policy-document file://s3-role-permissions-policy.json \
--policy-name replicationRolePolicy \
--profile acctA
```

5. Adicione uma configuração de replicação ao bucket **source**.

- a. Embora a API do Amazon S3 exija que a replicação seja configurada como XML, a AWS CLI requer 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"
            }
        }
    ]
}
```

- b. Atualize o JSON fornecendo valores para o **destination-bucket** e o **IAM-role-ARN**. Salve as alterações.
c. Execute o comando a seguir para adicionar a configuração de replicação ao seu bucket de origem. Não deixe fornece o nome do bucket de **source**.

```
$ 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:

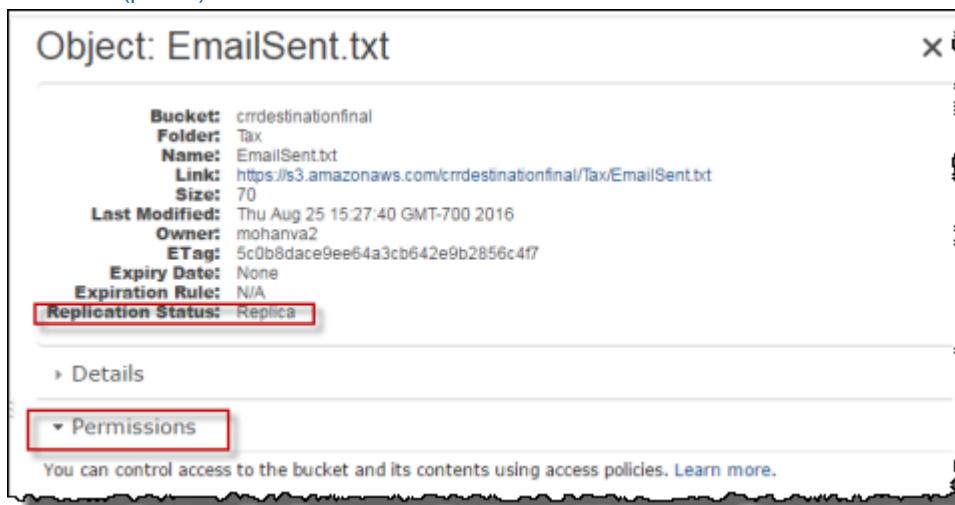
- a. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
- b. No bucket de **source**, crie uma pasta denominada **Tax**.
- c. Adicione exemplos de objetos à pasta **Tax** no bucket de **source**.

Note

O tempo necessário para o Amazon S3 replicar um objeto depende do tamanho do objeto. Para obter informações sobre como ver o status da replicação, consulte [Obtenção de informações sobre o status da replicação \(p. 873\)](#).

No bucket de **destination**, verifique o seguinte:

- Se 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 continua pertencendo ao proprietário do bucket de **source** e que o proprietário do bucket de **destination** não tem nenhuma permissão na réplica do objeto. Você pode adicionar uma configuração opcional para instruir o Amazon S3 a alterar a propriedade da réplica. Para ver um exemplo, consulte [Alterar o proprietário da réplica quando os buckets de origem e de destino forem de propriedade de contas diferentes \(p. 836\)](#).



Uso da SDKs AWS

Use os exemplos de código a seguir para adicionar uma configuração de replicação ao bucket com AWS SDK for Java e AWS SDK for .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 como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.identitymanagement.AmazonIdentityManagement;
import com.amazonaws.services.identitymanagement.AmazonIdentityManagementClientBuilder;
import com.amazonaws.services.identitymanagement.model.CreateRoleRequest;
import com.amazonaws.services.identitymanagement.model.PutRolePolicyRequest;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.BucketReplicationConfiguration;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.CreateBucketRequest;
import com.amazonaws.services.s3.model.DeleteMarkerReplication;
import com.amazonaws.services.s3.model.DeleteMarkerReplicationStatus;
import com.amazonaws.services.s3.model.ReplicationDestinationConfig;
import com.amazonaws.services.s3.model.ReplicationRule;
import com.amazonaws.services.s3.model.ReplicationRuleStatus;
import com.amazonaws.services.s3.model.SetBucketVersioningConfigurationRequest;
import com.amazonaws.services.s3.model.StorageClass;
import com.amazonaws.services.s3.model.replication.ReplicationFilter;
import com.amazonaws.services.s3.model.replication.ReplicationFilterPredicate;
import com.amazonaws.services.s3.model.replication.ReplicationPrefixPredicate;

import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class CrossRegionReplication {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String accountId = "*** Account ID ***";
        String roleName = "*** Role name ***";
        String sourceBucketName = "*** Source bucket name ***";
        String destBucketName = "*** Destination bucket name ***";
        String prefix = "Tax/";

        String roleARN = String.format("arn:aws:iam::%s:%s", accountId, roleName);
        String destinationBucketARN = "arn:aws:s3:::" + destBucketName;

        AmazonS3 s3Client = AmazonS3Client.builder()
            .withCredentials(new ProfileCredentialsProvider())
            .withRegion(clientRegion)
            .build();

        createBucket(s3Client, clientRegion, sourceBucketName);
        createBucket(s3Client, clientRegion, destBucketName);
        assignRole(roleName, clientRegion, sourceBucketName, destBucketName);

        try {

            // Create the replication rule.
            List<ReplicationFilterPredicate> andOperands = new
            ArrayList<ReplicationFilterPredicate>();
            andOperands.add(new ReplicationPrefixPredicate(prefix));

            Map<String, ReplicationRule> replicationRules = new HashMap<String,
            ReplicationRule>();
            replicationRules.put("ReplicationRule1",
                new ReplicationRule())
        }
    }
}
```

```

        .withPriority(0)
        .withStatus(ReplicationRuleStatus.Enabled)
        .withDeleteMarkerReplication(new
DeleteMarkerReplication().withStatus(DeleteMarkerReplicationStatus.DISABLED))
        .withFilter(new ReplicationFilter().withPredicate(new
ReplicationPrefixPredicate(prefix)))
        .withDestinationConfig(new ReplicationDestinationConfig()
        .withBucketARN(destinationBucketARN)
        .withStorageClass(StorageClass.Standard)));

    // Save the replication rule to the source bucket.
    s3Client.setBucketReplicationConfiguration(sourceBucketName,
        new BucketReplicationConfiguration()
        .withRoleARN(roleARN)
        .withRules(replicationRules));

    // Retrieve the replication configuration and verify that the configuration
    // matches the rule we just set.
    BucketReplicationConfiguration replicationConfig =
s3Client.getBucketReplicationConfiguration(sourceBucketName);
    ReplicationRule rule = replicationConfig.getRule("ReplicationRule1");
    System.out.println("Retrieved destination bucket ARN: " +
rule.getDestinationConfig().getBucketARN());
    System.out.println("Retrieved priority: " + rule.getPriority());
    System.out.println("Retrieved source-bucket replication rule status: " +
rule.getStatus());
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}

private static void createBucket(AmazonS3 s3Client, Regions region, String
bucketName) {
    CreateBucketRequest request = new CreateBucketRequest(bucketName,
region.getName());
    s3Client.createBucket(request);
    BucketVersioningConfiguration configuration = new
BucketVersioningConfiguration().withStatus(BucketVersioningConfiguration.ENABLED);

    SetBucketVersioningConfigurationRequest enableVersioningRequest = new
SetBucketVersioningConfigurationRequest(bucketName, configuration);
    s3Client.setBucketVersioningConfiguration(enableVersioningRequest);

}

private static void assignRole(String roleName, Regions region, String
sourceBucket, String destinationBucket) {
    AmazonIdentityManagement iamClient =
AmazonIdentityManagementClientBuilder.standard()
        .withRegion(region)
        .withCredentials(new ProfileCredentialsProvider())
        .build();
    StringBuilder trustPolicy = new StringBuilder();
    trustPolicy.append("{\\r\\n    ");
    trustPolicy.append("\\\"Version\\\":\\\"2012-10-17\\\",\\r\\n    ");
    trustPolicy.append("\\\"Statement\\\":[:\\r\\n        {\\r\\n            ");
    trustPolicy.append("\\\"Effect\\\":\\\"Allow\\\",\\r\\n            \\\"Principal\\\
\"\\\":\\\\r\\n        ");
}

```

```

        trustPolicy.append(" \\"Service\\\" : \\"s3.amazonaws.com\\\" \\\\r\\n      }, \\
    \\r\\n      ");
    trustPolicy.append(" \\"Action\\\" : \\"sts:AssumeRole\\\" \\\\r\\n      } \\\\r\\n
] \\\\r\\n}");}

CreateRoleRequest createRoleRequest = new CreateRoleRequest()
    .withRoleName(roleName)
    .withAssumeRolePolicyDocument(trustPolicy.toString());

iamClient.createRole(createRoleRequest);

StringBuilder permissionPolicy = new StringBuilder();
permissionPolicy.append(" \\\\r\\n      \\\\\"Version\\\" : \\"2012-10-17\\\", \\\\r\\n
\\\\\"Statement\\\" : [ \\\\r\\n          { \\\\r\\n              \"");
permissionPolicy.append(" \\"Effect\\\" : \\"Allow\\\", \\\\r\\n              \\\\\"Action
\\\" : [ \\\\r\\n                  \"");
permissionPolicy.append(" \\"s3:GetObjectVersionForReplication\\\", \\\\r\\n
");
permissionPolicy.append(" \\"s3:GetObjectVersionAcl\\\" \\\\r\\n      ], \\\\r\\n
\\\\\"Resource\\\" : [ \\\\r\\n                  \"");
permissionPolicy.append(" \\"arn:aws:s3:::");
permissionPolicy.append(sourceBucket);
permissionPolicy.append(" \"/*\\\" \\\\r\\n          ] \\\\r\\n          }, \\\\r\\n          { \\\\r\\
\\n              \"");
permissionPolicy.append(" \\"Effect\\\" : \\"Allow\\\", \\\\r\\n              \\\\\"Action
\\\" : [ \\\\r\\n                  \"");
permissionPolicy.append(" \\"s3>ListBucket\\\", \\\\r\\n      \\\\\"s3:GetReplicationConfiguration\\\" \\\\r\\n
");
permissionPolicy.append(" \"], \\\\r\\n          \\\\\"Resource\\\" : [ \\\\r\\n          \\\\\"arn:aws:s3:::\\
");
permissionPolicy.append(sourceBucket);
permissionPolicy.append(" \\\\r\\n          \"");
permissionPolicy.append(" \\\\r\\n          }, \\\\r\\n          { \\\\r\\n              \\\\\"Effect\\
\\\" : \\"Allow\\\", \\\\r\\n          \"");
permissionPolicy.append(" \\"Action\\\" : [ \\\\r\\n              \\\\\"s3:ReplicateObject\\\", \\\\r\\n
");
permissionPolicy.append(" \\"s3:ReplicateDelete\\\", \\\\r\\n      \\\\\"s3:ReplicateTags\\\", \\\\r\\n
");
permissionPolicy.append(" \\"s3:GetObjectVersionTagging\\\" \\\\r\\n          \\\\r\\n          ];
\\\r\\n          \"");
permissionPolicy.append(" \\"Resource\\\" : \\"arn:aws:s3:::");
permissionPolicy.append(destinationBucket);
permissionPolicy.append(" \"/*\\\" \\\\r\\n          ] \\\\r\\n          }");

PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
    .withRoleName(roleName)
    .withPolicyDocument(permissionPolicy.toString())
    .withPolicyName("crrRolePolicy");

iamClient.putRolePolicy(putRolePolicyRequest);

}
}

```

C#

O exemplo de código de AWS SDK for .NET a seguir adiciona uma configuração de replicação a um bucket e, depois, a recupera. Para usar esse código, dê nomes aos buckets e o nome de recurso da Amazon (ARN) à função do IAM. Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
```

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

namespace Amazon.DocSamples.S3
{
    class CrossRegionReplicationTest
    {
        private const string sourceBucket = "*** source bucket ***";
        // Bucket ARN example - arn:aws:s3:::destinationbucket
        private const string destinationBucketArn = "*** destination bucket ARN ***";
        private const string roleArn = "*** IAM Role ARN ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint sourceBucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        public static void Main()
        {
            s3Client = new AmazonS3Client(sourceBucketRegion);
            EnableReplicationAsync().Wait();
        }
        static async Task EnableReplicationAsync()
        {
            try
            {
                ReplicationConfiguration replConfig = new ReplicationConfiguration
                {
                    Role = roleArn,
                    Rules =
                    {
                        new ReplicationRule
                        {
                            Prefix = "Tax",
                            Status = ReplicationRuleStatus.Enabled,
                            Destination = new ReplicationDestination
                            {
                                BucketArn = destinationBucketArn
                            }
                        }
                    }
                };
                PutBucketReplicationRequest putRequest = new
PutBucketReplicationRequest
                {
                    BucketName = sourceBucket,
                    Configuration = replConfig
                };

                PutBucketReplicationResponse putResponse = await
s3Client.PutBucketReplicationAsync(putRequest);

                // Verify configuration by retrieving it.
                await RetrieveReplicationConfigurationAsync(s3Client);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }
    }
}
```

```
        }
        private static async Task RetrieveReplicationConfigurationAsync(IAmazonS3
client)
{
    // Retrieve the configuration.
    GetBucketReplicationRequest getRequest = new GetBucketReplicationRequest
    {
        BucketName = sourceBucket
    };
    GetBucketReplicationResponse getResponse = await
client.GetBucketReplicationAsync(getRequest);
    // Print.
    Console.WriteLine("Printing replication configuration information...");
    Console.WriteLine("Role ARN: {0}", getResponse.Configuration.Role);
    foreach (var rule in getResponse.Configuration.Rules)
    {
        Console.WriteLine("ID: {0}", rule.Id);
        Console.WriteLine("Prefix: {0}", rule.Prefix);
        Console.WriteLine("Status: {0}", rule.Status);
    }
}
}
```

Configurar a replicação quando os buckets de origem e destino pertencerem a contas diferentes

A configuração da replicação quando os buckets de *origem* e *destino* pertencem a Contas da AWS diferentes é semelhante à configuração de uma replicação quando os dois buckets pertencem à 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 adicionando uma política do bucket.

Para obter mais informações sobre como configurar a replicação usando criptografia no lado do servidor com o AWS Key Management Service em cenários entre contas, consulte [Conceder permissões adicionais para cenários entre contas \(p. 872\)](#).

Para configurar a replicação quando os buckets de origem e destino pertencem a Contas da AWS diferentes

1. Neste exemplo, você cria buckets de *origem* e *destino* em duas Contas da AWS diferentes. 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 a definição de perfis da credencial, consulte [Perfis nomeados](#) no Guia do usuário do AWS Command Line Interface.
2. Siga as instruções passo a passo em [Configuração de buckets na mesma conta \(p. 823\)](#) com as seguintes alterações:
 - Para todos os comandos da AWS CLI relacionados a atividades do bucket de *origem* (para criar o bucket de *origem*, habilitar o 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, 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": "2012-10-17",
    "Id": "",
```

```
"Statement": [
    {
        "Sid":"Set permissions for objects",
        "Effect":"Allow",
        "Principal":{
            "AWS":"arn:aws:iam::source-bucket-acct-ID:role/service-role/source-acct-
IAM-role"
        },
        "Action":["s3:ReplicateObject", "s3:ReplicateDelete"],
        "Resource":"arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    },
    {
        "Sid":"Set permissions on bucket",
        "Effect":"Allow",
        "Principal":{
            "AWS":"arn:aws:iam::source-bucket-acct-ID:role/service-role/source-acct-
IAM-role"
        },
        "Action":["s3>List*", "s3:GetBucketVersioning", "s3:PutBucketVersioning"],
        "Resource":"arn:aws:s3:::DOC-EXAMPLE-BUCKET"
    }
]
```

Escolha o bucket e adicione a política do bucket. Para obter instruções, consulte [Adicionar uma política de bucket usando o console do Amazon S3 \(p. 516\)](#).

Na replicação, o proprietário do objeto de origem possui a réplica por padrão. Quando os buckets de origem e destino pertencerem a Contas da AWS diferentes, você poderá adicionar configurações opcionais para alterar a propriedade da réplica para a Conta da AWS proprietária dos buckets de destino. Isso inclui conceder a permissão `ObjectOwnerOverrideToBucketOwner`. Para obter mais informações, consulte [Alterar o proprietário da réplica \(p. 864\)](#).

Alterar o proprietário da réplica quando os buckets de origem e de destino forem de propriedade de contas diferentes

Quando os buckets de `origem` e de `destino` em uma configuração de replicação pertencem a Contas da AWS diferentes, você pode instruir o Amazon S3 a alterar o proprietário 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 [Alterar o proprietário da réplica \(p. 864\)](#).

Note

Quando você usa a replicação do S3 e os buckets de origem e destino pertencem a Contas da AWS diferentes, o proprietário do bucket pode desabilitar ACLs (com a configuração aplicada pelo proprietário do bucket em Object Ownership) para alterar a propriedade da réplica para a Conta da AWS que possui o bucket de destino. Essa configuração imita o comportamento de substituição do proprietário existente sem a necessidade da permissão `s3:ObjectOwnerOverrideToBucketOwner`. Com isso, todos os objetos que são replicados para o bucket de destino com a configuração aplicada pelo proprietário do bucket pertencem ao proprietário do bucket de destino. Para obter informações sobre o Object Ownership, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Para obter mais informações sobre como configurar a replicação usando criptografia do lado do servidor com o AWS Key Management Service em cenários entre contas, consulte [Conceder permissões adicionais para cenários entre contas \(p. 872\)](#).

Uso do console do S3

Para obter instruções detalhadas, consulte [Configuração da replicação para buckets de origem e de destino pertencentes à mesma conta \(p. 823\)](#). Este tópico traz instruções para definir a configuração da replicação quando os buckets pertencerem às mesmas Contas da AWS ou a contas diferentes.

Usar a 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 para replicar objetos e adicione a configuração da replicação ao bucket de origem. Na configuração da replicação, você instrui 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 pertencerem a Contas da AWS diferentes (AWS CLI)

- Neste exemplo, você cria os buckets de *origem* e *destino* em duas Contas da AWS diferentes. Configure a AWS CLI com dois perfis nomeados. Este exemplo usa os perfis nomeados `acctA` e `acctB`, respectivamente. Para obter mais informações sobre a definição de 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, poderá executar todas as tarefas. Para obter mais informações, consulte [Conceder permissões ao usuário para passar uma função a um produto 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 uma função do 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 passar uma função a um produto da AWS](#) no Guia do usuário do IAM.

- Crie o bucket de *origem* e habilite o versionamento. Este exemplo cria 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. Este exemplo cria 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. É necessário adicionar permissões à política de bucket de *destino* para permitir a alteração da propriedade da réplica.

- a. Salve a política a seguir no *destination-bucket-policy.json*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "destination_bucket_policy_sid",
            "Principal": {
                "AWS": "source-bucket-owner-account-id"
            },
            "Action": [
                "s3:ReplicateObject",
                "s3:ReplicateDelete",
                "s3:ObjectOwnerOverrideToBucketOwner",
                "s3:ReplicateTags",
                "s3:GetObjectVersionTagging"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::destination/*"
            ]
        }
    ]
}
```

- b. Coloque a política acima no bucket de *destino*:

```
aws s3api put-bucket-policy --region $ {destination_region} --
bucket $ {destination} --policy file://destination_bucket_policy.json
```

5. Crie uma função do IAM. Você especifica essa função na configuração da replicação que adiciona ao bucket de *origem* posteriormente. O Amazon S3 assume essa função para replicar objetos em seu nome. A função do IAM é criada em duas etapas:

- Crie 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 chamado *s3-role-trust-policy.json* no diretório atual do 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 comando da AWS CLI a seguir para criar uma função.

```
$ aws iam create-role \
--role-name replicationRole \
--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-perm-pol-changeowner.json` no diretório atual do computador local. Essa política concede permissões para várias ações de bucket e objetos do Amazon S3. Nas etapas a seguir, crie uma função do IAM e anexe a 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": [
                "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 replicationRole \
--policy-document file://s3-role-perm-pol-changeowner.json \
```

```
--policy-name replicationRolechangeownerPolicy \
--profile accta
```

6. Adicione uma 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 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": {
            },
            "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.json \
--bucket source \
--profile accta
```

7. Verifique a propriedade da réplica no console do Amazon S3.

- a. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
b. Adicione objetos ao 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`.

Uso da SDKs AWS

Para obter um exemplo de código para adicionar a configuração da replicação, consulte [Uso da SDKs AWS \(p. 830\)](#). É necessário modificar a configuração de replicação de acordo. Para obter informações conceituais, consulte [Alterar o proprietário da réplica \(p. 864\)](#).

Replicar objetos criptografados

Por padrão, o Amazon S3 não replica objetos armazenados em repouso usando criptografia do lado do servidor com chaves do KMS. Para replicar objetos criptografados, modifique a configuração da replicação do bucket para instruir o Amazon S3 a replicar esses objetos. Este exemplo explica como usar o console do Amazon S3 e a AWS Command Line Interface (AWS CLI) para alterar a configuração de replicação do bucket de maneira que a replicação de objetos criptografados seja habilitada. Para obter mais informações, consulte [Replicação de objetos criados com criptografia no lado do servidor \(SSE\) usando chaves do KMS \(p. 867\)](#).

Note

Você pode usar uma AWS KMS keys de várias regiões no Amazon S3. No entanto, o Amazon S3 trata no momento as chaves de várias regiões como se fossem chaves de região única e não usa os recursos de várias regiões da chave. Para obter mais informações, consulte [Usar chaves de várias regiões](#) no Guia do desenvolvedor do AWS Key Management Service.

Uso do console do S3

Para obter instruções detalhadas, consulte [Configuração da replicação para buckets de origem e de destino pertencentes à mesma conta \(p. 823\)](#). Este tópico traz instruções para definir a configuração da replicação quando os buckets pertencerem às mesmas Contas da AWS ou a contas diferentes.

Usar a 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 para 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. As permissões da função do IAM incluem as permissões necessárias para replicar os objetos criptografados. Você também testa a configuração.

Para replicar objetos criptografados no lado do servidor (AWS CLI)

1. Neste exemplo, criamos tanto os buckets de *origem* quanto os 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 a definição de 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
```

3. 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 configurar a replicação quando os buckets de *origem* e de *destino* estiverem na mesma Conta da AWS, use o mesmo perfil. Neste exemplo, usamos acctA. Para testar a configuração da replicação quando os buckets pertencerem a Contas da AWS distintas, especifique diferentes perfis para cada um.

```
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
```

4. Crie uma função do IAM. Você especifica essa função na configuração da replicação que adiciona ao bucket de **origem** posteriormente. 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-kmsobj.json** no diretório atual do seu computador local. Essa política concede ao principal do serviço Amazon S3 as principais permissões para assumir a função para 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 replicationRolekmsobj \  
--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 objetos 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 computador local. Crie uma função do IAM e depois anexe a política à ela.

Important

Na política de permissões, especifique os IDs de chave do AWS KMS que serão usados para a criptografia dos buckets de origem e de destination. Você deve criar duas chaves separadas do KMS para os buckets source e destination. As

AWS KMS keys nunca são compartilhadas fora da Região da AWS em que foram criadas.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "s3>ListBucket",  
                "s3:GetReplicationConfiguration",  
                "s3:GetObjectVersionForReplication",  
                "s3:GetObjectVersionAcl",  
                "s3:GetObjectVersionTagging"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:s3:::source",  
                "arn:aws:s3:::source/*"  
            ]  
        },  
        {  
            "Action": [  
                "s3:ReplicateObject",  
                "s3:ReplicateDelete",  
                "s3:ReplicateTags"  
            ],  
            "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(in ARN format) 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(in ARN format) used to encrypt source objects."  
            ]  
        },  
        {  
            "Action": [  
                "kms:Encrypt"  
            ],  
            "Effect": "Allow",  
            "Condition": {  
                "StringLike": {  
                    "kms:ViaService": "s3.us-east-1.amazonaws.com",  
                    "kms:EncryptionContext:aws:s3:arn": [  
                        "arn:aws:s3:::destination/*"  
                    ]  
                }  
            },  
            "Resource": [  
                "AWS KMS key IDs(in ARN format) used to decrypt replicated objects."  
            ]  
        }  
    ]  
}
```

```
        "kms:ViaService": "s3.us-west-2.amazonaws.com",
        "kms:EncryptionContext:aws:s3:arn": [
            "arn:aws:s3:::destination/*"
        ]
    }
},
"Resource": [
    "AWS KMS key IDs(in ARN format) to use for encrypting object
replicas"
]
}
}
```

- ii. Crie uma política e anexe-a à função.

```
$ aws iam put-role-policy \
--role-name replicationRolekmsobj \
--policy-document file://s3-role-permissions-policykmsobj.json \
--policy-name replicationRolechangeownerPolicy \
--profile accta
```

5. Adicione a seguinte configuração de replicação ao bucket de *origem*. Isso instrui o Amazon S3 a replicar objetos com o prefixo *Tax/* no 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 passar uma função a um produto da AWS](#) no Guia do usuário do IAM.

```
<ReplicationConfiguration>
<Role>IAM-Role-ARN</Role>
<Rule>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
        <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>Tax</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <SourceSelectionCriteria>
        <SseKmsEncryptedObjects>
            <Status>Enabled</Status>
        </SseKmsEncryptedObjects>
    </SourceSelectionCriteria>
    <Destination>
        <Bucket>arn:aws:s3:::dest-bucket-name</Bucket>
        <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(in ARN format) to use for encrypting object replicas"  
                }  
            },  
            "SourceSelectionCriteria": {  
                "SseKmsEncryptedObjects": {  
                    "Status": "Enabled"  
                }  
            }  
        }  
    ]  
}
```

- b. Edite o JSON para fornecer valores para o bucket de **destino**, **ARN de ID do KMS** e **IAM-role-ARN**. Salve as alterações.
- c. 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://replication.json \  
--bucket source \  
--profile acctA
```

6. Teste a configuração para verificar se os objetos criptografados estão replicados. No console do Amazon S3:
 - a. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
 - b. No bucket de **origem**, crie uma pasta chamada **Tax**.
 - c. 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 dos objetos e se elas são criptografadas usando a chave do KMS especificada na configuração.

Uso da SDKs AWS

Para obter um exemplo de código para adicionar a configuração da replicação, consulte [Uso da SDKs AWS \(p. 830\)](#). É necessário modificar a configuração de replicação de acordo.

Para obter informações conceituais, consulte [Replicação de objetos criados com criptografia no lado do servidor \(SSE\) usando chaves do KMS \(p. 867\)](#).

Replicação de objetos com o Controle de Tempo de Replicação do S3 (S3 RTC)

O Controle do tempo de replicação do S3 (S3 RTC) ajuda a atender aos requisitos empresariais ou de conformidade relacionados à replicação de dados, além de fornecer visibilidade dos tempos de replicação do Amazon S3. O S3 RTC replica a maioria dos objetos obtidos por upload para o Amazon S3 em segundos, e 99,99 por cento desses objetos em 15 minutos.

Com o S3 RTC, você pode monitorar o número total e o tamanho dos objetos com replicação pendente e o tempo máximo de replicação para a região de destino. As métricas de replicação estão disponíveis por meio do [AWS Management Console](#) e do [Guia do usuário do Amazon CloudWatch](#). Para obter mais informações, consulte [the section called “Métricas de replicação do Amazon S3 no CloudWatch” \(p. 1072\)](#).

Uso do console do S3

Para obter instruções detalhadas, consulte [Configuração da replicação para buckets de origem e de destino pertencentes à mesma conta \(p. 823\)](#). Este tópico traz instruções para habilitar o S3 RTC na configuração da replicação quando os buckets pertencerem às mesmas Contas da AWS ou a contas diferentes.

Usar a AWS CLI

Para usar a AWS CLI para replicar objetos com o S3 RTC habilitado, crie buckets, habilite o versionamento nos buckets, crie uma função do IAM que conceda permissão ao Amazon S3 para replicar objetos e adicione a configuração da replicação ao bucket de origem. A configuração da replicação precisa ter o Controle do tempo de replicação do S3 (S3 RTC) habilitado.

Para replicar com o S3 RTC habilitado (AWS CLI)

- O exemplo a seguir define `ReplicationTime` e `Metric`, e adiciona configuração de replicação ao bucket de origem.

```
{  
    "Rules": [  
        {  
            "Status": "Enabled",  
            "Filter": {  
                "Prefix": "Tax"  
            },  
            "DeleteMarkerReplication": {  
                "Status": "Disabled"  
            },  
            "Destination": {  
                "Bucket": "arn:aws:s3:::destination",  
                "Metrics": {  
                    "Status": "Enabled",  
                    "EventThreshold": {  
                        "Minutes": 15  
                    }  
                },  
                "ReplicationTime": {  
                    "Status": "Enabled",  
                    "Time": {  
                        "Minutes": 15  
                    }  
                }  
            },  
            "Priority": 1  
        }  
    ],  
    "Role": "IAM-Role-ARN"  
}
```

}

Important

Metrics:EventThreshold:Minutes e ReplicationTime:Time:Minutes podem ter apenas 15, que é um valor válido.

Usar o AWS SDK para Java

O seguinte exemplo de Java adiciona a configuração da replicação com o Controle do tempo de replicação do S3 (S3 RTC):

```
import software.amazon.awssdk.auth.credentials.AwsBasicCredentials;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.model.DeleteMarkerReplication;
import software.amazon.awssdk.services.s3.model.Destination;
import software.amazon.awssdk.services.s3.model.Metrics;
import software.amazon.awssdk.services.s3.model.MetricsStatus;
import software.amazon.awssdk.services.s3.model.PutBucketReplicationRequest;
import software.amazon.awssdk.services.s3.model.ReplicationConfiguration;
import software.amazon.awssdk.services.s3.model.ReplicationRule;
import software.amazon.awssdk.services.s3.model.ReplicationRuleFilter;
import software.amazon.awssdk.services.s3.model.ReplicationTime;
import software.amazon.awssdk.services.s3.model.ReplicationTimeStatus;
import software.amazon.awssdk.services.s3.model.ReplicationTimeValue;

public class Main {

    public static void main(String[] args) {
        S3Client s3 = S3Client.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(() -> AwsBasicCredentials.create(
                "AWS_ACCESS_KEY_ID",
                "AWS_SECRET_ACCESS_KEY"))
        )
        .build();

        ReplicationConfiguration replicationConfig = ReplicationConfiguration
            .builder()
            .rules(
                ReplicationRule
                    .builder()
                    .status("Enabled")
                    .priority(1)
                    .deleteMarkerReplication(
                        DeleteMarkerReplication
                            .builder()
                            .status("Disabled")
                            .build())
            )
            .destination(
                Destination
                    .builder()
                    .bucket("destination_bucket_arn")
                    .replicationTime(
                        ReplicationTime.builder().time(
                            ReplicationTimeValue.builder().minutes(15).build())
                        .status(
                            ReplicationTimeStatus.ENABLED
                            ).build())
            )
            .metrics(
                Metrics.builder().eventThreshold(
```

```
        ReplicationTimeValue.builder().minutes(15).build()
    ).status(
        MetricsStatus.ENABLED
    ).build()
)
.build()
)
.filter(
    ReplicationRuleFilter
    .builder()
    .prefix("testtest")
    .build()
)
.build())
.role("role_arn")
.build();

// Put replication configuration
PutBucketReplicationRequest putBucketReplicationRequest = PutBucketReplicationRequest
    .builder()
    .bucket("source_bucket")
    .replicationConfiguration(replicationConfig)
    .build();

s3.putBucketReplication(putBucketReplicationRequest);
}
}
```

Para obter mais informações, consulte [Atendimento aos requisitos de conformidade usando o Controle do tempo de replicação do S3 \(S3 RTC\) \(p. 860\)](#).

Gerenciamento de regras de replicação usando o console do Amazon S3

A replicação é a cópia assíncrona automática de objetos em buckets na mesma região ou em diferentes Regiões da AWS. Ela replica os objetos recém-criados e as atualizações de objeto de um bucket de origem para um bucket de destino especificado.

Você usa o console do Amazon S3 para adicionar regras de replicação ao bucket de origem. As regras de replicação definem os objetos do bucket de origem que devem ser replicados e o bucket de destino ou buckets nos quais os objetos replicados estão armazenados. Para obter mais informações sobre a replicação, consulte [Replicação de objetos \(p. 801\)](#).

Você pode gerenciar as regras de replicação na página Replication (Replicação). Você pode adicionar, ver, habilitar, desabilitar, excluir e alterar a prioridade das regras de replicação. Para obter informações sobre como adicionar regras de replicação a um bucket, consulte [Uso do console do S3 \(p. 823\)](#).

Como gerenciar as regras de replicação para um bucket do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket desejado.
3. Escolha Management (Gerenciamento) e role para baixo até Replication rules (Regras de replicação).
4. Você pode alterar as regras de replicação das seguintes formas:
 - Para habilitar ou desabilitar uma regra de replicação, selecione a regra, escolha Actions (Ações), e, na lista suspensa, escolha Enable rule (Habilitar regra) ou Disable rule (Desabilitar regra). Você também pode desabilitar, habilitar ou excluir todas as regras do bucket na lista suspensa Actions (Ações).

- Para alterar as prioridades de regra, selecione a regra e escolha Edit (Editar), o que faz com que o Assistente de replicação comece a ajudá-lo a fazer a alteração. Para obter informações sobre como usar o assistente, consulte [Uso do console do S3 \(p. 823\)](#).

Defina prioridades para a regra, de maneira a evitar conflitos causados pelos objetos incluídos no escopo de mais de uma regra. No caso de sobreposição de regras, o Amazon S3 usa a prioridade da regra para determinar qual regra aplicar. Quanto maior o número, maior a prioridade. Para obter mais informações sobre prioridade de regra, consulte [Configuração de replicação \(p. 808\)](#).

Replicar objetos existentes com o S3 Batch Replication

O recurso S3 Batch Replication oferece uma maneira de replicar objetos que existiam antes que uma configuração de replicação estivesse em vigor, objetos que foram replicados anteriormente e objetos cuja replicação falhou. Isso é feito por meio do uso de um trabalho de operações em lote. Isso é diferente da replicação em tempo real, que replica de modo contínuo e automático, os objetos novos entre buckets do Amazon S3. Para começar a usar a replicação em lote, você pode:

- Iniciar a replicação em lote para uma nova regra ou destino de replicação : você pode criar um único trabalho de replicação em lote ao criar a primeira regra em uma nova configuração de replicação ou ao adicionar um novo destino a uma configuração existente por meio do AWS Management Console.
- Iniciar a replicação em lote para uma configuração de replicação existente: você pode criar um novo trabalho de replicação em lote usando S3 Batch Operations por meio dos AWS SDKs, da AWS Command Line Interface (AWS CLI) ou o console do Simple Storage Service (Amazon S3).

Quando o trabalho de replicação em lote termina, você recebe um relatório de conclusão. Para obter mais informações sobre como usar o relatório para examinar o trabalho, consulte [Rastreamento de relatórios de status e conclusão \(p. 982\)](#).

Considerações sobre o recurso S3 Batch Replication

- O bucket de origem deve ter uma configuração de replicação existente. Para habilitar a replicação, consulte [Configuração da replicação \(p. 807\)](#) e [Demonstrações: exemplos para configurar a replicação \(p. 822\)](#).
- Se o ciclo de vida do S3 estiver configurado para seu bucket, recomendamos desabilitar as regras de ciclo de vida enquanto o trabalho de replicação em lote estiver ativo. Isso garantirá a paridade entre os buckets de origem e de destino. Caso contrário, esses buckets podem divergir e o bucket de destino não será uma réplica exata do bucket de origem. Considere o seguinte:
 - Seu bucket de origem tem várias versões em um objeto e um marcador de exclusão.
 - Seus buckets de origem e destino têm uma política de ciclo de vida para remover marcadores de exclusão expirados.

A replicação em lote pode replicar o marcador de exclusão para o bucket de destino antes de replicar as versões do objeto. Isso pode fazer com que o marcador de exclusão seja marcado como expirado e seja removido do bucket de destino antes que os objetos sejam copiados.

- A função do AWS Identity and Access Management (IAM) que você especifica para executar o trabalho de operações em lote deve ter permissões para executar a operação de replicação de lote subjacente. Para obter mais informações sobre como criar funções do IAM, consulte [Configurar políticas do IAM para replicação em lote \(p. 851\)](#).
- A replicação em lote requer um manifesto que pode ser gerado pelo Simple Storage Service (Amazon S3). O manifesto gerado deve ser armazenado na mesma Região da AWS que o bucket de origem. Se escolher não gerar o manifesto, você poderá fornecer um relatório do Amazon S3 Inventory ou um arquivo CSV que contenha os objetos que deseja replicar.

- A replicação em lote não é compatível com a nova replicação de objetos que foram excluídos com o ID de versão do objeto do bucket de destino. Para replicar esses objetos novamente, você pode copiar os objetos de origem no local com um trabalho de cópia em lote. Copiar esses objetos no local criará versões do objeto no bucket de origem e iniciará a replicação automaticamente para o destino.

Para obter mais informações sobre cópia em lotes, consulte [Exemplos que usam o Batch Operations para copiar objetos \(p. 954\)](#).

Especificando um manifesto para um trabalho de replicação em lote

Um manifesto é um objeto do Amazon S3 que contém as chaves de objeto em que você deseja que o Simple Storage Service (Amazon S3) atue. Se desejar criar um trabalho de replicação em lote, você deverá fornecer um manifesto gerado pelo usuário ou fazer com que o Simple Storage Service (Amazon S3) gere um manifesto com base na configuração de replicação.

Se fornecer um manifesto gerado pelo usuário, ele deverá estar na forma de um relatório de inventário do Amazon S3 Inventory ou um arquivo CSV. Se os objetos no manifesto estiverem em um bucket versionado, você deverá especificar os IDs de versão dos objetos. Somente o objeto com o ID de versão especificado no manifesto será replicado. Para saber mais sobre como especificar um manifesto, consulte [Especificar um manifesto \(p. 948\)](#).

Se você preferir que o Simple Storage Service (Amazon S3) gere um arquivo de manifesto em seu nome, os objetos listados usarão o mesmo bucket de origem, prefixo e etiquetas da configuração de replicação. Com um manifesto gerado, o Simple Storage Service (Amazon S3) replicará todas as versões elegíveis dos objetos.

Note

Se escolher que o manifesto seja gerado, ele deverá ser armazenado na mesma Região da AWS que o bucket de origem.

Filtros para um trabalho de replicação em lote

Ao criar o trabalho de replicação em lote, você pode especificar filtros adicionais, como data de criação e status de replicação do objeto para reduzir o escopo do trabalho.

É possível filtrar os objetos a serem replicados com base no valor de `ObjectReplicationStatuses`, fornecendo um ou mais dos seguintes valores:

- "`NONE`": indica que o Simple Storage Service (Amazon S3) nunca tentou replicar o objeto antes.
- "`FAILED`": indica que o Simple Storage Service (Amazon S3) tentou replicar o objeto, mas não teve êxito.
- "`COMPLETED`": indica que o Simple Storage Service (Amazon S3) já replicou com êxito o objeto anteriormente.
- "`REPLICA`": indica que esse é um objeto que foi replicado pelo Simple Storage Service (Amazon S3) de outra origem.

Para obter mais informações sobre a replicação, consulte [Obtenção de informações sobre o status da replicação \(p. 873\)](#).

Se você não filtrar com base no status da replicação, as operações em lote tentarão replicar tudo o que for elegível. Dependendo do objetivo, você pode definir `ObjectReplicationStatuses` com um dos seguintes valores:

- Se desejar replicar somente objetos existentes que nunca foram replicados, inclua apenas "NONE".
- Se desejar repetir a replicação somente dos objetos cuja replicação falhou anteriormente, inclua apenas "FAILED".
- Se desejar replicar os objetos existentes e tentar novamente replicar os objetos cuja replicação falhou anteriormente, inclua "NONE" e "FAILED".
- Se desejar preencher um bucket de destino com os objetos que foram replicados para outro destino, inclua "COMPLETED".
- Se você desejar replicar objetos anteriormente replicados, inclua "REPLICA".

Configurar políticas do IAM para replicação em lote

Como o S3 Batch Replication é um tipo de trabalho de operações em lote, você deve criar uma função do AWS Identity and Access Management (IAM) de operações em lote para conceder permissões ao Amazon S3 para executar ações em seu nome. Você também deve anexar uma política do IAM de replicação em lote à função do IAM de operações em lote. O exemplo a seguir cria uma função do IAM que concede permissão às operações em lote para iniciar um trabalho de replicação em lote.

Criar uma política e uma função do IAM

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. Em Access management (Gerenciamento de acesso), escolha Roles (Funções).
3. Selecione Create Role (Criar função).
4. Escolha AWS service (Serviço da AWS) como o tipo de entidade de confiança, Amazon S3 como o serviço e S3 Batch Operations (Operações em lote do S3) como caso de uso.
5. Escolha Next: Permissions (Próximo: permissões).
6. Selecione Create Policy (Criar política).
7. Escolha JSON e insira uma das seguintes políticas com base no manifesto.

Note

Permissões diferentes serão necessárias se você estiver gerando um manifesto ou fornecendo um. Para obter mais informações, consulte [Especificando um manifesto para um trabalho de replicação em lote \(p. 850\)](#).

Política se estiver usando e armazenando um manifesto gerado pelo S3

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "s3:InitiateReplication"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:s3:::*** replication source bucket ***/*"  
            ]  
        },  
        {  
            "Action": [  
                "s3:GetReplicationConfiguration",  
                "s3:PutInventoryConfiguration"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:s3:::*** replication target bucket ***/*"  
            ]  
        }  
    ]  
}
```

```
        "arn:aws:s3:::*** replication source bucket ***"
    ],
},
{
    "Action":[
        "s3:GetObject",
        "s3:GetObjectVersion"
    ],
    "Effect":"Allow",
    "Resource":[
        "arn:aws:s3:::*** manifest bucket ***/*"
    ],
},
{
    "Effect":"Allow",
    "Action":[
        "s3:PutObject"
    ],
    "Resource":[
        "arn:aws:s3:::*** completion report bucket ****/*",
        "arn:aws:s3:::*** manifest bucket ****/*"
    ]
}
]
```

Política se estiver usando um manifesto fornecido pelo usuário

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Action":[
                "s3:InitiateReplication"
            ],
            "Effect":"Allow",
            "Resource":[
                "arn:aws:s3:::*** replication source bucket ***/*"
            ]
        },
        {
            "Action":[
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Effect":"Allow",
            "Resource":[
                "arn:aws:s3:::*** manifest bucket ***/*"
            ]
        },
        {
            "Effect":"Allow",
            "Action":[
                "s3:PutObject"
            ],
            "Resource":[
                "arn:aws:s3:::*** completion report bucket ****/*"
            ]
        }
    ]
}
```

8. Escolha Next: Tags (Próximo: etiquetas).
9. Escolha Next: Review (Próximo: revisar).

10. Escolha um nome para a política e escolha Create policy (Criar política).
11. Anexe essa política à sua função e escolha Next: Tags (Próximo: etiquetas).
12. Escolha Next: Review (Próximo: revisar).
13. Escolha um nome para a função e escolha Create role (Criar função).

Verifique a política de confiança

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. Em Access management (Gerenciamento de acesso), escolha Roles (Funções) e selecione a função que acabou de criar.
3. Na guia Trust relationships (Relações de confiança), escolha Edit trust relationship (Editar relação de confiança).
4. Verifique se essa função está usando a seguinte política de confiança:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "batchoperations.s3.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

Criar um trabalho de replicação em lote para uma primeira regra de replicação ou um novo destino

Ao criar a primeira regra em uma nova configuração de replicação ou ao adicionar um novo destino a uma configuração existente por meio do AWS Management Console, você tem a opção de criar um trabalho de replicação em lote.

Para usar a replicação em lote para uma configuração existente sem adicionar um novo destino, consulte [Criar um trabalho de replicação em lote para regras de replicação existentes \(p. 854\)](#).

Replicação em lote para uma primeira regra de replicação ou um novo destino usando o AWS Management Console

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket que contém os objetos que você deseja replicar.
3. Para criar uma regra de replicação ou editar uma existente, selecione Management (Gerenciamento) e role para baixo até Replication Rules (Regras de replicação):
 - Para criar uma regra de replicação, selecione Create replication rule (Criar regra de replicação).

Note

Para obter exemplos sobre como configurar uma regra de replicação básica, consulte [Demonstrações: exemplos para configurar a replicação \(p. 822\)](#).
 - Para editar uma regra de replicação existente, selecione a regra e Edit rule (Editar regra).

4. Crie sua regra de replicação ou edite o destino da regra de replicação existente e selecione Save (Salvar).

Depois de criar a primeira regra em uma nova configuração de replicação ou editar uma configuração existente para adicionar um novo destino, a caixa de diálogo Replicate existing objects? (Replicar objetos existentes?) é exibida, oferecendo a você a opção de criar um trabalho de replicação em lote.

5. Para criar um trabalho de replicação em lote, selecione Yes, replicate existing objects (Sim, replicar objetos existentes).

Para executar seu trabalho de replicação em lote posteriormente, selecione No, do not replicate existing objects (Não, não replique objetos existentes).

6. Crie o trabalho de replicação em lote do S3. O trabalho do S3 Batch Replication tem várias configurações:

Opção de executar trabalho

Se você quiser que o trabalho de replicação em lote do S3 seja executado imediatamente, selecione Job runs automatically when ready (O trabalho é executado automaticamente quando pronto). Se quiser executar o trabalho mais tarde, selecione Job waits to be run when ready (O trabalho espera para ser executado quando estiver pronto).

Se escolher Job runs automatically when ready (O trabalho é executado automaticamente quando pronto), você não poderá criar e salvar um manifesto de operações em lote. Para salvar o manifesto de operações em lote, escolha Job waits to be run when ready (O trabalho espera para ser executado quando pronto).

Manifesto de operações em lote

O manifesto é uma lista de todos os objetos em que você deseja que a ação especificada seja executada. Você pode escolher salvar o manifesto de operações em lote. Como acontece com os arquivos do S3 Inventory, o manifesto será salvo como um arquivo CSV e armazenado em um bucket. Para saber mais sobre os manifestos de operações em lote, consulte [Especificificar um manifesto \(p. 948\)](#).

Relatórios de conclusão

S3 Batch Operations executam uma tarefa para cada objeto especificado no manifesto. Os relatórios de conclusão oferecem uma maneira fácil de visualizar os resultados das tarefas em um formato consolidado, sem a necessidade de configurações adicionais. Você pode solicitar um relatório de conclusão para todas as tarefas ou somente para as que apresentarem falha. Para saber mais sobre relatórios de conclusão, consulte [Relatórios de conclusão \(p. 987\)](#).

Permissões

Uma das causas mais comuns de falhas de replicação são permissões insuficientes na função do AWS Identity and Access Management (IAM) fornecida. Para obter informações sobre como criar essa função, consulte [Configurar políticas do IAM para replicação em lote \(p. 851\)](#).

7. Escolha Create Batch Operations job (Criar trabalho de operações em lote).

Criar um trabalho de replicação em lote para regras de replicação existentes

Você pode configurar o S3 Batch Replication para uma configuração de replicação existente usando os AWS SDKs, a AWS Command Line Interface (AWS CLI) ou o console do Simple Storage Service (Amazon S3). Para ter uma visão geral da replicação em lote, consulte [Replicar objetos existentes com o S3 Batch Replication \(p. 849\)](#).

Como pré-requisito, você deve criar uma função do AWS Identity and Access Management (IAM) de operações em lote para conceder ao Simple Storage Service (Amazon S3) permissões para executar ações em seu nome, consulte [Configurar políticas do IAM para replicação em lote \(p. 851\)](#).

Quando o trabalho de replicação em lote termina, você recebe um relatório de conclusão. Para obter mais informações sobre como usar o relatório para examinar o trabalho, consulte [Rastreamento de relatórios de status e conclusão \(p. 982\)](#).

Uso do console do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Selecione Batch Operations (Operações em lote) no painel de navegação do console do Amazon S3.
3. Selecione Create job (Criar trabalho).
4. Selecione a Region (região) onde você deseja criar o trabalho.
5. Selecione o Manifest format (Formato do manifesto). Esse exemplo mostrará como criar um manifesto com base em uma configuração de replicação do S3 existente.

Note

O manifesto é uma lista de todos os objetos em que você deseja que a ação especificada seja executada. Para saber mais sobre os manifestos de operações em lote, consulte [Especificando um manifesto \(p. 948\)](#). Se você tiver um manifesto preparado, escolha S3 inventory report (manifest.json) [Relatório de inventário do S3 (manifest.json)] ou CSV. Se os objetos no manifesto estiverem em um bucket com versão, você deverá especificar os IDs de versão dos objetos. Para obter mais informações sobre como criar um manifesto, consulte [Especificando um manifesto \(p. 948\)](#).

6. Para criar um manifesto com base na configuração de replicação, escolha Create manifest using S3 Replication configuration (Criar manifesto usando a configuração do S3 Replication). Depois, escolha o bucket de origem de sua configuração de replicação.
7. (Opcional) Você pode incluir filtros adicionais, como data de criação e status de replicação do objeto. Para obter exemplos de como filtrar por status de replicação, consulte [Especificando um manifesto para um trabalho de replicação em lote \(p. 850\)](#).
8. Para salvar um manifesto, selecione Save Batch Operations manifest (Salvar manifesto de operações em lote).
 - a. Se você optar por gerar e salvar um manifesto, deverá escolher Bucket in this account (Bucket nesta conta) ou Bucket in another AWS Account (Bucket em outra Conta da AWS). Especifique o nome do bucket na caixa de texto.

Note

O manifesto gerado deve ser armazenado na mesma Região da AWS que o bucket de origem.

- b. Selecione o Encryption key type (Tipo de chave de criptografia).
9. (Opcional) Forneça uma Description (Descrição).
 10. Ajuste a Priority (Prioridade) do trabalho, se necessário. Números mais altos indicam maior prioridade. O Simple Storage Service (Amazon S3) tenta executar trabalhos de prioridade mais alta antes dos trabalhos de prioridade mais baixa. Para obter mais informações sobre prioridade de trabalhos, consulte [Atribuir prioridade aos trabalhos \(p. 980\)](#).
 11. (Opcional) Gere um relatório de conclusão. Para gerar o relatório, selecione Generate completion report (Gerar relatório de conclusão).

Se escolher gerar um relatório de conclusão, você deverá escolher se o relatório informará Failed tasks only (Apenas as tarefas que falharam) ou All tasks (Todas as tarefas) e fornecer um bucket de destino para o relatório.

12. Selecione uma função do IAM válida.

Note

Para obter mais informações sobre como criar uma função do IAM, consulte [Configurar políticas do IAM para replicação em lote \(p. 851\)](#).

13. (Opcional) Adicione etiquetas de trabalho à tarefa de replicação em lote.
14. Escolha Next (Próximo).
15. Revise a configuração do trabalho e selecione Create job (Criar trabalho).

Usar o AWS CLI com um manifesto do S3

O exemplo a seguir cria um trabalho do S3 Batch Replication usando um manifesto gerado pelo S3 para a Conta da AWS **111122223333**. Este exemplo tentará replicar os objetos existentes e os objetos cuja replicação falhou anteriormente. Para obter informações sobre como filtrar por status de replicação, consulte [Especificando um manifesto para um trabalho de replicação em lote \(p. 850\)](#).

```
aws s3control create-job --account-id 111122223333 --operation
'{"S3ReplicateObject":{}}' --report '{"Bucket":"arn:aws:s3:::***  

completion report bucket ***", "Prefix": "batch-replication-report",  

"Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}' --manifest-  

generator '{"S3JobManifestGenerator": {"ExpectedBucketOwner": "111122223333",  

"SourceBucket": "arn:aws:s3:::*** replication source bucket ***", "EnableManifestOutput":  

false, "Filter": {"EligibleForReplication": true, "ObjectReplicationStatuses":  

["NONE", "FAILED"]}}}' --priority 1 --role-arn arn:aws:iam::111122223333:role/batch-  

Replication-IAM-policy --no-confirmation-required --region source-bucket-region
```

Note

O trabalho deve ser iniciado na mesma Região da AWS do bucket de origem da replicação. A função do IAM role/**batch-Replication-IAM-policy** foi criada anteriormente. Consulte [Configurar políticas do IAM para replicação em lote \(p. 851\)](#).

Após iniciar com êxito um trabalho de replicação em lote, você receberá o ID do trabalho como resposta. Você pode monitorar o trabalho usando o comando a seguir.

```
aws s3control describe-job --account-id 111122223333 --job-id job-id --region source-  

bucket-region
```

Usar o AWS CLI com um manifesto fornecido pelo usuário

O exemplo a seguir cria um trabalho do S3 Batch Replication usando um manifesto gerado pelo usuário para a Conta da AWS **111122223333**. Se os objetos no manifesto estiverem em um bucket versionado, você deverá especificar os IDs de versão dos objetos. Somente o objeto com o ID da versão especificado no manifesto será replicado. Para obter mais informações sobre como criar um manifesto, consulte [Especificar um manifesto \(p. 948\)](#).

```
aws s3control create-job --account-id 111122223333 --operation
'{"S3ReplicateObject":{}}' --report '{"Bucket":"arn:aws:s3:::***  

completion report bucket ***", "Prefix": "batch-replication-report",  

"Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}'  

--manifest '{"Spec":{"Format":"S3BatchOperations_CS_20180820","Fields":  

["Bucket","Key","VersionId"],"Location":{"ObjectArn": "arn:aws:s3:::*** completion  

report bucket ***/manifest.csv", "ETag": "Manifest Etag"}}}}' --priority 1 --role-arn  

arn:aws:iam::111122223333:role/batch-Replication-IAM-policy --no-confirmation-required --  

region source-bucket-region
```

Note

O trabalho deve ser iniciado na mesma Região da AWS do bucket de origem da replicação. A função do IAM role/[batch-Replication-IAM-policy](#) foi criada anteriormente. Consulte [Configurar políticas do IAM para replicação em lote \(p. 851\)](#).

Após iniciar com êxito um trabalho de replicação em lote, você receberá o ID do trabalho como resposta. Você pode monitorar o trabalho usando o comando a seguir.

```
aws s3control describe-job --account-id 111122223333 --job-id job-id --region source-bucket-region
```

Configurações de replicação adicionais

Esta seção descreve opções adicionais de configuração de replicação disponíveis no Amazon S3. Para obter informações sobre a configuração de replicação principal, consulte [Configuração da replicação \(p. 807\)](#).

Tópicos

- [Monitoramento do progresso com métricas de replicação e notificações de eventos do Amazon S3 \(p. 857\)](#)
- [Atendimento aos requisitos de conformidade usando o Controle do tempo de replicação do S3 \(S3 RTC\) \(p. 860\)](#)
- [Replicação de marcadores de exclusão entre intervalos \(p. 862\)](#)
- [Replicação de alterações de metadados com sincronização de modificação de réplica do Amazon S3 \(p. 863\)](#)
- [Alterar o proprietário da réplica \(p. 864\)](#)
- [Replicação de objetos criados com criptografia no lado do servidor \(SSE\) usando chaves do KMS \(p. 867\)](#)

Monitoramento do progresso com métricas de replicação e notificações de eventos do Amazon S3

As métricas de replicação do S3 fornecem métricas detalhadas para as regras de replicação na configuração de replicação. Com as métricas de replicação, você pode monitorar o andamento minuto a minuto da replicação rastreando bytes pendentes, operações pendentes e latência de replicação. Além disso, você pode configurar Notificações de eventos do Amazon S3 para receber eventos de falha de replicação para ajudar na solução de problemas de configuração.

Quando ativadas, as métricas de replicação do S3 publicam as seguintes métricas no Amazon CloudWatch:

Bytes pendentes de replicação—O número total de bytes de objetos com replicação pendente para uma determinada regra de replicação.

Latência de replicação— o número máximo de segundos pelo qual os buckets de destino da replicação estão atrás do bucket de origem para uma determinada regra de replicação.

Operações pendentes de replicação— o número de operações pendentes de replicação para uma determinada regra de replicação. As operações incluem objetos, marcadores de exclusão, tags, ACLs e operações de Bloqueio de objeto.

Note

As métricas de replicação do S3 são cobradas usando a mesma taxa das métricas personalizadas do Amazon CloudWatch. Para obter mais informações, consulte [Definição de preço do Amazon CloudWatch](#).

As métricas de replicação do S3 são ativadas automaticamente quando você habilita o Controle do tempo de replicação do S3 (S3 RTC). O S3 RTC inclui outros recursos, como um contrato de nível de serviço (SLA) e notificações para limites perdidos. Para obter mais informações, consulte [Atendimento aos requisitos de conformidade usando o Controle do tempo de replicação do S3 \(S3 RTC\)](#) (p. 860).

Tópicos

- [Ativação de métricas de replicação do S3](#) (p. 858)
- [Recebimento de eventos de falha de replicação com notificações de eventos do Amazon S3](#) (p. 859)
- [Exibição de métricas de replicação usando o console do Amazon S3](#) (p. 859)

Ativação de métricas de replicação do S3

Você pode começar a usar métricas de replicação do S3 com uma regra de replicação nova ou existente. Você pode optar por aplicar a regra de replicação a um bucket do S3 inteiro ou a objetos do Amazon S3 com um prefixo ou tag específica.

Este tópico traz instruções para habilitar as métricas de replicação do S3 em sua configuração da replicação quando os buckets pertencerem às mesmas Contas da AWS ou a contas diferentes.

Para habilitar métricas de replicação usando a AWS Command Line Interface (AWS CLI), você deve adicionar uma configuração de replicação ao bucket de origem com Metrics habilitado. Neste exemplo de configuração, os objetos sob o prefixo **Tax** (Imposto) são replicados para o bucket de destino **DOC-EXAMPLE-BUCKET**, e métricas são geradas para esses objetos.

```
{  
    "Rules": [  
        {  
            "Status": "Enabled",  
            "Filter": {  
                "Prefix": "Tax"  
            },  
            "Destination": {  
                "Bucket": "arn:aws:s3:::DOC-EXAMPLE-BUCKET",  
                "Metrics": {  
                    "Status": "Enabled"  
                }  
            },  
            "Priority": 1  
        }  
    ],  
    "Role": "IAM-Role-ARN"  
}
```

Para obter instruções completas sobre como criar regras de replicação, consulte [Configuração da replicação para buckets de origem e de destino pertencentes à mesma conta](#) (p. 823).

Para obter mais informações sobre como exibir métricas de replicação no console do S3, consulte [Exibição de métricas de replicação usando o console do Amazon S3](#) (p. 859).

Recebimento de eventos de falha de replicação com notificações de eventos do Amazon S3

As notificações de eventos do Amazon S3 podem notificá-lo na instância rara quando os objetos não forem replicados para a região de destino. Os eventos do Amazon S3 estão disponíveis no Amazon Simple Queue Service (Amazon SQS), no Amazon Simple Notification Service (Amazon SNS) ou no AWS Lambda. Para obter mais informações, consulte [Configurar notificações de eventos do Amazon S3](#).

Exibição de métricas de replicação usando o console do Amazon S3

Existem três tipos de métricas do Amazon CloudWatch para o Amazon S3: métricas de armazenamento, métricas de solicitação e métricas de replicação. As métricas de replicação são ativadas automaticamente quando você habilita a replicação com o S3 Replication Time Control (S3 RTC) usando o AWS Management Console ou a API do Amazon S3. As métricas de replicação estão disponíveis 15 minutos após a habilitação de uma regra de replicação com o Controle do tempo de replicação do S3 (S3 RTC).

As métricas de replicação controlam os IDs de regra da configuração de replicação. Um ID de regra de replicação pode ser específico para um prefixo, uma tag ou uma combinação de ambos. Para obter mais informações sobre o Controle do tempo de replicação do S3 (S3 RTC), consulte [Atendimento aos requisitos de conformidade usando o Controle do tempo de replicação do S3 \(S3 RTC\)](#) (p. 860).

Para obter mais informações sobre métricas do CloudWatch para o Amazon S3, consulte [Monitoramento de métricas com o Amazon CloudWatch](#) (p. 1067).

Pré-requisitos

Habilite uma regra de replicação que tenha o S3 RTC.

Como exibir métricas de replicação

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket que contém os objetos para os quais você deseja métricas de replicação.
3. Escolha a guia Metrics.
4. Em Replication metrics (Métricas de replicação), escolha Replication rules (Regras de replicação).
5. Escolha Display charts (Exibir gráficos).

O Amazon S3 exibe Replication Latency (in seconds) (Latência de replicação (em segundos)), Operations pending replication (Operações pendentes de replicação) em gráficos.

6. Para exibir todas as métricas de replicação, incluindo Bytes pending replication (Bytes pendentes de replicação), Replication Latency (in seconds) (Latência de replicação (em segundos)), e Operations pending replication (Operações pendentes replicação) juntos em uma página separada, escolha View 1 more chart (Exibir mais 1 gráfico).

Depois, você pode exibir as métricas de replicação Replication Latency (in seconds) (Latência de replicação (em segundos)), Operations pending replication (Operações pendentes de replicação) e Bytes pending replication (Replicação de bytes pendentes) para as regras selecionadas. O Amazon CloudWatch começa a relatar métricas de replicação 15 minutos após você habilitar o S3 RTC na respectiva regra de replicação. Você pode visualizar métricas de replicação no console do Amazon S3 ou do CloudWatch. Para obter mais informações, consulte [Métricas de replicação com o S3 RTC](#) (p. 860).

Atendimento aos requisitos de conformidade usando o Controle do tempo de replicação do S3 (S3 RTC)

O Controle do tempo de replicação do S3 (S3 RTC) ajuda a atender aos requisitos empresariais ou de conformidade relacionados à replicação de dados, além de fornecer visibilidade dos tempos de replicação do Amazon S3. O S3 RTC replica a maioria dos objetos obtidos por upload para o Amazon S3 em segundos, e 99,99 por cento desses objetos em 15 minutos.

O S3 RTC por padrão inclui métricas de replicação S3 e notificações de eventos S3, com as quais você pode monitorar o número total de operações da API do S3 que estão com replicação pendente, o tamanho total dos objetos com replicação pendente e o tempo máximo de replicação. As métricas de replicação podem ser habilitadas independentemente do S3 RTC; consulte [Monitoramento do progresso com métricas de replicação](#). Além disso, o S3 RTC fornece eventos `OperationMissedThreshold` e `OperationReplicatedAfterThreshold` que notificam o proprietário do bucket se a replicação do objeto exceder ou replicar após o limite de 15 minutos.

Com o S3 RTC, os eventos do Amazon S3 podem notificar você nas raras vezes em que objetos não replicam em até 15 minutos e quando esses objetos são replicados com êxito para a região de destino. Os eventos do Amazon S3 estão disponíveis por meio do Amazon SQS, Amazon SNS ou AWS Lambda. Para obter mais informações, consulte [the section called “Notificações de eventos do Amazon S3” \(p. 1083\)](#).

Tópicos

- [Habilitar o Controle do tempo de replicação do S3 \(p. 860\)](#)
- [Métricas de replicação com o S3 RTC \(p. 860\)](#)
- [Usar notificações de eventos do Amazon S3 para rastrear objetos de replicação \(p. 861\)](#)
- [Melhores práticas e diretrizes do S3 RTC \(p. 861\)](#)

Habilitar o Controle do tempo de replicação do S3

Você pode começar a usar o Controle do tempo de replicação do S3 (S3 RTC) com uma regra de replicação nova ou existente. Você pode optar por aplicar a regra de replicação a um bucket do S3 inteiro ou a objetos do Amazon S3 com um prefixo ou tag específica. Quando você habilita o S3 RTC, as métricas de replicação também são habilitadas na regra de replicação.

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 replicará o marcador de exclusão, por padrão. No entanto, você pode adicionar replicação do marcador de exclusão a regras não baseadas em tags.

Note

As métricas de replicação são cobradas na mesma taxa que as métricas personalizadas do Amazon CloudWatch. Para obter mais informações, consulte [Definição de preço do Amazon CloudWatch](#).

Para obter mais informações sobre como criar uma regra com o S3 RTC, consulte [Replicação de objetos com o Controle de Tempo de Replicação do S3 \(S3 RTC\) \(p. 846\)](#).

Métricas de replicação com o S3 RTC

As regras de replicação com o Controle do tempo de replicação do S3 (S3 RTC) habilitado publica métricas de replicação. Com métricas de replicação, é possível monitorar o número total de operações de API do S3 que estão pendentes de replicação, o tamanho total de objetos pendentes de replicação e o tempo máximo de replicação para a região de destino. Você pode monitorar cada conjunto de dados replicado separadamente.

As métricas de replicação ficam disponíveis 15 minutos após a ativação do S3 RTC. As métricas de replicação estão disponíveis por meio do [console do Amazon S3](#), da [API do Amazon S3](#), dos AWS SDKs,

da [AWS Command Line Interface \(AWS CLI\)](#) e do [Amazon CloudWatch](#). Para obter mais informações, consulte [Monitoramento de métricas com o Amazon CloudWatch \(p. 1067\)](#).

Para obter mais informações sobre como encontrar métricas de replicação por meio do console do Amazon S3, consulte [Exibição de métricas de replicação usando o console do Amazon S3 \(p. 859\)](#).

Usar notificações de eventos do Amazon S3 para rastrear objetos de replicação

Você pode controlar o tempo de replicação de objetos que não foram replicados em 15 minutos monitorando notificações de eventos específicas que o Controle do tempo de replicação do S3 (S3 RTC) publica. Esses eventos são publicados quando um objeto qualificado para replicação usando o S3 RTC não foi replicado em 15 minutos e quando esse objeto é replicado para a região de destino.

Os eventos de replicação estão disponíveis dentro de 15 minutos após a ativação do S3 RTC. Os eventos do Amazon S3 estão disponíveis por meio do Amazon SQS, Amazon SNS ou AWS Lambda. Para obter mais informações, consulte [Notificações de eventos do Amazon S3 \(p. 1083\)](#).

Melhores práticas e diretrizes do S3 RTC

Ao replicar dados no Amazon S3 usando o Controle do tempo de replicação do S3 (S3 RTC), siga estas diretrizes de melhores práticas para otimizar a performance da replicação para suas workloads.

Tópicos

- [Diretrizes de performance da taxa de replicação e de solicitação do Amazon S3 \(p. 861\)](#)
- [Estimar as taxas de solicitação de replicação \(p. 861\)](#)
- [Exceder os limites da taxa de transferência de dados do S3 RTC \(p. 862\)](#)
- [Taxas de solicitação de replicação de objetos criptografados do AWS KMS \(p. 862\)](#)

Diretrizes de performance da taxa de replicação e de solicitação do Amazon S3

Ao fazer upload e recuperar armazenamento do Amazon S3, as aplicações podem realizar milhares de transações por segundo em performance de solicitação. Por exemplo, um aplicativo pode atingir pelo menos 3.500 solicitações PUT/COPY/POST/DELETE ou 5.500 solicitações GET/HEAD por segundo por prefixo em um bucket do S3, incluindo as solicitações que a replicação do S3 faz em seu nome. Não há limite para o número de prefixes em um bucket. Você pode aumentar sua performance de leitura ou gravação paralelizando as leituras. Por exemplo, se você criar 10 prefixes em um bucket do S3 para paralelizar leituras, poderá escalar a performance de leitura para 55.000 solicitações de leitura por segundo.

O Amazon S3 é dimensionado automaticamente em resposta a taxas de solicitação sustentadas acima dessas diretrizes ou taxas de solicitação sustentadas simultâneas às solicitações LIST. Enquanto o Amazon S3 estiver sendo otimizado internamente para a nova taxa de solicitação, você pode receber respostas de solicitação HTTP 503 temporariamente até que a otimização seja concluída. Isso pode ocorrer com aumentos nas taxas de solicitação por segundo ou quando você habilita o S3 RTC pela primeira vez. Durante esses períodos, sua latência de replicação pode aumentar. O acordo de nível de serviço (SLA) do S3 RTC não se aplica a períodos em que as diretrizes de performance do Amazon S3 em solicitações por segundo são excedidas.

O SLA do S3 RTC também não se aplica durante períodos em que a taxa de transferência de dados de replicação excede o limite padrão de 1 Gbps. Se você espera que a taxa de transferência de replicação exceda 1 Gbps, entre em contato com o [AWS Support Center](#) ou use o [Service Quotas](#) para solicitar um aumento no limite.

Estimar as taxas de solicitação de replicação

A taxa total de solicitações, incluindo as solicitações que a replicação do Amazon S3 faz em seu nome, deve estar dentro das diretrizes de taxa de solicitação do Amazon S3 para os buckets de origem e destino

da replicação. Para cada objeto replicado, a replicação do Amazon S3 faz até cinco solicitações GET/HEAD e uma solicitação PUT para o bucket de origem e uma solicitação PUT para cada bucket de destino.

Por exemplo, se você espera replicar 100 objetos por segundo, a replicação do Amazon S3 pode executar mais 100 solicitações PUT em seu nome para um total de 200 PUTs por segundo para o bucket do S3 de origem. A replicação do Amazon S3 também pode executar até 500 GET/HEAD (5 solicitações GET/HEAD para cada objeto replicado).

Note

Você será cobrado pelos custos de apenas uma solicitação PUT por objeto replicado. Para obter mais informações, consulte as informações sobre definição de preço nas [Perguntas frequentes sobre replicação do Amazon S3](#).

[Exceder os limites da taxa de transferência de dados do S3 RTC](#)

Se você espera que a taxa de transferência de dados do S3 Replication Time Control exceda o limite padrão de 1 Gbps, entre em contato com o [AWS Support Center](#) ou use o [Service Quotas](#) para solicitar um aumento no limite.

[Taxas de solicitação de replicação de objetos criptografados do AWS KMS](#)

Quando você replica objetos criptografados com criptografia do lado do servidor (SSE-KMS) usando a replicação do Amazon S3, são aplicados limites de solicitações por segundo do AWS Key Management Service (AWS KMS). O AWS KMS pode rejeitar uma solicitação válida, pois a taxa de solicitação excede o limite do número de solicitações por segundo. Quando uma solicitação é rejeitada, o AWS KMS retorna um erro `ThrottlingException`. O limite de taxa de solicitação do AWS KMS se aplica a solicitações feitas diretamente e a solicitações feitas pela replicação do Amazon S3 em seu nome.

Por exemplo, se você espera replicar 1.000 objetos por segundo, poderá subtrair 2.000 solicitações do limite de taxa de solicitação do AWS KMS. A taxa de solicitação resultante por segundo está disponível para as workloads do AWS KMS excluindo a replicação. Você pode usar [métricas de solicitação do AWS KMS no Amazon CloudWatch](#) para monitorar a taxa total de solicitações do AWS KMS em sua Conta da AWS.

[Replicação de marcadores de exclusão entre intervalos](#)

Por padrão, quando a replicação do Amazon S3 está habilitada e um objeto é excluído no bucket de origem, o Amazon S3 adiciona um marcador de exclusão somente no bucket de origem. Esta ação protege os dados contra exclusões mal-intencionadas.

Se você tiver a replicação de marcadores de exclusão ativada, esses marcadores serão copiados para os buckets de destino e o Amazon S3 se comporta como se o objeto tivesse sido excluído nos buckets de origem e de destino. Para obter mais informações sobre como funcionam os marcadores de exclusão, consulte [Trabalhar com marcadores de exclusão \(p. 702\)](#).

Note

A replicação de marcadores de exclusão não é suportada para regras de replicação baseadas em tags. A replicação de marcadores de exclusão também não adere ao SLA de 15 minutos concedido ao usar o Controle de tempo de replicação do S3.

Se você não estiver usando a versão de configuração de replicação mais recente, as operações de exclusão afetarão a replicação de forma diferente. Para obter mais informações, consulte [Como a exclusão de operações afeta a replicação \(p. 805\)](#).

[Habilitando a replicação de marcadores de exclusão](#)

Você pode começar a usar a replicação de marcadores de exclusão com uma regra de replicação nova ou existente. Você pode aplicá-lo a um bucket inteiro do S3 ou a objetos do Amazon S3 que tenham um prefixo específico.

Para habilitar a replicação de marcadores de exclusão usando o console do Amazon S3, consulte [Uso do console do S3 \(p. 823\)](#). Este tópico fornece instruções para habilitar a replicação de marcadores de exclusão na configuração de replicação quando os buckets pertencem às mesmas Contas da AWS ou a contas diferentes.

Para habilitar a replicação de marcadores de exclusão usando a AWS Command Line Interface (AWS CLI), você deve adicionar uma configuração de replicação ao bucket de origem com `DeleteMarkerReplication` habilitado.

No exemplo de configuração a seguir, os marcadores de exclusão são replicados para o bucket de destino `DOC-EXAMPLE-BUCKET` para objetos sob o prefixo `Tax` (Imposto).

```
{  
    "Rules": [  
        {  
            "Status": "Enabled",  
            "Filter": {  
                "Prefix": "Tax"  
            },  
            "DeleteMarkerReplication": {  
                "Status": "Enabled"  
            },  
            "Destination": {  
                "Bucket": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"  
            },  
            "Priority": 1  
        }  
    ],  
    "Role": "IAM-Role-ARN"  
}
```

Para obter instruções completas sobre como criar regras de replicação por meio da AWS CLI, consulte [Configuração da replicação para buckets de origem e de destino pertencentes à mesma conta \(p. 823\)](#) na seção Demonstrações da replicação.

Replicação de alterações de metadados com sincronização de modificação de réplica do Amazon S3

A sincronização de modificação de réplica do Amazon S3 pode ajudá-lo a manter metadados de objeto, como tags, ACLs e configurações de bloqueio de objeto replicadas entre réplicas e objetos de origem. Por padrão, o Amazon S3 replica metadados dos objetos de origem somente para as réplicas. Quando a sincronização de modificação de réplica está ativada, o Amazon S3 replica as alterações de metadados feitas nas cópias da réplica de volta ao objeto de origem, tornando a replicação bidirecional.

Ativação da sincronização de modificação da réplica

Você pode usar a sincronização de modificação de réplica do Amazon S3 com regras de replicação novas ou existentes. Você pode aplicá-lo a um bucket inteiro do S3 ou a objetos do Amazon S3 que tenham um prefixo específico.

Para habilitar a sincronização de modificação de réplica usando o console do Amazon S3, consulte [Demonstrações: exemplos para configurar a replicação \(p. 822\)](#). Este tópico fornece instruções para habilitar a sincronização da modificação de réplicas na configuração de replicação quando os buckets pertencem às mesmas Contas da AWS ou a contas diferentes.

Para habilitar a sincronização da modificação de réplicas usando a AWS Command Line Interface (AWS CLI), você deve adicionar uma configuração de replicação ao bucket que contém as réplicas com `ReplicaModifications` habilitado. Para configurar a replicação bidirecional, crie uma regra de

replicação diretamente do bucket de origem ([DOC-EXAMPLE-BUCKET1](#)) para o bucket que contém as réplicas ([DOC-EXAMPLE-BUCKET2](#)). Em seguida, crie uma segunda regra de replicação diretamente do bucket que contém as réplicas ([DOC-EXAMPLE-BUCKET2](#)) para o bucket de origem ([DOC-EXAMPLE-BUCKET1](#)). Os buckets podem estar na mesma região ou em diferentes Regiões da AWS.

Note

Você deve habilitar a sincronização de modificação de réplica em ambos os buckets para replicar alterações de metadados de réplica, como listas de controle de acesso (ACLs) a objetos, tags de objeto ou configurações de bloqueio de objeto nos objetos replicados. Como todas as regras de replicação, essas regras podem ser aplicadas a todo o bucket do Amazon S3 ou a um subconjunto de objetos do Amazon S3 filtrados com base em prefixo ou tags de objeto.

No exemplo de configuração a seguir, o Amazon S3 replica alterações de metadados sob o prefixo **Tax** (Imposto) para o bucket [DOC-EXAMPLE-BUCKET](#), que conteria os objetos de origem.

```
{  
    "Rules": [  
        {  
            "Status": "Enabled",  
            "Filter": {  
                "Prefix": "Tax"  
            },  
            "SourceSelectionCriteria": {  
                "ReplicaModifications": {  
                    "Status": "Enabled"  
                }  
            },  
            "Destination": {  
                "Bucket": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"  
            },  
            "Priority": 1  
        }  
    ],  
    "Role": "IAM-Role-ARN"  
}
```

Para obter instruções completas sobre como criar regras de replicação usando a AWS CLI, consulte [Configuração da replicação para buckets de origem e de destino pertencentes à mesma conta \(p. 823\)](#).

Alterar o proprietário da réplica

Na replicação, por padrão o proprietário do objeto de origem também é proprietário da réplica. Quando os buckets de origem e destino pertencerem a Contas da AWS diferentes e você quiser alterar a propriedade da réplica para a Conta da AWS que possui os buckets de destino, 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. Para obter mais informações sobre a opção de substituição do proprietário, consulte [Adicionar a opção de substituição do proprietário à configuração da replicação \(p. 865\)](#). Para obter mais informações sobre como definir a configuração de replicação, consulte [Replicação de objetos \(p. 801\)](#).

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 ao Amazon S3 permissões 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.

Para obter mais informações, consulte [Adicionar a opção de substituição do proprietário à configuração da replicação \(p. 865\)](#). Para ver um exemplo funcional com instruções detalhadas, consulte [Alterar o proprietário da réplica quando os buckets de origem e de destino forem de propriedade de contas diferentes \(p. 836\)](#).

Configuração aplicada pelo proprietário do bucket em Object Ownership

Quando você usa Amazon S3 Replication e os buckets de origem e destino pertencem a Contas da AWS diferentes, o proprietário do bucket de destino pode desabilitar ACLs (com a configuração aplicada de proprietário do bucket para a propriedade do bucket) para alterar o proprietário da réplica para a Conta da AWS que é a proprietária do bucket de destino. Essa configuração imita o comportamento de substituição do proprietário existente sem a necessidade da permissão `s3:ObjectOwnerOverrideToBucketOwner`. Com isso, todos os objetos que são replicados para o bucket de destino com a configuração aplicada pelo proprietário do bucket pertencem ao proprietário do bucket de destino. Para obter informações sobre o Object Ownership, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Adicionar a opção de substituição do proprietário à configuração da replicação

Warning

Adicione a opção de substituição do proprietário somente quando os buckets de origem e de destino pertencerem a Contas da AWS diferentes. O Amazon S3 não verifica se os buckets pertencem à mesma conta ou a contas diferentes. Se você adicionar a substituição do proprietário quando os dois 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 `Destination`:

- 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 Tax para replicar os objetos que têm o prefixo de chaves Amazon S3 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>
<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>
```

Conceder permissão do Amazon S3 para alterar a propriedade da réplica

Conceda permissões do Amazon S3 para alterar a propriedade da réplica adicionando permissão para a ação `s3:ObjectOwnerOverrideToBucketOwner` na política de permissões associada à função do IAM. Essa é a função do IAM especificada na configuração de replicação que permite que o Amazon S3 assuma e replique objetos em seu nome.

```
...
{
    "Effect": "Allow",
    "Action": [
        "s3:ObjectOwnerOverrideToBucketOwner"
    ],
    "Resource": "arn:aws:s3:::destination-bucket/*"
}
...
```

Adicionar a 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 destino aceite a propriedade das réplicas do objeto. O exemplo de declaração de política do bucket 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 possui a réplica. O Amazon S3 replica a versão do objeto 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, ocorrerá a situação a seguir.

- 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 aos buckets 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.

Replicação de objetos criados com criptografia no lado do servidor (SSE) usando chaves do KMS

Por padrão, o Amazon S3 não replica objetos armazenados em repouso usando criptografia no lado do servidor com chaves gerenciadas pelo cliente armazenadas no AWS KMS. Esta seção explica outras configurações que você adiciona para orientar o Amazon S3 a replicar esses objetos.

Note

Você pode usar uma AWS KMS keys de várias regiões no Amazon S3. No entanto, o Amazon S3 trata no momento as chaves de várias regiões como se fossem chaves de região única e não usa os recursos de várias regiões da chave. Para obter mais informações, consulte [Usar chaves de várias regiões](#) no Guia do desenvolvedor do AWS Key Management Service.

Para obter um exemplo com instruções passo a passo, consulte [Replicar objetos criptografados \(p. 841\)](#). Para obter informações sobre como criar uma configuração da replicação, consulte [Replicação de objetos \(p. 801\)](#).

Important

A replicação de dados criptografados é um processo no lado do servidor que ocorre totalmente dentro do Amazon S3. Objetos criados com criptografia do lado do servidor usando as chaves de criptografia fornecidas pelo usuário (SSE-C) não serão replicados.

Tópicos

- [Especificar informações adicionais na configuração de replicação \(p. 868\)](#)
- [Conceder permissões adicionais para a função do IAM \(p. 869\)](#)
- [Conceder permissões adicionais para cenários entre contas \(p. 872\)](#)
- [Considerações sobre o limite de transação do AWS KMS \(p. 872\)](#)

Especificar informações adicionais na configuração de replicação

Na configuração de replicação, você faz o seguinte:

- Na configuração de Destination, adicione a chave gerenciada pelo cliente do AWS KMS de criptografia simétrica que você deseja que o Amazon S3 use para criptografar réplicas de objetos.
- Aceite explicitamente ao habilitar a replicação de objetos criptografados usando as chaves do KMS adicionando o elemento SourceSelectionCriteria.

```
<ReplicationConfiguration>
  <Rule>
    ...
    <SourceSelectionCriteria>
      <SseKmsEncryptedObjects>
        <Status>Enabled</Status>
      </SseKmsEncryptedObjects>
    </SourceSelectionCriteria>

    <Destination>
      ...
      <EncryptionConfiguration>
        <ReplicaKmsKeyId>AWS KMS key ID for the Região da AWS of the destination bucket.</ReplicaKmsKeyId>
      </EncryptionConfiguration>
    </Destination>
    ...
  </Rule>
</ReplicationConfiguration>
```

Important

A chave do KMS deve ter sido criada na mesma Região da AWS que os buckets de destino. A chave do KMS deve ser válida. A API de replicação do bucket de PUT não verifica a validade de chaves do KMS. Se você usar uma chave do KMS inválida, receberá o código de status 200 OK como resposta, mas a replicação falhará.

O exemplo a seguir mostra uma configuração de replicação, que inclui 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 Região da AWS of the destination buckets (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. Esta regra aplica-se aos objetos com o prefixo de chaves Tax. O Amazon S3 usa o ID da chave do AWS KMS para criptografar essas réplicas de objeto.

Conceder permissões adicionais para a função do IAM

Para replicar objetos criptografados em repouso no AWS Key Management Service (AWS KMS), conceda as permissões adicionais a seguir à função do IAM especificada na configuração da replicação. Você concede essas permissões ao atualizar a política de permissões associada à função do IAM. Objetos criados com criptografia do lado do servidor usando as chaves de criptografia fornecidas pelo usuário (SSE-C) não serão replicados.

- Ação **s3:GetObjectVersionForReplication** para objetos de origem: permite que o Amazon S3 replique objetos não criptografados e objetos criados com criptografia no lado do servidor por meio de chaves de criptografia gerenciadas pelo Amazon S3 (SSE-S3) ou chaves de 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. Além disso, a permissão para a ação `s3:GetObjectVersion` permite a replicação de objetos não criptografados e objetos criptografados com SSE-S3, mas não de objetos criados usando uma chave do KMS.

- Ações **kms:Decrypt** e **kms:Encrypt** do AWS KMS:

- Permissões `kms:Decrypt` para a chave do KMS usada para descriptografar o objeto de origem
- Permissões de `kms:Encrypt` para a chave do KMS utilizada na criptografia da réplica de objetos

Note

Essa política de chave permite usar as políticas do IAM para controlar o acesso à chave do KMS. Se a política de chaves não permitir, as políticas do IAM que tentarem controlar o acesso a uma chave do KMS não serão eficazes. Para obter mais informações, consulte [Política de chaves padrão](#) no Guia do desenvolvedor do AWS Key Management Service.

Recomendamos que você restrinja essas permissões apenas aos buckets e objetos de destino usando chaves de condição do AWS KMS. A Conta da AWS proprietária da função do IAM precisa ter permissões para essas ações do AWS KMS (`kms:Encrypt` e `kms:Decrypt`) para as chaves do KMS listadas na política. Se as chaves do KMS pertencerem a outra Conta da AWS, o proprietário da chave do KMS precisará conceder essas permissões à Conta da AWS da proprietária da função do IAM. Para obter mais informações sobre como gerenciar o acesso a essas chaves do KMS, consulte [Usar políticas do IAM com o AWS KMS](#) no Guia do desenvolvedor do AWS Key Management Service.

Chaves de bucket do Amazon S3 e replicação

Quando uma chave de bucket do S3 estiver habilitada para o bucket de origem ou de destino, o contexto de criptografia será o Amazon Resource Name (ARN) do bucket e não o ARN do objeto, por exemplo, `arn:aws:s3:::bucket_ARN`. Você precisa atualizar suas políticas do IAM para usar o ARN de bucket para o contexto de criptografia. No entanto, se uma chave de bucket do S3 estiver habilitada somente no bucket de destino e não no bucket de origem, você não precisará atualizar suas políticas do IAM para usar o ARN do bucket para o contexto de criptografia.

O exemplo abaixo mostra o contexto de criptografia com o ARN de bucket.

```
"kms:EncryptionContext:aws:s3:arn": [
    "arn:aws:s3:::bucket_ARN"
]
```

Para obter mais informações, consulte [Contexto de criptografia \(p. 362\)](#) e [Alterações na observação antes de habilitar uma chave de bucket do S3 \(p. 371\)](#).

Exemplo de políticas: usar criptografia do lado do servidor AWS KMS (SSE-KMS) com replicação

Os exemplos de políticas do IAM a seguir mostram instruções para o uso de criptografia do lado do servidor AWS KMS com replicação.

Neste exemplo, o contexto de criptografia é o ARN do objeto. Se você usar o SSE-KMS com uma chave de bucket do S3 habilitada, use o ARN do bucket como o contexto de criptografia. Para obter mais informações, consulte [Contexto de criptografia \(p. 362\)](#).

Example Usar criptografia lateral do servidor do AWS KMS (SSE-KMS) — buckets de destino separados

A política de exemplo a seguir mostra instruções para usar o AWS KMS com buckets de destino separados.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": ["kms:Decrypt"],
            "Effect": "Allow",
            "Resource": "List of AWS KMS key ARNs used to encrypt source objects.",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": "s3.source-bucket-region.amazonaws.com",
                    "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::source-bucket-name/key-prefix1/*"
                }
            }
        },
        {
            "Action": ["kms:Encrypt"],
            "Effect": "Allow",
            "Resource": "AWS KMS key ARNs (for the Região da AWS of the destination bucket 1). Used to encrypt object replicas created in destination bucket 1.",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": "s3.destination-bucket-1-region.amazonaws.com",
                    "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::destination-bucket-name-1/key-prefix1/*"
                }
            }
        },
        {
            "Action": ["kms:Encrypt"],
            "Effect": "Allow",
            "Resource": "AWS KMS key ARNs (for the Região da AWS of destination bucket 2). Used to encrypt object replicas created in destination bucket 2.",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": "s3.destination-bucket-2-region.amazonaws.com",
                    "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::destination-bucket-name-2/key-prefix1/*"
                }
            }
        }
    ]
}
```

```
    ]  
}
```

Example Replicação de objetos criados com criptografia no lado do servidor usando chaves de criptografia gerenciadas pelo Amazon S3 e chaves do KMS.

Veja a seguir uma política completa do IAM que concede as permissões necessárias para replicar objetos não criptografados, objetos criados com criptografia no lado do servidor usando chaves de criptografia gerenciadas pelo Amazon S3 e chaves de 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": [  
                "s3GetObjectVersionForReplication",  
                "s3GetObjectVersionAcl"  
            ],  
            "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 ARNs 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 ARNs (for the Região da AWS of the destination buckets) to use for encrypting object replicas"
]
}
```

Conceder permissões adicionais para cenários entre contas

Em um cenário entre contas, no qual os buckets de **origem** e **destino** pertencem a diferentes Contas da AWS, é possível usar uma chave gerenciada pelo cliente para criptografar réplicas de objetos. No entanto, o proprietário da chave do KMS deve conceder ao proprietário do bucket de origem permissão para usar a chave do KMS.

Para conceder ao proprietário do bucket de origem permissão para usar a chave do KMS (console do IAM)

1. Faça login no AWS Management Console e abra o console do AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar a Região da AWS, use o Region selector (Seletor de regiões) no canto superior direito da página.
3. Para exibir as chaves em sua conta que você cria e gerencia, no painel de navegação, escolha Customer managed keys (Chaves gerenciadas de cliente).
4. Selecione a chave do KMS.
5. Em General configuration (Configurações gerais), selecione a guia Key policy (Política de chaves).
6. Escolha Other Conta da AWSs [Outra(s) Conta da AWS].
7. Selecione Add another Conta da AWS (Adicionar outra Conta da AWS).
8. Em arn:aws:iam::, insira o ID da conta do bucket de origem.
9. Escolha Save Changes (Salvar alterações).

Para conceder ao proprietário do bucket de origem permissão para usar a chave do KMS (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](#) na Referência de API do AWS Key Management Service.

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 AWS KMS por segundo excede o limite atual.

Para obter mais informações, consulte [Limites](#) no Guia do desenvolvedor do AWS Key Management Service.

Para solicitar um aumento de limite, use Service Quotas. Para obter mais informações, consulte [Limites da Amazon Web Services](#). Se não houver compatibilidade de Service Quotas com sua região, [abra um caso do AWS Support](#).

Obtenção de informações sobre o status da replicação

O status da replicação pode ajudá-lo a determinar o estado atual de um objeto que está sendo replicado. O status de replicação de um objeto de origem retornará PENDINGCOMPLETED, ou FAILED. O status de replicação de uma réplica retornará REPLICIA.

Tópicos

- [Visão geral do status da replicação \(p. 873\)](#)
- [Status da replicação, se replicar para vários intervalos de destino \(p. 874\)](#)
- [Status da replicação se a sincronização de modificação de réplica do Amazon S3 estiver ativada \(p. 874\)](#)
- [Localização do status de replicação \(p. 874\)](#)

Visão geral do status da replicação

Na replicação, 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 o objeto GET) ou metadados de objeto (usando o objeto HEAD) 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. Os objetos mudam para um estado FAILED em caso de problemas como a ausência das permissões da função de replicação, do AWS KMS ou do bucket. Para falhas temporárias, por exemplo, se um bucket ou região não estiver disponível, o status da replicação não fará a transição para FAILED, mas permanecerá PENDING. Depois que o recurso estiver online novamente, o S3 retomará a replicação desses objetos.

- 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 REPLICIA.

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.

Status da replicação, se replicar para vários intervalos de destino

Quando você replica objetos para vários intervalos de destino, o cabeçalho `x-amz-replication-status` age de forma diferente. O cabeçalho do objeto de origem retorna apenas um valor de `COMPLETED` quando a replicação é bem-sucedida para todos os destinos. O cabeçalho permanece no valor `PENDING` até que a replicação tenha sido concluída para todos os destinos. Se um ou mais destinos falharem na replicação, o cabeçalho retornará `FAILED`.

Status da replicação se a sincronização de modificação de réplica do Amazon S3 estiver ativada

Quando suas regras de replicação habilitam a sincronização de modificação de réplica do Simple Storage Service (Amazon S3), as réplicas podem informar um status diferente de `REPLICA`. Se alterações de metadados estiverem no processo de replicação, o cabeçalho do `x-amz-replication-status` retorna `PENDING`. Se a sincronização de modificação de réplica falhar ao replicar metadados, o cabeçalho retornará `FAILED`. Se os metadados forem replicados corretamente, as réplicas retornarão o cabeçalho `REPLICA`.

Localização do status de replicação

Para obter o status de replicação dos objetos em um bucket, você pode usar a ferramenta Amazon S3 Inventory. O Amazon S3 envia um arquivo CSV para o bucket de destino especificado na configuração de inventário. Você também pode usar o Amazon Athena para consultar o status da replicação no relatório de inventário. Para obter mais informações sobre o Amazon S3 Inventory, consulte [Inventário do Amazon S3 \(p. 786\)](#).

Você também pode encontrar o status de replicação do objeto usando o console, a AWS Command Line Interface (AWS CLI) ou o AWS SDK.

Uso do console do S3

No console do S3, você pode exibir o status da replicação de um objeto na página Details (Detalhes) do objeto na Object management overview (Visão geral de gerenciamento de objeto).

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets (Buckets), escolha o nome do bucket.
3. Na lista Objects (Objetos), escolha o nome do objeto.
A página Details (Detalhes) do objeto é aberta.
4. Em Object management overview (Visão geral do gerenciamento de objetos), você pode ver o Replication status (Status da replicação).

Usar a AWS CLI

Use o comando `head-object` para recuperar metadados do objeto, como segue.

```
aws s3api head-object --bucket source-bucket --key object-key --version-id object-version-id
```

O comando retorna os metadados do objeto, inclusive `ReplicationStatus`, conforme exibido no exemplo de resposta 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":{

}

}
```

Uso da SDKs AWS

Os fragmentos de código a seguir obtêm status de replicação com o AWS SDK for Java e o AWS SDK for .NET, respectivamente.

Java

```
GetObjectMetadataRequest metadataRequest = new GetObjectMetadataRequest(bucketName,
    key);
ObjectMetadata metadata = s3Client.getObjectMetadata(metadataRequest);

System.out.println("Replication Status : " +
    metadata.getRawMetadataValue(Headers.OBJECT_REPLICATION_STATUS));
```

.NET

```
GetObjectMetadataRequest getmetadataRequest = new GetObjectMetadataRequest
{
    BucketName = sourceBucket,
    Key        = objectKey
};

GetObjectMetadataResponse getmetadataResponse =
    client.GetObjectMetadata(getmetadataRequest);
Console.WriteLine("Object replication status: {0}",
    getmetadataResponse.ReplicationStatus);
```

Solução de problemas de replicação

Se as réplicas dos objetos não aparecerem no bucket de destino depois de configurar a replicação, use as dicas a seguir para identificar e corrigir os problemas.

- Para a maioria dos objetos, a replicação ocorre em até 15 minutos, mas, às vezes, pode levar algumas horas. Raramente a replicação pode levar mais tempo. O tempo que o Amazon S3 leva para replicar um objeto depende de vários fatores, incluindo o par de regiões de origem e destino e o 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 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) dos buckets 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.
- Verifique se o versionamento não foi suspenso em nenhum bucket. Tanto o bucket de origem quanto o de destino devem ter o versionamento habilitado.
- Se for conceder a propriedade do objeto ao proprietário do bucket, você deverá adicionar a ação `s3:ObjectOwnerOverrideToBucketOwner` à política de permissões associada à função do IAM. Essa é a função do IAM especificada na configuração de replicação que permite que o Amazon S3 assuma e replique objetos em seu nome.
- 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 permita ao proprietário do bucket de origem replicar objetos. Para ver um exemplo, consulte [Configurar a replicação quando os buckets de origem e destino pertencerem a contas diferentes \(p. 835\)](#).
- 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 do bucket de origem pode conceder a outras Contas da AWS permissão para carregar 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 \(p. 526\)](#).
- 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 replicará o objeto.
- A replicação falhará se a política de bucket negar o acesso à função de replicação para qualquer uma das seguintes ações:

Bucket de origem:

```
"s3:GetReplicationConfiguration",
"s3>ListBucket",
"s3:GetObjectVersionForReplication",
"s3:GetObjectVersionAcl",
"s3:GetObjectVersionTagging"
```

Buckets de destino:

```
"s3:ReplicateObject",
"s3:ReplicateDelete",
"s3:ReplicateTags"
```

Tópicos relacionados

[Replicação de objetos \(p. 801\)](#)

Considerações adicionais

O Amazon S3 também oferece suporte às configurações do bucket para o seguinte:

- Versionamento: para obter mais informações, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).
- Hospedagem de sites: para obter mais informações, consulte [Hospedagem de um site estático usando o Amazon S3 \(p. 1188\)](#).
- Acesso ao bucket por meio de uma política ou lista de controle de acesso (ACL) — Para obter mais informações, consulte [Políticas de bucket e políticas de usuário \(p. 437\)](#) e [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#).
- Armazenamento de logs: para obter mais informações, consulte [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#).
- Gerenciamento do ciclo de vida dos objetos de um bucket: para obter mais informações, consulte [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

Este tópico explica como a configuração de replicação do bucket afeta o comportamento dessas configurações do bucket.

Tópicos

- [Configuração de ciclo de vida e réplicas de objeto \(p. 877\)](#)
- [Configuração do versionamento e configuração de replicação \(p. 877\)](#)
- [Configuração de log e de replicação \(p. 878\)](#)
- [CRR e a região de destino \(p. 878\)](#)
- [Pausar a replicação \(p. 878\)](#)

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 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 em um 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.

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 do servidor ([Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#)) ou os logs do AWS CloudTrail ([Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail \(p. 1023\)](#)) estiverem habilitados no seu bucket de origem ou de destino, o Simple Storage Service (Amazon S3) incluirá neles as solicitações relativas a replicação. Por exemplo, o Simple Storage Service (Amazon S3) registra cada objeto que ele replica.

CRR e a região de destino

A replicação entre regiões (CRR) do Amazon S3 é usada para copiar objetos entre buckets do S3 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.

Suponha que você escolheu Leste dos EUA (Norte da Virgínia) (us-east-1) como a região do bucket de origem. Se você escolher Oeste dos EUA (Oregon) (us-west-2) como a região dos buckets de destino, pagará mais do que se escolher a região 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](#).

Não há cobranças de transferência de dados associadas à mesma região de replicação (SRR).

Pausar a replicação

Para pausar temporariamente a replicação, desabilite a regra em questão na configuração da replicação.

Se a replicação estiver ativada e você remover a função do IAM que concede ao Amazon S3 as permissões necessárias, a replicação falhará. O Amazon S3 relata o status da replicação para objetos afetados como FAILED.

Categorizando seu armazenamento usando tags

Use a marcação de objetos para classificar o armazenamento. Cada tag é um par de chave-valor.

Você pode adicionar tags a objetos novos ao fazer upload deles ou pode adicioná-las aos objetos existentes.

- 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.
- Para obter mais informações sobre restrições de tags, consulte [Restrições de tags definidas pelo usuário](#).

Examples

Considere os seguintes exemplos de marcação:

Example Informação de PHI

Suponha que um objeto contenha dados de informações de saúde protegidas (PHI). Você pode marcar o objeto usando o par de chave/valor a seguir.

```
PHI=True
```

ou

```
Classification=PHI
```

Example Arquivos do projeto

Suponha que você armazene arquivos de projeto em seu bucket do S3. Você pode marcar esses objetos com uma chave denominada `Project` e um valor, como mostrado a seguir.

```
Project=Blue
```

Example Várias tags

Você pode adicionar várias tags a um objeto, como mostrado a seguir.

```
Project=x
Classification=confidential
```

Prefixos e tags de nome de chave

Os prefixos de nome da chave do objeto também permitem categorizar o armazenamento. No entanto, a categorização baseada em prefixo é unidimensional. 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 `project/projectx` 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.

Benefícios adicionais

Além de classificação de dados, a marcação oferece benefícios como os seguintes:

- As tags de objeto permitem ter controle de acesso de permissões. Por exemplo, você pode conceder a um usuário do IAM permissões a objetos somente leitura com tags específicas.
- As tags de objeto permitem o gerenciamento de ciclo de vida do objeto em que você pode especificar um filtro com base em tag, além de um 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

É aceitável usar tags para identificar objetos que contêm dados confidenciais, como informações de identificação pessoal (PII) ou informações de saúde protegidas (PHI). No entanto, as próprias tags não devem conter nenhuma informação confidencial.

Adição de conjuntos de tags de objeto a vários objetos do Amazon S3 com uma única solicitação

Para adicionar conjuntos de tags de objeto a mais de um objeto do Amazon S3 com uma única solicitação, você pode usar operações em lote do S3. Você fornece às operações em lote do S3 uma lista de objetos nos quais operar. O S3 Batch Operations chama a respectiva API para executar a operação especificada. Um único trabalho de operações em lote pode realizar a operação especificada em bilhões de objetos contendo exabytes de dados.

O recurso S3 Batch Operations rastreia o progresso, envia notificações e armazena um relatório de conclusão detalhado de todas as ações, fornecendo uma experiência totalmente gerenciada, auditável e sem servidor. Use o S3 Batch Operations via AWS Management Console, AWS CLI, AWS SDKs ou API REST. Para obter mais informações, consulte [the section called “Conceitos básicos do Batch Operations” \(p. 938\)](#).

Para obter mais informações sobre tags de objeto, consulte [Gerenciar tags de objeto \(p. 884\)](#).

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 do objeto

- [Atribuição de tags de objeto PUT](#): Substitui tags em um objeto. Especifique tags no corpo da 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.

Note

Se você enviar essa solicitação com o conjunto de tags vazio, o Amazon S3 excluirá o conjunto de tags existente no objeto. Se você usar esse método, será cobrado por uma solicitação de nível 1 (PUT). Para obter mais informações, consulte [Definição de preço do Amazon S3](#).

A solicitação de [Atribuição de tags de objeto DELETE](#) é preferível, pois atinge o mesmo resultado sem incorrer em cobranças.

- [Atribuição de tags de objeto GET](#): retorna o conjunto de tags associado a um objeto. O Amazon S3 retorna tags de objetos 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 atribuição de tags

- [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 [Atribuição de tags de objeto GET](#).

- [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 a `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 atribuição de tags de objetos do S3 é muito consistente. Para obter mais informações, consulte [Modelo de consistência de dados do Amazon S3 \(p. 7\)](#).

Configurações 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 S3 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 [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

Marcação e replicação de objetos

Se você tiver configurado a replicação no bucket, o Amazon S3 replicará as tags, contanto que você conceda permissão ao Amazon S3 para ler as tags. Para obter mais informações, consulte [. Configuração da replicação \(p. 807\)](#).

Notificações de evento de marcação de objetos

É possível configurar uma notificação de evento do Amazon S3 para receber um aviso quando uma etiqueta de objeto for adicionada ou excluída de um objeto. O tipo de evento `s3:ObjectTagging:Put` notifica você quando ocorre PUT de uma etiqueta em um objeto ou quando uma etiqueta existente é atualizada. O tipo de evento `s3:ObjectTagging:Delete` notifica você quando uma etiqueta é removida de um objeto. Para obter mais informações, consulte [Habilitar notificações de eventos](#).

Para obter mais informações sobre a marcação de objetos, consulte os seguintes tópicos:

Tópicos

- [Marcação e políticas de controle de acesso \(p. 882\)](#)
- [Gerenciar tags de objeto \(p. 884\)](#)

Marcação 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:

- [Exemplo: operações de objeto \(p. 442\)](#)
- [Exemplo: operações de bucket \(p. 442\)](#)

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 podem ser usadas para conceder permissões condicionais com base em tags de objeto:

- `s3:ExistingObjectTag/<tag-key>` – Use esta 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 esta 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 esta 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 [Exemplos de chave de condição do Amazon S3 \(p. 447\)](#). 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:::awsexamplebucket1/*",
            "Principal": "*",
            "Condition": {
                "s3:ExistingObjectTag/MyTagKey": "MyTagValue"
            }
        }
    ]
}
```

```

        "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:::awsexamplebucket1/*"
            ],
            "Principal": {
                "CanonicalUser": [
                    "64-digit-alphanumeric-value"
                ],
                "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:::awsexamplebucket1/*"
            ],
            "Principal": {
                "AWS": [

```

```
        "arn:aws:iam::account-number-without-hyphens:user/username"
    ],
},
"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 Manual do usuário do IAM.

Example 3: permitir que um usuário adicione tags de objeto que incluem 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:::awsexamplebucket1/*"
            ],
            "Principal": {
                "AWS": [
                    "arn:aws:iam::account-number-without-hyphens:user/username"
                ]
            },
            "Condition": {
                "StringEquals": {
                    "s3:RequestObjectTag/Project": "X"
                }
            }
        ]
    }
}
```

Gerenciar tags de objeto

Esta seção explica como você pode gerenciar tags de objeto usando os AWS SDKs for Java e .NET ou o console do Amazon S3.

A marcação de objetos é uma forma de categorizar o armazenamento. Cada tag é um par de valores chave que segue as regras a seguir:

- 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.
- Os valores de chave e tag diferenciam maiúsculas de minúsculas.

Para obter mais informações sobre tags de objeto, consulte [Categorizando seu armazenamento usando tags \(p. 878\)](#). Para obter mais informações sobre restrições de tags, consulte [Restrições de tags definidas pelo usuário no Manual do usuário do AWS Billing and Cost Management](#).

Uso do console do S3

Para adicionar tags a um objeto

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets (Buckets), escolha o nome do bucket que contém os objetos aos quais você deseja adicionar às tags.

Você também pode, opcionalmente, navegar até uma pasta.

3. Na lista Objects (Objetos), marque a caixa de seleção ao lado dos nomes dos objetos aos quais você deseja adicionar tags.
4. No menu Actions (Ações), escolha Edit (Editar).
5. Revise os objetos listados e escolha Add tags (Adicionar tags).
6. Cada tag de objeto é um par de chave-valor. Insira uma Key (Chave) e um Value (Valor). Para adicionar outra tag, escolha Add Tag (Adicionar tag).

Você pode digitar até 10 tags para um objeto.

7. Selecione Save changes.

O Amazon S3 adiciona as tags aos objetos especificados.

Para obter mais informações, consulte também [Exibir propriedades do objeto no console do Amazon S3 \(p. 272\)](#) e [Fazer upload de objetos \(p. 171\)](#) neste guia.

Uso da SDKs AWS

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 [Categorizando seu armazenamento usando tags \(p. 878\)](#). Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

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

```
import com.amazonaws.services.s3.model.*;
import java.io.File;
import java.util.ArrayList;
import java.util.List;

public class ManagingObjectTags {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Object key ***";
        String filePath = "*** File path ***";

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

            // Create an object, add two new tags, and upload the object to Amazon S3.
            PutObjectRequest putRequest = new PutObjectRequest(bucketName, keyName, new
File(filePath));
            List<Tag> tags = new ArrayList<Tag>();
            tags.add(new Tag("Tag 1", "This is tag 1"));
            tags.add(new Tag("Tag 2", "This is tag 2"));
            putRequest.setTagging(new ObjectTagging(tags));
            PutObjectResult putResult = s3Client.putObject(putRequest);

            // Retrieve the object's tags.
            GetObjectTaggingRequest getTaggingRequest = new
GetObjectTaggingRequest(bucketName, keyName);
            GetObjectTaggingResult getTagsResult =
s3Client.getObjectTagging(getTaggingRequest);

            // Replace the object's tags with two new tags.
            List<Tag> newTags = new ArrayList<Tag>();
            newTags.add(new Tag("Tag 3", "This is tag 3"));
            newTags.add(new Tag("Tag 4", "This is tag 4"));
            s3Client.setObjectTagging(new SetObjectTaggingRequest(bucketName, keyName,
new ObjectTagging(newTags)));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

O exemplo a seguir mostra como usar o AWS SDK for .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 [Categorizando seu armazenamento usando tags \(p. 878\)](#).

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
```

```
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    public class ObjectTagsTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** key name for the new object ***";
        private const string filePath = @ "*** file path ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            PutObjectWithTagsTestAsync().Wait();
        }

        static async Task PutObjectWithTagsTestAsync()
        {
            try
            {
                // 1. Put an object with tags.
                var putRequest = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    FilePath = filePath,
                    TagSet = new List<Tag>{
                        new Tag { Key = "Keyx1", Value = "Value1" },
                        new Tag { Key = "Keyx2", Value = "Value2" }
                    }
                };

                PutObjectResponse response = await client.PutObjectAsync(putRequest);
                // 2. Retrieve the object's tags.
                GetObjectTaggingRequest getTagsRequest = new GetObjectTaggingRequest
                {
                    BucketName = bucketName,
                    Key = keyName
                };

                GetObjectTaggingResponse objectTags = await
client.GetObjectTaggingAsync(getTagsRequest);
                for (int i = 0; i < objectTags.Tagging.Count; i++)
                    Console.WriteLine("Key: {0}, Value: {1}",
objectTags.Tagging[i].Key, objectTags.Tagging[i].Value);

                // 3. Replace the tagset.

                Tagging newTagSet = new Tagging();
                newTagSet.TagSet = new List<Tag>{
                    new Tag { Key = "Key3", Value = "Value3" },
                    new Tag { Key = "Key4", Value = "Value4" }
                };

                PutObjectTaggingRequest putObjTagsRequest = new
PutObjectTaggingRequest()
                {

```

```
        BucketName = bucketName,
        Key = keyName,
        Tagging = newTagSet
    };
    PutObjectTaggingResponse response2 = await
client.PutObjectTaggingAsync(putObjTagsRequest);

    // 4. Retrieve the object's tags.
    GetObjectTaggingRequest getTagsRequest2 = new
GetObjectTaggingRequest();
    getTagsRequest2.BucketName = bucketName;
    getTagsRequest2.Key = keyName;
    GetObjectTaggingResponse objectTags2 = await
client.GetObjectTaggingAsync(getTagsRequest2);
    for (int i = 0; i < objectTags2.Tagging.Count; i++)
        Console.WriteLine("Key: {0}, Value: {1}",
objectTags2.Tagging[i].Key, objectTags2.Tagging[i].Value);

}
catch (AmazonS3Exception e)
{
    Console.WriteLine(
        "Error encountered ***. Message:'{0}' when writing an object"
        , e.Message);
}
catch (Exception e)
{
    Console.WriteLine(
        "Encountered an error. Message:'{0}' when writing an object"
        , e.Message);
}
}
}
}
```

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 [Categorizando seu armazenamento usando tags \(p. 878\)](#).

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 [Como entender os relatórios de uso e faturamento da AWS para o Amazon S3 \(p. 894\)](#)) e colunas adicionais para suas chaves de tag.

A AWS fornece dois tipos de tags de alocação de custo, uma tag gerada pela AWS e tags definidas pelo usuário. A AWS define, cria e aplica a tag `createdBy` gerada pela AWS para você após 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 etiquetas geradas pela AWS, consulte [Etiquetas de alocação de custos geradas pela AWS](#).

- Para criar tags no console, consulte [Visualização das propriedades de um bucket do S3 \(p. 134\)](#).
- Para criar tags usando a API do Amazon S3, consulte [Marcação de PUT bucket na Referência da API do Amazon Simple Storage Service](#).
- Para criar tags usando a AWS CLI, consulte [put-bucket-tagging na Referência de comandos da AWS CLI](#).
- Para obter mais informações sobre como ativar tags, consulte [Uso de tags de alocação de custos](#) no Guia do usuário do AWS Billing.

Tags de alocação de custos definidas pelo usuário

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. 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.

Tags de bucket do S3

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 50 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.

AWSA 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.

Mais informações

- [Uso de tags de alocação de custos](#) no Guia do usuário do AWS Billing.
- [Como entender os relatórios de uso e faturamento da AWS para o Amazon S3 \(p. 894\)](#)
- [AWSRelatórios de faturamento da para o Amazon S3 \(p. 890\)](#)

Relatórios de uso e faturamento dos buckets do S3

Ao usar o Amazon Simple Storage Service (Amazon S3), você não precisa pagar taxas iniciais ou se comprometer com a quantidade de conteúdo que armazenará. Quanto aos outros produtos da Amazon Web Services (AWS), você paga conforme o uso e apenas por aquilo que usa.

AWSA fornece os seguintes relatórios do Amazon S3:

- Relatórios de faturamento: vários relatórios que fornecem exibições de alto nível de toda a atividade dos serviços da AWS que você está usando, incluindo o Amazon S3. A AWS sempre cobra as tarifas do Amazon S3 do proprietário do bucket do S3, a menos que o bucket tenha sido criado como um bucket de pagamento a cargo do solicitante. Para obter mais informações sobre Pagamento pelo solicitante, consulte [Configuração de buckets de Pagamento pelo solicitante para transferências de armazenamento e uso \(p. 155\)](#). Para obter mais informações sobre relatórios de faturamento, consulte [AWSRelatórios de faturamento da para o Amazon S3 \(p. 890\)](#).
- Relatório de uso: um resumo das atividades 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 agregados. Para obter mais informações, consulte [AWSRelatório de uso da para o Amazon S3 \(p. 892\)](#).

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

Tópicos

- [AWSRelatórios de faturamento da para o Amazon S3 \(p. 890\)](#)
- [AWSRelatório de uso da para o Amazon S3 \(p. 892\)](#)
- [Como entender os relatórios de uso e faturamento da AWS para o Amazon S3 \(p. 894\)](#)

AWSRelatórios de faturamento da para o 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 [Exibição da fatura no Guia do usuário do AWS Billing](#).

Você pode configurar o AWS Cost and Usage Report para monitorar o uso da AWS e fornecer as cobranças estimadas associadas à sua conta. Para obter mais informações, consulte [O que são AWS Cost and Usage Report?](#) no Guia de relatórios de uso e custos da AWS.

Você também pode fazer download de um relatório de uso que forneça mais detalhes sobre o uso do armazenamento do Amazon S3 do que os relatórios de faturamento. Para obter mais informações, consulte [AWSRelatório de uso da para o Amazon S3 \(p. 892\)](#).

A tabela a seguir lista as taxas 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 objetos, de quanto tempo você os armazenou durante o mês e da classe de armazenamento: S3 Standard, S3 Intelligent-Tiering, S3 Standard – IA (IA para acesso pouco frequente), S3 One Zone – IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive ou Reduced Redundancy Storage (RRS). Para obter mais informações sobre classes de armazenamento, consulte Uso de classes de armazenamento do Amazon S3 (p. 731) . Esteja ciente de que, se o versionamento estiver habilitado, você será cobrado por cada versão de objeto que for retido. Para obter mais informações

Cobrança	Comentários
	sobre versionamento, consulte Como funciona o Versionamento do S3 (p. 680) .
Monitoramento e automação	Você paga uma taxa mensal de monitoramento e automação por objeto armazenado na classe de armazenamento do S3 Intelligent-Tiering para monitorar padrões de acesso e mover objetos entre camadas de acesso no S3 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 de solicitações, consulte Definição de preço do Amazon S3 .
Recuperações	Você paga para recuperar objetos que são armazenados no armazenamento S3 Standard – IA, S3 One Zone – IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive.
Exclusões adiantadas	Se excluir um objeto armazenado no armazenamento S3 Standard-IA, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive antes de terminado o compromisso mínimo de armazenamento, você pagará uma taxa de exclusão antecipada por esse objeto.
Gerenciamento de armazenamento	Você paga pelos recursos de gerenciamento de armazenamento (Amazon S3 Inventory, análise e marcação de objetos) habilitados nos buckets de sua conta.
Largura de banda	<p>Você paga por toda a largura de banda de entrada e de saída do Amazon S3, exceto:</p> <ul style="list-style-type: none"> • Dados transferidos da Internet • 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. • Dados transferidos para o Amazon CloudFront (CloudFront). <p>Você também paga uma taxa por todos os dados transferidos usando o 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 [Perguntas frequentes sobre o Amazon S3](#).

Para obter informações sobre como entender os códigos e as abreviações usadas nos relatórios de uso e faturamento do Amazon S3, consulte [Como entender os relatórios de uso e faturamento da AWS para o Amazon S3 \(p. 894\)](#).

Mais informações

- [AWSRelatório de uso da para o Amazon S3 \(p. 892\)](#)
- [Usar tags de alocação de custos para buckets do S3 \(p. 888\)](#)
- [Gerenciamento de custos e faturamento da AWS](#)
- [Preços do Amazon S3](#)
- [Perguntas frequentes sobre o Amazon S3](#)
- [Definição de preço do S3 Glacier](#)

AWSRelatório de uso da para o Amazon S3

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. Para obter mais detalhes sobre o uso do armazenamento do Amazon S3, faça download dos 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 agrupados.

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

- Serviço: Amazon S3
- Operação: a operação executada em seu bucket ou objeto. Para obter uma explicação detalhada das operações do Amazon S3, consulte [Controle de operações em seus relatórios de uso \(p. 907\)](#).
- 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 S3 Intelligent-Tiering, S3 Standard – IA, S3 One Zone-Infrequent Access (S3 One Zone – IA), S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive
- StorageObjectCount: a contagem de objetos armazenados em um determinado bucket

Para obter uma explicação detalhada dos tipos de uso do Amazon S3, consulte [Como entender os relatórios de uso e faturamento da AWS para o Amazon S3 \(p. 894\)](#).

- Recurso: o nome do bucket associado ao uso listado.
- 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: A unidade de medida comum para dados é gigabytes (GB). No entanto, dependendo do serviço e do relatório, terabytes (TB) podem aparecer.
 - O número de solicitações durante o período especificado
 - A quantidade de dados transferidos
 - A quantidade de dados armazenados em uma determinada hora

- A quantidade de dados associados a restaurações do armazenamento de S3 Standard – IA, S3 One Zone – IA, S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive

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 [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#).

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	PutObjectForRepl	USW1-Requests-SIA-Tier1	ca-example-bucket	6/1/2017 0:00	7/1/2017 0:00	178294
AmazonS3	PutObjectForRepl	USW1-USW2-AWS-In-Bytes	ca-example-bucket	6/1/2017 0:00	7/1/2017 0:00	387929083
AmazonS3	GetObjectForRepl	USW2-Requests-NoCharge	admin-created3	6/1/2017 0:00	7/1/2017 0:00	108
AmazonS3	GetObjectForRepl	USW2-USW1-AWS-Out-Bytes	my-test-bucket-bash	6/1/2017 0:00	7/1/2017 0:00	387910021

Para obter mais informações, consulte [Como entender os relatórios de uso e faturamento da AWS para o Amazon S3 \(p. 894\)](#).

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 AWS Management Console e abra o console do 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 AWS Cost and Usage Reports.
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 sobre os tipos de uso do Amazon S3, consulte [Como entender os relatórios de uso e faturamento da AWS para o Amazon S3 \(p. 894\)](#).
 - Operation (Operação): para obter uma explicação detalhada sobre as operações do Amazon S3, consulte [Controle de operações em seus relatórios de uso \(p. 907\)](#).
 - Time Period (Período): o período de cobertura do relatório.
 - Report Granularity (Granularidade do relatório): para que o relatório inclua subtotais por hora, dia ou mês.
7. Escolha o formato Download e siga as instruções para abrir ou salvar o relatório.

Mais informações

- [Como entender os relatórios de uso e faturamento da AWS para o Amazon S3 \(p. 894\)](#)
- [AWS Relatórios de faturamento da para o Amazon S3 \(p. 890\)](#)

Como entender os relatórios de uso e faturamento da AWS para o Amazon S3

Os relatórios de uso e faturamento do Amazon S3 usam códigos e abreviações. Para tipos de uso na tabela a seguir, substitua *region1*, *region2* e *region3* por abreviaturas desta lista:

- APE1: Ásia-Pacífico (Hong Kong)
- APN1: Ásia-Pacífico (Tóquio)
- APN2: Ásia-Pacífico (Seul)
- APN3: Ásia-Pacífico (Osaka)
- APS1: Ásia-Pacífico (Cingapura)
- APS2: Ásia-Pacífico (Sydney)
- APS3: Ásia-Pacífico (Mumbai)
- APS4: Ásia-Pacífico (Jacarta)
- CAN1: Canadá (Central)
- CPT: África (Cidade do Cabo)
- EUN1: UE (Estocolmo)
- EUC1: UE (Frankfurt)
- EU: UE (Irlanda)
- EUS1: Europa (Milão)
- EUW2: UE (Londres)
- EUW3: UE (Paris)
- MES1: Oriente Médio (Bahrein)
- SAE1: América do Sul (São Paulo)
- UGW1: AWS GovCloud (Oeste dos EUA)
- UGE1: AWS GovCloud (Leste dos EUA)
- USE1 (ou sem prefixo): 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 mais informações sobre preços por Região da AWS, consulte [Preços do Amazon S3](#).

A primeira coluna da tabela a seguir indica os tipos de uso que aparecem em seus relatórios de uso e faturamento. A unidade de medida comum para dados é gigabytes (GB). No entanto, dependendo do serviço e do relatório, terabytes (TB) podem aparecer.

Tipos de uso

Tipo de uso	Unidades	Granularity	Descrição
<i>region1</i> - <i>region2</i> -AWS-In-ABytes	GB	Por hora	A quantidade de dados acelerados transferidos para Região da AWS1 de Região da AWS2
<i>region1</i> - <i>region2</i> -AWS-In-ABytes-T1	GB	Por hora	A quantidade de dados acelerados T1 transferidos para Região da AWS1 de Região da AWS2, onde T1 refere-se a solicitações do

Tipo de uso	Unidades	Granularity	Descrição
			CloudFront para POPs nos Estados Unidos, Europa e Japão
<i>region1-region2-AWS-In-ABytes-T2</i>	GB	Por hora	A quantidade de dados acelerados T2 transferidos para Região da AWS1 de Região da AWS2, onde T2 refere-se a solicitações do CloudFront para POPs em todos os outros locais da borda da AWS
<i>region1-region2-AWS-In-Bytes</i>	GB	Por hora	A quantidade de dados transferidos para Região da AWS1 de Região da AWS2
<i>region1-region2-AWS-Out-ABytes</i>	GB	Por hora	A quantidade de dados acelerados transferidos de Região da AWS1 para Região da AWS2
<i>region1-region2-AWS-Out-ABytes-T1</i>	GB	Por hora	A quantidade de dados acelerados T1 transferidos de Região da AWS1 de Região da AWS2, onde T1 refere-se a solicitações do CloudFront para POPs nos Estados Unidos, Europa e Japão
<i>region1-region2-AWS-Out-ABytes-T2</i>	GB	Por hora	A quantidade de dados acelerados T2 transferidos para Região da AWS1 para Região da AWS2, onde T2 refere-se a solicitações do CloudFront para POPs em todos os outros locais da borda da AWS
<i>region1-region2-AWS-Out-Bytes</i>	GB	Por hora	A quantidade de dados transferidos de Região da AWS1 para Região da AWS2
<i>region-BatchOperations-Jobs</i>	Contagem	Por hora	O número de trabalhos de Operações em lote do S3 executados
<i>region-BatchOperations-Objects</i>	Contagem	Por hora	O número de Operações de objetos executadas pelas operações em lote do S3

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -Bulk-Retrieval-Bytes	GB	Por hora	A quantidade de dados recuperados com solicitações em massa do S3 Glacier Flexible Retrieval ou do S3 Glacier Deep Archive
<i>region</i> -BytesDeleted-GDA	GB	Mensal	A quantidade de dados excluídos por uma operação DeleteObject do armazenamento S3 Glacier Deep Archive
<i>region</i> -BytesDeleted-GIR	GB	Mensal	A quantidade de dados excluídos por uma operação DeleteObject do armazenamento S3 Glacier Instant Retrieval.
<i>region</i> -BytesDeleted-GLACIER	GB	Mensal	A quantidade de dados excluídos por uma operação DeleteObject do armazenamento S3 Glacier Flexible Retrieval
<i>region</i> -BytesDeleted-INT	GB	Mensal	A quantidade de dados excluídos por uma operação DeleteObject do armazenamento S3 Intelligent-Tiering
<i>region</i> -BytesDeleted-RRS	GB	Mensal	A quantidade de dados excluídos por uma operação DeleteObject do armazenamento RRS (Reduced Redundancy Storage, armazenamento de redundância reduzida)
<i>region</i> -BytesDeleted-SIA	GB	Mensal	A quantidade de dados excluídos por uma operação DeleteObject do armazenamento S3 Standard – IA
<i>region</i> -BytesDeleted-STANDARD	GB	Mensal	A quantidade de dados excluídos por uma operação DeleteObject do armazenamento S3 Standard
<i>region</i> -BytesDeleted-ZIA	GB	Mensal	A quantidade de dados excluídos por uma operação DeleteObject do armazenamento S3 One Zone-IA

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -C3DataTransfer-In-Bytes	GB	Por hora	A quantidade de dados transferidos para o Amazon S3 do Amazon EC2 na mesma Região da AWS
<i>region</i> -C3DataTransfer-Out-Bytes	GB	Por hora	A quantidade de dados transferidos do Amazon S3 para o Amazon EC2 na mesma Região da AWS
<i>region</i> -CloudFront-In-Bytes	GB	Por hora	A quantidade de dados transferidos para uma Região da AWS de uma distribuição do CloudFront
<i>region</i> -CloudFront-Out-Bytes	GB	Por hora	A quantidade de dados transferidos de uma Região da AWS para uma distribuição do CloudFront
<i>region</i> -DataTransfer-In-Bytes	GB	Por hora	A quantidade de dados transferidos para o Amazon S3 pela Internet
<i>region</i> -DataTransfer-Out-Bytes	GB	Por hora	A quantidade de dados transferidos do Amazon S3 para a internet ¹
<i>region</i> -DataTransfer-Regional-Bytes	GB	Por hora	A quantidade de dados transferidos do Amazon S3 para os recursos da AWS dentro da mesma Região da AWS
<i>region</i> -EarlyDelete-ByteHrs	GB-Horas	Por hora	Uso pro rata de armazenamento de objetos excluídos do armazenamento S3 Glacier Flexible Retrieval antes do término do compromisso mínimo de 90 dias ²
<i>region</i> -EarlyDelete-GDA	GB-Horas	Por hora	Uso pro rata de armazenamento de objetos excluídos do armazenamento S3 Glacier Deep Archive antes do término do compromisso mínimo de 180 dias ²

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -EarlyDelete-GIR	GB-Horas	Por hora	Uso pro rata de armazenamento de objetos excluídos do armazenamento S3 Glacier Instant Retrieval antes do término do compromisso mínimo de 90 dias.
<i>region</i> -EarlyDelete-GIR-SmObjects	GB-Horas	Por hora	Uso pro rata de armazenamento de objetos pequenos (menores que 128 KB) que foram excluídos do S3 Glacier Instant Retrieval antes que o compromisso mínimo de 90 dias terminasse.
<i>region</i> -EarlyDelete-SIA	GB-Horas	Por hora	Uso pro-rata de armazenamento de objetos excluídos do S3 Standard – IA antes do término do compromisso mínimo de 30 dias ³
<i>region</i> -EarlyDelete-SIA-SmObjects	GB-Horas	Por hora	Uso pro rata de armazenamento de objetos pequenos (menores que 128 KB) que foram excluídos do S3 Standard – IA antes que o compromisso mínimo de 30 dias terminasse ³
<i>region</i> -EarlyDelete-ZIA	GB-Horas	Por hora	Uso pro-rata de armazenamento de objetos excluídos do S3 One Zone-IA antes do término do compromisso mínimo de 30 dias ³
<i>region</i> -EarlyDelete-ZIA-SmObjects	GB-Horas	Por hora	Uso pro rata de armazenamento de objetos pequenos (menores que 128 KB) excluídos do S3 One Zone-IA antes do término do compromisso mínimo de 30 dias ³
<i>region</i> -Expedited-Retrieval-Bytes	GB	Por hora	A quantidade de dados recuperados com solicitações Expedited do S3 Glacier Flexible Retrieval

Tipo de uso	Unidades	Granularity	Descrição
<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> -Monitoring-Automation-INT	Objetos	Por hora	O número de objetos exclusivos monitorados e autonivelados na classe de armazenamento S3 Intelligent-Tiering
<i>region</i> -OverwriteBytes-Copy-GDA	GB	Mensal	A quantidade de dados substituídos por uma operação CopyObject do armazenamento S3 Glacier Deep Archive
<i>region</i> -OverwriteBytes-Copy-GIR	GB	Mensal	A quantidade de dados sobreescritos por uma operação CopyObject do armazenamento S3 Glacier Instant Retrieval.
<i>region</i> -OverwriteBytes-Copy-GLACIER	GB	Mensal	A quantidade de dados substituídos por uma operação CopyObject do armazenamento S3 Glacier Flexible Retrieval
<i>region</i> -OverwriteBytes-Copy-INT	GB	Mensal	A quantidade de dados substituídos por uma operação CopyObject do armazenamento S3 Intelligent-Tiering
<i>region</i> -OverwriteBytes-Copy-RRS	GB	Mensal	A quantidade de dados substituídos por uma operação CopyObject do armazenamento RRS (Reduced Redundancy Storage, armazenamento de redundância reduzida)
<i>region</i> -OverwriteBytes-Copy-SIA	GB	Mensal	A quantidade de dados substituídos por uma operação CopyObject do armazenamento S3 Standard – IA
<i>region</i> -OverwriteBytes-Copy-STANDARD	GB	Mensal	A quantidade de dados substituídos por uma operação CopyObject do armazenamento S3 Standard

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -OverwriteBytes-Copy-ZIA	GB	Mensal	A quantidade de dados substituídos por uma operação CopyObject do armazenamento S3 One Zone-IA
<i>region</i> -OverwriteBytes-Put-GDA	GB	Mensal	A quantidade de dados substituídos por uma operação PutObject do armazenamento S3 Glacier Deep Archive
<i>region</i> -OverwriteBytes-Put-GIR	GB	Mensal	A quantidade de dados substituídos por uma operação PutObject do armazenamento S3 Glacier Instant Retrieval.
<i>region</i> -OverwriteBytes-Put-GLACIER	GB	Mensal	A quantidade de dados substituídos por uma operação PutObject do armazenamento S3 Glacier Flexible Retrieval
<i>region</i> -OverwriteBytes-Put-INT	GB	Mensal	A quantidade de dados substituídos por uma operação PutObject do armazenamento S3 Intelligent-Tiering
<i>region</i> -OverwriteBytes-Put-RRS	GB	Mensal	A quantidade de dados sobreescritos por uma operação PutObject do armazenamento RRS (Reduzido do armazenamento de redundância)
<i>region</i> -OverwriteBytes-Put-SIA	GB	Mensal	A quantidade de dados substituídos por uma operação PutObject do armazenamento S3 Standard – IA
<i>region</i> -OverwriteBytes-Put-STANDARD	GB	Mensal	A quantidade de dados substituídos por uma operação PutObject do armazenamento S3 Standard
<i>region</i> -OverwriteBytes-Put-ZIA	GB	Mensal	A quantidade de dados substituídos por uma operação PutObject do armazenamento S3 One Zone-IA

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -Requests-GDA-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY, POST, InitiateMultipartUpload, UploadPart ou CompleteMultipartUpload em objetos do DEEP Archive
<i>region</i> -Requests-GDA-Tier2	Contagem	Por hora	O número de solicitações GET e HEAD
<i>region</i> -Requests-GDA-Tier3	Contagem	Por hora	O número de solicitações de restauração padrão do S3 Glacier Deep Archive
<i>region</i> -Requests-GDA-Tier5	Contagem	Por hora	O número de solicitações de restauração em massa do S3 Glacier Deep Archive
<i>region</i> -Requests-GIR-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY ou POST em objetos do S3 Glacier Instant Retrieval.
<i>region</i> -Requests-GIR-Tier2	Contagem	Por hora	O número de solicitações GET e todas as outras solicitações que não são GIR-Tier1 em objetos do S3 Glacier Instant Retrieval.
<i>region</i> -Requests-GLACIER-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY, POST, InitiateMultipartUpload, UploadPart ou CompleteMultipartUpload em objetos do S3 Glacier Flexible Retrieval
<i>region</i> -Requests-GLACIER-Tier2	Contagem	Por hora	O número de solicitações GET e todas as outras não listadas em objetos do S3 Glacier Flexible Retrieval
<i>region</i> -Requests-INT-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY ou POST em objetos do S3 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 do S3 Intelligent-Tiering
<i>region</i> -Requests-SIA-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY ou POST em objetos do S3 Standard-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 S3 Standard – IA
<i>region</i> -Requests-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY ou POST para STANDARD, RRS e tags, além de solicitações LIST para todos os buckets e objetos
<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 ciclo de vida para solicitações de restauração do S3 Glacier Flexible Retrieval ou do S3 Glacier Deep Archive e do S3 Glacier Flexible Retrieval padrão
<i>region</i> -Requests-Tier4	Contagem	Por hora	O número de transições do ciclo de vida para o armazenamento S3 Glacier Instant Retrieval S3 Intelligent-Tiering, S3 Standard – IA ou S3 One Zone – IA
<i>region</i> -Requests-Tier5	Contagem	Por hora	O número de solicitações de restauração Bulk do S3 Glacier Flexible Retrieval
<i>region</i> -Requests-Tier6	Contagem	Por hora	O número de solicitações de restauração Expedited do S3 Glacier Flexible Retrieval
<i>region</i> -Requests-ZIA-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY ou POST em objetos do S3 One Zone-IA
<i>region</i> -Requests-ZIA-Tier2	Contagem	Por hora	Número de solicitações GET e todas as outras solicitações non-ZIA-Tier1 em objetos do S3 One Zone-IA
<i>region</i> -Retrieval-GIR	GB	Por hora	A quantidade de dados recuperados do armazenamento S3 Glacier Instant Retrieval.

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -Retrieval-SIA	GB	Por hora	A quantidade de dados recuperados do armazenamento S3 Standard – IA
<i>region</i> -Retrieval-ZIA	GB	Por hora	A quantidade de dados recuperados do armazenamento S3 One Zone-IA
<i>region</i> -S3G-DataTransfer-In-Bytes	GB	Por hora	A quantidade de dados transferidos para o Amazon S3 para restaurar objetos do armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive
<i>region</i> -S3G-DataTransfer-Out-Bytes	GB	Por hora	A quantidade de dados transferidos do Amazon S3 para fazer a transição de objetos para o armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive
<i>region</i> -Select-Returned-Bytes	GB	Por hora	A quantidade de dados retornados com as solicitações Select do armazenamento S3 Standard
<i>region</i> -Select-Returned-GIR-Bytes	GB	Por hora	A quantidade de dados retornados com solicitações Select do armazenamento S3 Glacier Instant Retrieval.
<i>region</i> -Select-Returned-INT-Bytes	GB	Por hora	A quantidade de dados retornados com as solicitações Select do armazenamento S3 Intelligent-Tiering
<i>region</i> -Select-Returned-SIA-Bytes	GB	Por hora	A quantidade de dados retornados com as solicitações Select do armazenamento S3 Standard – IA
<i>region</i> -Select-Returned-ZIA-Bytes	GB	Por hora	A quantidade de dados retornados com as solicitações Select do armazenamento S3 One Zone-IA

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -Select-Scanned-Bytes	GB	Por hora	A quantidade de dados verificados com as solicitações Select do armazenamento S3 Standard
<i>region</i> -Select-Scanned-GIR-Bytes	GB	Por hora	A quantidade de dados verificados com solicitações Select do armazenamento S3 Glacier Instant Retrieval.
<i>region</i> -Select-Scanned-INT-Bytes	GB	Por hora	A quantidade de dados verificados com solicitações Select do armazenamento S3 Intelligent-Tiering
<i>region</i> -Select-Scanned-SIA-Bytes	GB	Por hora	A quantidade de dados verificados com as solicitações Select do armazenamento S3 Standard – IA
<i>region</i> -Select-Scanned-ZIA-Bytes	GB	Por hora	A quantidade de dados verificados com as solicitações Select do armazenamento S3 One Zone-IA
<i>region</i> -Standard-Retrieval-Bytes	GB	Por hora	A quantidade de dados recuperados com solicitações padrão do S3 Glacier Flexible Retrieval ou do S3 Glacier Deep Archive
<i>region</i> -StorageAnalytics-ObjCount	Objetos	Por hora	O número de objetos exclusivos monitorados em cada configuração de análise de classe de armazenamento.
<i>region</i> -StorageLens-ObjCount	Objetos	Diariamente	O número de objetos exclusivos em cada painel do S3 Storage Lens que são monitorados pelas recomendações e métricas avançadas do S3 Storage Lens.
<i>region</i> -StorageLensFreeTier-ObjCount	Objetos	Diariamente	O número de objetos exclusivos em cada painel do S3 Storage Lens que são monitorados pelas métricas de uso do S3 Storage Lens.

Tipo de uso	Unidades	Granularity	Descrição
<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	GB-Horas	Diariamente	O número de GB-horas em que os dados ficaram no armazenamento S3 Standard
<i>region</i> -TimedStorage-GDA-ByteHrs	GB-Horas	Diariamente	O número de GB-horas em que os dados ficaram no armazenamento S3 Glacier Deep Archive
<i>region</i> -TimedStorage-GDA-Staging	GB-Horas	Diariamente	O número de GB-horas em que os dados ficaram no armazenamento de preparação do S3 Glacier Deep Archive
<i>region</i> -TimedStorage-GIR-ByteHrs	GB-Horas	Diariamente	O número de GB-horas em que os dados ficaram no armazenamento de preparação do S3 Glacier Instant Retrieval.
<i>region</i> -TimedStorage-GIR-SmObjects	GB-Horas	Diariamente	O número de GB-horas em que objetos pequenos (menores que 128 KB) ficaram no armazenamento S3 Glacier Instant Retrieval.
<i>region</i> -TimedStorage-GlacierByteHrs	GB-Horas	Diariamente	O número de GB-horas em que os dados ficaram no armazenamento S3 Glacier Flexible Retrieval
<i>region</i> -TimedStorage-GlacierStaging	GB-Horas	Diariamente	O número de GB-horas em que os dados ficaram no armazenamento de preparação S3 Glacier Flexible Retrieval
<i>region</i> -TimedStorage-INT-FA-ByteHrs	GB-Horas	Diariamente	O número de GB-horas em que os dados ficaram armazenados no nível de acesso frequente do armazenamento S3 Intelligent-Tiering ⁵

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -TimedStorage-INT-IA-ByteHrs	GB-Horas	Diariamente	O número de GB-horas em que os dados permaneceram armazenados no nível de acesso pouco frequente do armazenamento S3 Intelligent-Tiering
<i>region</i> -TimedStorage-INT-AA-ByteHrs	GB-Horas	Diariamente	O número de GB-horas em que os dados permaneceram armazenados no nível de acesso de arquivamento do armazenamento S3 Intelligent-Tiering
<i>region</i> -TimedStorage-INT-AIA-ByteHrs	GB-Horas	Diariamente	O número de GB-horas que os dados ficaram armazenados no nível de acesso instantâneo ao arquivo do armazenamento S3 Intelligent-Tiering
<i>region</i> -TimedStorage-INT-DAA-ByteHrs	GB-Horas	Diariamente	O número de GB-horas em que os dados permaneceram armazenados no nível de acesso de arquivamento profundo do armazenamento S3 Intelligent-Tiering
<i>region</i> -TimedStorage-RRS-ByteHrs	GB-Horas	Diariamente	O número de GB-horas em que os dados ficaram no armazenamento Reduced Redundancy Storage (RRS)
<i>region</i> -TimedStorage-SIA-ByteHrs	GB-Horas	Diariamente	O número de GB-horas em que os dados ficaram no armazenamento S3 Standard – IA
<i>region</i> -TimedStorage-SIA-SmObjects	GB-Horas	Diariamente	O número de GB-horas em que pequenos objetos (menores que 128 KB) ficaram no armazenamento S3 Standard – IA ⁴
<i>region</i> -TimedStorage-ZIA-ByteHrs	GB-Horas	Diariamente	O número de GB-horas em que os dados foram armazenados no armazenamento S3 One Zone-IA

Tipo de uso	Unidades	Granularity	Descrição
region-TimedStorage-ZIA-SmObjects	GB-Horas	Diariamente	Número de GB-horas em que pequenos objetos (menores que 128 KB) ficaram no armazenamento One Zone-IA
StorageObjectCount	Contagem	Diariamente	O número de objetos armazenados em um determinado bucket

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. Quando objetos arquivados na classe de armazenamento do S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive são excluídos, substituídos ou fizerem transição para outra classe de armazenamento antes do término do compromisso de armazenamento mínimo, que é de 90 dias para o S3 Glacier Instant Retrieval e S3 Glacier Flexible Retrieval ou de 180 dias para S3 Glacier Deep Archive, há uma cobrança pro rata por gigabyte para os dias restantes.
3. Para objetos que estão no armazenamento S3 Standard – IA ou S3 One Zone – 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.
4. Para pequenos objetos (menores que 128 KB) que estão no armazenamento S3 Standard – IA ou S3 One Zone – 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. Não há tamanho mínimo de objeto faturável para objetos na classe de armazenamento S3 Intelligent-Tiering. Objetos menores do que 128 KB não são monitorados ou elegíveis para nivelamento automático. Objetos menores são sempre armazenados no nível Acesso frequente do S3 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 Todas as operações ou uma operação específica, por exemplo, `GetObject`, para relatar.

Mais informações

- [AWSRelatório de uso da para o Amazon S3 \(p. 892\)](#)
- [AWSRelatórios de faturamento da para o Amazon S3 \(p. 890\)](#)
- [Preços do Amazon S3](#)
- [Perguntas frequentes sobre o Amazon S3](#)
- [Definição de preço do S3 Glacier](#)
- [Perguntas frequentes sobre o S3 Glacier](#)

Filtragem e recuperação de dados usando o Amazon S3 Select

Com o Amazon S3 Select, é possível usar instruções de linguagem de consulta estruturada (SQL) para filtrar o conteúdo de um objeto do Amazon S3 e recuperar somente o subconjunto de dados necessário. Ao usar o Amazon S3 Select para filtrar esses dados, é possível reduzir a quantidade de dados transferidos pelo Amazon S3. Isso reduz o custo e a latência de recuperação desses dados.

O Amazon S3 Select 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 Amazon S3 Select e S3 Glacier Select. \(p. 912\)](#).

É possível executar consultas SQL usando AWS SDKs, 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 em 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 na entrada ou no resultado é de 1 MB.
- O Amazon S3 Select só pode emitir dados aninhados usando o formato de saída JSON.
- Não é possível especificar as classes de armazenamento S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive ou REDUCED_REDUNDANCY. Para obter mais informações sobre classes de armazenamento, consulte [Classes de armazenamento](#).

Limitações adicionais são aplicáveis no uso do Amazon S3 Select com objetos Parquet:

- O Amazon S3 Select é compatível somente com a compactação colunar com GZIP ou Snappy. O Amazon S3 Select não é compatível com a compactação de objetos inteiros no caso de objetos Parquet.
- O Amazon S3 Select não é compatível com a saída do Parquet. É necessário especificar o formato de saída como CSV ou JSON.
- O tamanho máximo do grupo de linhas não compactadas é 512 MB.
- 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 `IInputSerialization`. 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 [SELECTObjectContent](#) na Referência de APIs do Amazon Simple Storage Service. Também é possível um exemplo de código do SDK nas seções a seguir.

Solicitações usando intervalos de verificação

Com o Amazon S3 Select, é possível verificar um subconjunto de um objeto especificando um intervalo de bytes a ser consultado. Esse recurso permite paralelizar a verificação de todo o objeto dividindo o trabalho em solicitações separadas do Amazon S3 Select para uma série de intervalos de verificação não sobrepostos. Os intervalos de verificação não precisam estar alinhados aos limites de registro. Uma solicitação de intervalo de verificação do Amazon S3 Select é executada no intervalo de bytes especificado. Um registro que começa no intervalo de verificação especificado, mas se estende para além do intervalo de verificação, será processado pela consulta. Por exemplo; a seguir mostra um objeto do Amazon S3 que contém uma série de registros em um formato CSV delimitado por linha:

```
A, B  
C, D  
D, E  
E, F  
G, H  
I, J
```

Use o parâmetro `ScanRange` do Amazon S3 Select e inicie no (byte) 1 e termine no (byte) 4. Assim, o intervalo de verificação começaria em "," e faria a verificação até o final do registro, começando em "C" e retornaria o resultado C, D, pois esse é o final do registro.

As solicitações de intervalo de verificação do Amazon S3 Select são compatíveis com objetos Parquet, CSV (sem delimitadores entre aspas) e JSON (somente no modo LINES). Os objetos CSV e JSON devem estar descompactados. Para objetos JSON e CSV baseados em linha, quando um intervalo de verificação é especificado como parte da solicitação do Amazon S3 Select, todos os registros que começam no intervalo de verificação são processados. Para objetos Parquet, todos os grupos de linhas que começam no intervalo de verificação solicitado são processados.

As solicitações de intervalo de varredura do Amazon S3 Select estão disponíveis para uso na CLI, na API e no SDK do Amazon S3. É possível usar o parâmetro `ScanRange` na solicitação do Amazon S3 Select desse recurso. Para obter mais informações, consulte o [conteúdo do objeto do Amazon S3 Select](#) na Referência de APIs do Amazon Simple Storage Service.

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 de erro e descrições, consulte a seção [Lista de Códigos de erro de conteúdo de objetos SELECT](#) da página Respostas de erro na Referência de APIs do Amazon Simple Storage Service.

Para obter mais informações sobre o Amazon S3 Select, consulte os tópicos abaixo:

Tópicos

- [Exemplos de uso do Amazon S3 Select em objetos \(p. 910\)](#)
- [Referência SQL para Amazon S3 Select e S3 Glacier Select. \(p. 912\)](#)

Exemplos de uso do Amazon S3 Select em objetos

Você pode usar o S3 Select com a API REST do Amazon S3 e o AWS SDK para selecionar conteúdo de objetos.

Uso da API REST

Você pode usar o AWS SDK 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 SELECT](#).

Uso de AWS SDKs

Você pode usar o Amazon S3 Select para selecionar conteúdo de um objeto usando o método `selectObjectContent`, que retornará com sucesso os resultados da expressão SQL.

Java

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 como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
package com.amazonaws;

import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CSVInput;
import com.amazonaws.services.s3.model.CSVOutput;
import com.amazonaws.services.s3.model.CompressionType;
import com.amazonaws.services.s3.model.ExpressionType;
import com.amazonaws.services.s3.model.InputSerialization;
import com.amazonaws.services.s3.model.OutputSerialization;
import com.amazonaws.services.s3.model.SelectObjectContentEvent;
import com.amazonaws.services.s3.model.SelectObjectContentEventVisitor;
import com.amazonaws.services.s3.model.SelectObjectContentRequest;
import com.amazonaws.services.s3.model.SelectObjectContentResult;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.concurrent.atomic.AtomicBoolean;

import static com.amazonaws.util.IOUtils.copy;

/**
 * This example shows how to query data from S3Select and consume the response in the
 * form of an
 * InputStream of records and write it to a file.
 */

public class RecordInputStreamExample {

    private static final String BUCKET_NAME = "${my-s3-bucket}";
    private static final String CSV_OBJECT_KEY = "${my-csv-object-key}";
    private static final String S3_SELECT_RESULTS_PATH = "${my-s3-select-results-path}";
    private static final String QUERY = "select s._1 from S3Object s";
```

```
public static void main(String[] args) throws Exception {
    final AmazonS3 s3Client = AmazonS3ClientBuilder.defaultClient();

    SelectObjectContentRequest request = generateBaseCSVRequest(BUCKET_NAME,
CSV_OBJECT_KEY, QUERY);
    final AtomicBoolean isResultComplete = new AtomicBoolean(false);

    try (OutputStream fileOutputStream = new FileOutputStream(new File
(S3_SELECT_RESULTS_PATH));
         SelectObjectContentResult result = s3Client.selectObjectContent(request))
{
    InputStream resultInputStream = result.getPayload().getRecordsInputStream(
        new SelectObjectContentEventVisitor() {
            @Override
            public void visit(SelectObjectContentEvent.StatsEvent event)
            {
                System.out.println(
                    "Received Stats, Bytes Scanned: " +
event.getDetails().getBytesScanned()
                    + " Bytes Processed: " +
event.getDetails().getBytesProcessed());
            }

            /*
             * An End Event informs that the request has finished
successfully.
            */
            @Override
            public void visit(SelectObjectContentEvent.EndEvent event)
            {
                isResultComplete.set(true);
                System.out.println("Received End Event. Result is
complete.");
            }
        });
    copy(resultInputStream, fileOutputStream);
}

/*
 * The End Event indicates all matching records have been transmitted.
 * If the End Event is not received, the results may be incomplete.
 */
if (!isResultComplete.get()) {
    throw new Exception("S3 Select request was incomplete as End Event was not
received.");
}

private static SelectObjectContentRequest generateBaseCSVRequest(String bucket,
String key, String query) {
    SelectObjectContentRequest request = new SelectObjectContentRequest();
    request.setBucketName(bucket);
    request.setKey(key);
    request.setExpression(query);
    request.setExpressionType(ExpressionType.SQL);

    InputSerialization inputSerialization = new InputSerialization();
    inputSerialization.setCsv(new CSVInput());
    inputSerialization.setCompressionType(CompressionType.NONE);
    request.setInputSerialization(inputSerialization);

    OutputSerialization outputSerialization = new OutputSerialization();
    outputSerialization.setCsv(new CSVOutput());
```

```
        request.setOutputSerialization(outputSerialization);

        return request;
    }
}
```

JavaScript

Para obter um exemplo de JavaScript usando o AWS SDK for JavaScript com a API SelectObjectContent do S3 para selecionar registros dos arquivos JSON e CSV armazenados no Amazon S3, consulte a publicação de blog [Introducing support for Amazon S3 Select in the AWS SDK for JavaScript](#).

Python

Para obter um exemplo do Python sobre o uso de consultas de linguagem de consulta estruturada (SQL) para pesquisar dados carregados para o Amazon S3 como um arquivo de valor separado por vírgulas (CSV) usando o S3 Select, consulte a publicação do blog [Querying data without servers or databases using Amazon S3 Select](#).

Referência SQL para Amazon S3 Select e S3 Glacier Select.

Essa referência contém uma descrição de elementos de linguagem de consulta estruturada (SQL) que são compatíveis com o Amazon S3 Select e com o Seleção do S3 Glacier.

Tópicos

- [Comando SELECT \(p. 912\)](#)
- [Tipos de dados \(p. 918\)](#)
- [Operadores \(p. 920\)](#)
- [Palavras-chave reservadas \(p. 921\)](#)
- [Funções SQL \(p. 925\)](#)

Comando SELECT

O Seleção do Amazon S3 e o Seleção do S3 Glacier são compatíveis somente com o comando `SELECT` de SQL. As seguintes cláusulas padrão ANSI são compatíveis com `SELECT`:

- `SELECT` lista
- `FROM`Cláusula
- `WHERE`Cláusula
- Cláusula `LIMIT` (somente Seleção do Amazon S3)

Note

As consultas do Amazon S3 Select e do Seleção do S3 Glacier não são compatíveis com subconsultas nem com 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 Seleção do Amazon S3 e o Seleção do S3 Glacier são compatíveis com os seguintes formulários da cláusula FROM:

```
FROM table_name
FROM table_name alias
FROM table_name AS alias
```

Em que **table_name** é um S3Object (para o Seleção do Amazon S3), ARCHIVE ou OBJECT (para o Seleção do S3 Glacier) que faz referência 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.

Segundo o SQL padrão, a cláusula FROM cria linhas filtradas na cláusula WHERE e projetadas na lista SELECT.

Para objetos JSON armazenados no Seleção do Amazon S3, 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 Seleção do Amazon S3 emitirá o valor MISSING. Durante a serialização de saída (após a conclusão da consulta), o Seleção do Amazon S3 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": "1"}, {"expr": "y > x"}, {"id": "2", "expr": "z = DEBUG"} ]}  
{ "created": "June 27", "modified": "July 6" }
```

```
SELECT id FROM S3Object[*].Rules[*].id
```

```
{"id":"1"}  
{}  
{"id":"2"}  
{}
```

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 Seleção do Amazon S3 emitiu `MISSING`, que foi, então, 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 Seleção do Amazon S3 emitiu `MISSING`, que foi, então, alterado para um registro vazio durante a serialização de saída e retornou.

Se você não quiser que o Seleção do Amazon S3 retorne registros vazios quando 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":"1"}  
{"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": "my stuff" }, { "name": "backup" } ], "owner": "User" }
```

```
SELECT d.dir_name, d.files FROM S3Object[*] d
```

```
{"dir_name":"important_docs","files":[{"name":".."}, {"name":.."aws"}, {"name":"downloads"}]}  
{"dir_name":"other_docs","files":[{"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 booleano. Somente linhas para as quais a condição é avaliada como TRUE são retornadas no resultado.

Cláusula LIMIT (somente para o 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 S3 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 Nª 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 (somente para o 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 Seleção do S3 Glacier, 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 S3 Glacier no formato CSV com os cabeçalhos de coluna especificados e com `FileInfo` 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 Seleção do S3 Glacier 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 \(p. 921\)](#).

O exemplo a seguir é 1) um objeto do Amazon S3 ou do S3 Glacier no formato CSV com os cabeçalhos de coluna especificados, com `FileInfo` 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 Seleção do S3 Glacier 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 \(p. 929\)](#).

Tipos de dados compatíveis

O Amazon S3 Select e o Seleção do S3 Glacier são compatíveis com o conjunto de tipos de dados primitivos a seguir.

Nome	Descrição	Exemplos
<code>bool</code>	TRUE ou FALSE	<code>FALSE</code>
<code>int, inteiro</code>	Número inteiro assinado de 8 bytes no intervalo de -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807.	<code>100000</code>
<code>string</code>	String de tamanho variável codificada por UTF8. O limite padrão é um caractere. O limite máximo de caracteres é 2.147.483.647.	<code>'xyz'</code>
<code>flutuante</code>	Número de ponto flutuante de 8 bytes.	<code>CAST(0.456 AS FLOAT)</code>
<code>decimal, numérico</code>	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). Note O Amazon S3 Select ignora a escala e a precisão quando as duas são fornecidas ao mesmo tempo.	<code>123.456</code>

Nome	Descrição	Exemplos
timestamp	<p>Os time stamps representam um momento específico, sempre incluem um deslocamento local e são capazes de oferecer precisão arbitrária.</p> <p>No formato de texto, os time stamps seguem a nota W3C sobre formatos de data e hora, 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.</p>	CAST('2007-04-05T14:30Z' AS TIMESTAMP)

Tipos de Parquet compatíveis

O Amazon S3 Select é compatível com os seguintes tipos de Parquet.

- DATE
- DECIMAL
- ENUM
- INT(8)
- INT(16)
- INT(32)
- INT(64)
- LIST

Note

Para a saída do tipo de Parquet LIST, o Amazon S3 Select só é compatível com o formato JSON. No entanto, se a consulta limitar os dados a valores simples, o tipo de Parquet LIST também poderá ser consultado em formato CSV.

- STRING
- Precisão compatível com o TIMESTAMP (MILLIS/MICROS/NANOS)

Note

Não há compatibilidade com carimbos de data/hora salvos como INT(96).

Devido ao intervalo do tipo INT(64), os carimbos de data/hora que usem a unidade NANOS só podem representar valores entre 1677-09-21 00:12:43 e 2262-04-11 23:47:16. Valores fora desse intervalo não podem ser representados com a unidade NANOS.

Mapeamento de tipos de Parquet para tipos de dados compatíveis no Amazon S3 Select

Tipos de Parquet	Tipos de dados compatíveis
DATE	timestamp
DECIMAL	decimal, numérico
ENUM	string

Tipos de Parquet	Tipos de dados compatíveis
INT(8)	int, inteiro
INT(16)	int, inteiro
INT(32)	int, inteiro
INT(64)	decimal, numérico
LIST	Cada tipo de Parquet na lista é mapeado para o tipo de dados correspondente
STRING	string
TIMESTAMP	timestamp

Operadores

O Amazon S3 Select e o Seleção do S3 Glacier são compatíveis com os seguintes operadores.

Operadores lógicos

- AND
- NOT
- OR

Operadores de comparação

- <
- >
- <=
- >=
- =
- <>
- !=
- BETWEEN
- IN – Por exemplo: IN ('a', 'b', 'c')

Operadores de correspondência de padrões

- LIKE
- _ (corresponde a qualquer caractere)
- % (corresponde a qualquer sequência de caracteres)

Operadores unitários

- IS NULL
- IS NOT NULL

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

Abaixo está a lista de palavras-chave reservadas para o Amazon S3 Select e o Seleção do S3 Glacier. 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 Seleção do S3 Glacier são compatíveis com várias funções SQL.

Tópicos

- [Funções agregadas \(somente no Amazon S3 Select\) \(p. 925\)](#)
- [Funções condicionais \(p. 926\)](#)
- [Funções da conversão \(p. 929\)](#)
- [Funções de data \(p. 929\)](#)
- [Funções de string \(p. 935\)](#)

[Funções agregadas \(somente no Amazon S3 Select\)](#)

O Amazon S3 Select é compatível com as seguintes funções agregadas.

Note

O S3 Glacier Select não é compatível com as funções agregadas.

Função	Tipo de argumento	Tipo de retorno
AVG(<i>expression</i>)	INT, FLOAT, DECIMAL	DECIMAL para um argumento INT, FLOAT para um argumento de ponto flutuante, caso contrário, é igual ao tipo de dados do argumento.
COUNT	-	INT
MAX(<i>expression</i>)	INT, DECIMAL	O mesmo que o tipo de argumento.
MIN(<i>expression</i>)	INT, DECIMAL	O mesmo que o tipo de argumento.
SUM(<i>expression</i>)	INT, FLOAT, DOUBLE, DECIMAL	INT para o 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 Seleção do S3 Glacier são compatíveis com as seguintes funções condicionais.

Tópicos

- [CASE \(p. 926\)](#)
- [COALESCE \(p. 928\)](#)
- [NULLIF \(p. 928\)](#)

CASE

A expressão CASE é uma expressão condicional, semelhante às instruções if/then/else encontradas em outras linguagens. CASE é usada para especificar um resultado onde há várias condições. Há dois tipos de expressões CASE: simples e pesquisada.

Em expressões CASE simples, uma expressão é comparada a um valor. Quando uma correspondência é encontrada, a ação especificada na cláusula THEN é aplicada. Se nenhuma correspondência é encontrada, a ação especificada na cláusula ELSE é aplicada.

Em expressões CASE pesquisadas, cada CASE é avaliado com base em uma expressão booleana e a instrução CASE retorna o primeiro CASE correspondente. Se nenhum CASE correspondente é encontrado entre as cláusulas WHEN, a ação na cláusula ELSE é retornada.

Sintaxe

Note

Atualmente, o Amazon S3 Select não é compatível com ORDER BY ou consultas que contenham novas linhas. Use consultas sem quebras de linha.

Instrução CASE simples usada para correspondência de condições:

```
CASE expression WHEN value THEN result [WHEN...] [ELSE result] END
```

Instrução CASE pesquisada usada para avaliação de cada condição:

```
CASE WHEN boolean condition THEN result [WHEN ...] [ELSE result] END
```

Exemplos

Note

Se você usar o console do Amazon S3 para executar os exemplos a seguir e o arquivo CSV tiver uma linha de cabeçalho, selecione Exclude the first line of CSV data (Excluir a primeira linha de dados do CSV).

Exemplo 1: use uma expressão CASE simples para substituir New York City por Big Apple em uma consulta. Substitua todos os outros nomes de cidade por outros.

```
SELECT venuecity, CASE venuecity WHEN 'New York City' THEN 'Big Apple' ELSE 'other' END  
FROM S3Object;
```

Resultado da consulta:

venuecity	case
Los Angeles	other
New York City	Big Apple
San Francisco	other
Baltimore	other
...	

Exemplo 2: use uma expressão CASE pesquisada para atribuir números de grupo com base no valor PRICEPAID para vendas individuais de ingresso:

```
SELECT pricepaid, CASE WHEN CAST(pricepaid as FLOAT) < 10000 THEN 'group 1' WHEN  
CAST(pricepaid as FLOAT) > 10000 THEN 'group 2' ELSE 'group 3' END FROM S3Object;
```

Resultado da consulta:

pricepaid	case
12624.00	group 2
10000.00	group 3

```
10000.00 | group 3
9996.00 | group 1
9988.00 | group 1
...
```

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
```

NULIF

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

```
NULIF ( expression1, expression2 )
```

Parâmetros

expression1, expression2

As expressões de destino nas quais a função opera.

Exemplos

```
NULIF(1, 1)          -- null
NULIF(1, 2)          -- 1
NULIF(1.0, 1)        -- null
NULIF(1, '1')        -- 1
NULIF([1], [1])      -- null
NULIF(1, NULL)       -- 1
NULIF(NULL, 1)        -- null
NULIF(null, null)     -- null
```

```
NULLIF(missing, null)    -- null
NULLIF(missing, missing) -- null
```

Funções da conversão

O Amazon S3 Select e o Seleção do S3 Glacier são compatíveis com as seguintes funções de conversão.

Tópicos

- [CAST \(p. 929\)](#)

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 \(p. 918\)](#).

Exemplos

```
CAST('2007-04-05T14:30Z' AS TIMESTAMP)
CAST(0.456 AS FLOAT)
```

Funções de data

O Amazon S3 Select e o Seleção do S3 Glacier são compatíveis com as seguintes funções de data.

Tópicos

- [DATE_ADD \(p. 929\)](#)
- [DATE_DIFF \(p. 930\)](#)
- [EXTRACT \(p. 931\)](#)
- [TO_STRING \(p. 931\)](#)
- [TO_TIMESTAMP \(p. 934\)](#)
- [UTCNOW \(p. 935\)](#)

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
-- necessary)
-- 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 \(p. 929\)](#).

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`)          -- 1
DATE_DIFF(year, `2010T`, `2010-05T`)
  2010-01-01T00:00:00.000Z)                           -- 4 (2010T is equivalent to
DATE_DIFF(month, `2010T`, `2011T`)                   -- 12
DATE_DIFF(month, `2011T`, `2010T`)                   -- -12
DATE_DIFF(day, `2010-01-01T23:00`, `2010-01-02T01:00`) -- 0 (need to be at least 24h apart
  to be 1 day apart)
```

EXTRACT

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
EXTRACT(MONTH FROM `2010-10T`)                      -- 1 (equivalent to
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 12)
hh	03	Hora do dia preenchida com zeros (01 a 12)
H	3	Hora do dia (0 a 23)

Formato	Exemplo	Descrição
HH	03	Hora do dia preenchida com zeros (00 a 23)
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)
SSSS	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
XX or XXXX	+0700 or Z	Deslocamento em horas e minutos ou "Z" se o deslocamento for 0
XXX or XXXXX	+07:00 or Z	Deslocamento em horas e minutos ou "Z" se o deslocamento for 0
x	7	Deslocamento em horas
xx or xxxx	700	Deslocamento em horas e minutos
xxx or xxxxx	+07:00	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`, 'MMMd d, y h:m a')   -- "July 20, 1969 8:18 PM"
TO_STRING(`1969-07-20T20:18Z`, 'y-MM-dd''T''H:m:ssX') -- "1969-07-20T20:18:00Z"
TO_STRING(`1969-07-20T20:18+08:00Z`, 'y-MM-dd''T''H:m:ssX') -- "1969-07-20T20:18:00Z"
TO_STRING(`1969-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXX') -- "1969-07-20T20:18:00+0800"
TO_STRING(`1969-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXXX') -- "1969-07-20T20:18:00+08:00"

```

TO_TIMESTAMP

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 Seleção do S3 Glacier são compatíveis com as seguintes funções de string.

Tópicos

- [CHAR_LENGTH, CHARACTER_LENGTH \(p. 935\)](#)
- [LOWER \(p. 936\)](#)
- [SUBSTRING \(p. 936\)](#)
- [TRIM \(p. 937\)](#)
- [UPPER \(p. 937\)](#)

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 < 1`, sem comprimento especificado, ela será definida como 1.
- Se `start < 1`, sem comprimento especificado, ela será definida como `start + length - 1`.
- Se `start + length - 1 < 0`, uma string vazia será retornada.
- Se `start + length - 1 >= 0`, a substring começando no índice 1 de comprimento `start + length - 1` será retornada.

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!@#$') -- 'ABCDEFG!@#$'
```

Executar operações em lote de grande escala em objetos do Amazon S3

Você pode usar operações em lote do S3 para executar operações em lote de grande escala em objetos do Amazon S3. As operações em lote do S3 podem executar uma única operação em listas de objetos do Amazon S3 especificadas. Um único trabalho pode realizar uma operação especificada em bilhões de objetos contendo exabytes de dados. O Amazon S3 monitora o progresso, envia notificações e armazena um relatório de conclusão detalhado de todas as ações, fornecendo uma experiência totalmente gerenciada, auditável e sem servidor. Você pode usar o S3 Batch Operations por meio da AWS Management Console, AWS CLI, Amazon SDKs ou REST API.

Use as operações em lote do S3 para copiar objetos e definir tags de objetos ou listas de controle de acesso (ACLs). Também é possível iniciar restaurações de objetos no S3 Glacier Flexible Retrieval ou invocar uma função do AWS Lambda para executar ações personalizadas usando seus objetos. É possível executar essas operações em uma lista personalizada de objetos ou usar um relatório do Amazon S3 Inventory para facilitar a geração de listas de objetos. As operações em lote do Amazon S3 usam as mesmas APIs do Amazon S3 que você já usa com o Amazon S3, portanto, a interface é familiar.

Conceitos básicos sobre operações em lote do S3

Você pode usar operações em lote do S3 para executar operações em lote de grande escala em objetos do Amazon S3. As operações em lote do S3 podem executar uma única operação ou ação em listas de objetos do Amazon S3 que você especificar.

Terminologia

Esta seção usa os termos trabalhos, operações e tarefas. Confira suas definições abaixo:

Trabalho

Um trabalho é a unidade básica de trabalho para operações em lote do 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 para cada objeto no manifesto.

Operação

A operação é o tipo de [ação](#) da API, como copiar objetos, que você deseja que o trabalho de operações em lote execute. Cada trabalho executa um único tipo de operação em todos os objetos especificados no manifesto.

Tarefa

Uma tarefa é a unidade de execução para um trabalho. Uma tarefa representa uma única chamada para uma operação de API do AWS Lambda ou do Amazon S3 a fim de executar a operação do trabalho em um único objeto. Ao longo da vida útil de um trabalho, as operações em lote do S3 criam uma tarefa para cada objeto especificado no manifesto.

Como funciona um trabalho de operações em lote do S3

Um trabalho é a unidade básica de trabalho para operações em lote do S3. Uma tarefa contém todas as informações necessárias para executar a operação especificada em uma lista de objetos. Para criar um trabalho, dê uma lista de objetos às operações em lote do S3 e especifique a ação a ser realizada neles.

Para obter informações sobre as operações compatíveis com o S3 Batch Operations, consulte [Operações suportadas pelo S3 Batch Operations \(p. 953\)](#).

Um trabalho em lote realiza a operação especificada em cada objeto incluído em seu manifesto. Um manifesto lista os objetos que você deseja que um trabalho em lote processe e ele é armazenado como um objeto em um bucket. Você pode usar um relatório de [Inventário do Amazon S3 \(p. 786\)](#) formatado em CSV como um manifesto, o que facilita a criação de grandes listas de objetos localizados 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 de criar um trabalho, o Amazon S3 processará a lista de objetos no manifesto e executará a operação especificada em cada objeto. Enquanto um trabalho está em execução, é possível monitorar 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 trabalhos, consulte [Gerenciar trabalhos de operações em lote do S3 \(p. 979\)](#).

Conceder permissões para operações em lote do Amazon S3

Antes de criar e executar trabalhos do S3 Batch Operations, você deve conceder as permissões necessárias. Para criar um trabalho do Amazon S3 Batch Operations, a permissão de usuário `s3:CreateJob` é necessária. A mesma entidade que cria o trabalho deve ter a permissão `iam:PassRole` para passar a função do AWS Identity and Access Management (IAM) especificada para o trabalho ao Batch Operations.

Para obter informações gerais sobre como especificar recursos do IAM, consulte [Elementos de política JSON do IAM: Resource](#) no Guia do usuário do IAM. As seções a seguir fornecem informações sobre como criar uma função do IAM e anexar políticas.

Tópicos

- [Criar uma função do IAM das operações em lote do S3 \(p. 939\)](#)
- [Anexar políticas de permissões \(p. 940\)](#)

Criar uma função do IAM das operações em lote do S3

O Amazon S3 deve ter permissão para executar o S3 Batch Operations em seu nome. Conceda essas permissões por meio de uma função do AWS Identity and Access Management (IAM). Esta seção fornece exemplos das políticas de permissões e confiança usadas ao criar uma função do IAM. Para obter mais informações, consulte [Funções do IAM](#) no Guia do usuário do IAM. Veja exemplos em [Controlar permissões para o recurso Operações em lote do S3 usando tags de trabalho \(p. 998\)](#) e [Copiar objetos usando o S3 Batch Operations \(p. 954\)](#).

Em suas políticas do IAM, você também pode usar chaves de condição para filtrar permissões de acesso para trabalhos de operações em lote do S3. Para obter mais informações e uma lista completa das chaves de condição específicas do Amazon S3, consulte [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#).

Política de confiança

Para permitir que o principal do serviço de operações em lote do S3 assuma a função do IAM, anexe a política de confiança a seguir à função.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {
```

```
        "Effect": "Allow",
        "Principal": {
            "Service": "batchoperations.s3.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
}
```

Anexar políticas de permissões

Dependendo do tipo de operação, é possível anexar uma das seguintes políticas:

Antes de configurar as permissões, observe o seguinte:

- 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 essas permissões.
- Para manifestos de relatório do Amazon S3 Inventory, as operações em lote do S3 necessitam de permissão de leitura para o objeto manifest.json e todos os arquivos de dados CSV associados.
- Permissões específicas da versão, como s3:GetObjectVersion, somente são necessárias ao especificar o ID de versão dos objetos.
- Se você estiver executando o S3 Batch Operations em objetos criptografados, a função do IAM também deverá ter acesso às chaves do AWS KMS usadas para criptografá-las.

Copiar objetos: PutObject

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:PutObjectTagging"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::DestinationBucket/*"
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectTagging",
                "s3>ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::SourceBucket",
                "arn:aws:s3:::SourceBucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [

```

```
        "arn:aws:s3:::ManifestBucket/*"
    ],
},
{
    "Effect": "Allow",
    "Action": [
        "s3:PutObject"
    ],
    "Resource": [
        "arn:aws:s3:::ReportBucket/*"
    ]
}
]
```

Substituir marcação de objetos: PutObjectTagging

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectTagging",
                "s3:PutObjectVersionTagging"
            ],
            "Resource": "arn:aws:s3:::TargetResource/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::ManifestBucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::ReportBucket/*"
            ]
        }
    ]
}
```

Excluir marcação de objetos: DeleteObjectTagging

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3>DeleteObjectTagging",
                "s3>DeleteObjectVersionTagging"
            ],
            "Resource": [

```

```
        "arn:aws:s3:::TargetResource/*"
    ],
},
{
    "Effect": "Allow",
    "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
    ],
    "Resource": [
        "arn:aws:s3:::ManifestBucket/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "s3:PutObject"
    ],
    "Resource": [
        "arn:aws:s3:::ReportBucket/*"
    ]
}
]
```

Substituir lista de controle de acesso: PutObjectAcl

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectAcl",
                "s3:PutObjectVersionAcl"
            ],
            "Resource": "arn:aws:s3:::TargetResource/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::ManifestBucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::ReportBucket/*"
            ]
        }
    ]
}
```

Restaurar objetos: RestoreObject

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "s3:RestoreObject"
        ],
        "Resource": "arn:aws:s3:::TargetResource/*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:GetObject",
            "s3:GetObjectVersion"
        ],
        "Resource": [
            "arn:aws:s3:::ManifestBucket/*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::ReportBucket/*"
        ]
    }
]
```

Aplicar retenção do bloqueio de objetos: PutObjectRetention

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetBucketObjectLockConfiguration",
            "Resource": [
                "arn:aws:s3:::TargetResource"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectRetention",
                "s3:BypassGovernanceRetention"
            ],
            "Resource": [
                "arn:aws:s3:::TargetResource/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::ManifestBucket/*"
            ]
        }
    ]
}
```

```
        "Effect": "Allow",
        "Action": [
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::ReportBucket/*"
        ]
    }
}
```

Aplicar retenção legal do bloqueio de objetos: PutObjectLegalHold

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetBucketObjectLockConfiguration",
            "Resource": [
                "arn:aws:s3:::TargetResource"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "s3:PutObjectLegalHold",
            "Resource": [
                "arn:aws:s3:::TargetResource/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::ManifestBucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::ReportBucket/*"
            ]
        }
    ]
}
```

Replicar objetos existentes: InitiateReplication com um manifesto gerado pelo S3

Use essa política se estiver usando e armazenando um manifesto gerado pelo S3. Para obter mais informações sobre o uso de operações em lote para replicação de objetos existentes, consulte [Replicar objetos existentes com o S3 Batch Replication \(p. 849\)](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {

```

```
        "Action": [
            "s3:InitiateReplication"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::*** replication source bucket ***/*"
        ]
    },
    {
        "Action": [
            "s3:GetReplicationConfiguration",
            "s3:PutInventoryConfiguration"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::*** replication source bucket ***"
        ]
    },
    {
        "Action": [
            "s3:GetObject",
            "s3:GetObjectVersion"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::*** manifest bucket ***/*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::*** completion report bucket ****/*",
            "arn:aws:s3:::*** manifest bucket ****/*"
        ]
    }
]
```

Replicar objetos existentes: InitiateReplication com um manifesto do usuário

Use essa política se estiver usando um manifesto fornecido pelo usuário. Para obter mais informações sobre o uso de operações em lote para replicação de objetos existentes, consulte [Replicar objetos existentes com o S3 Batch Replication \(p. 849\)](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:InitiateReplication"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::*** replication source bucket ***/*"
            ]
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],

```

```
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::*** manifest bucket ***/*"
        ],
    },
{
    "Effect": "Allow",
    "Action": [
        "s3:PutObject"
    ],
    "Resource": [
        "arn:aws:s3:::*** completion report bucket ****/*"
    ]
}
}
```

Criar um trabalho de operações em lote do S3

Com o S3 Batch Operations, você pode executar operações em lote em grande escala em uma lista de objetos específicos do Amazon S3. Esta seção descreve as informações necessárias para criar um trabalho do Operações em lote do S3 e os resultados de uma solicitação `Create Job`. Ela também fornece instruções para criar um trabalho do Batch Operatons usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Ao criar um trabalho do S3 Batch Operations, você pode solicitar um relatório de conclusão para todas as tarefas ou somente para aquelas que apresentaram falha. Contanto que pelo menos uma tarefa tenha sido invocada com êxito, as operações em lote do S3 geram um relatório para tarefas que foram concluídas, que falharam ou que foram canceladas. Para obter mais informações, consulte [Exemplos: relatórios de conclusão de operações em lote do S3 \(p. 990\)](#).

Tópicos

- [Elementos da solicitação de trabalho de Operações em lote \(p. 946\)](#)
- [Especificar um manifesto \(p. 948\)](#)
- [Criar um trabalho \(p. 949\)](#)
- [Respostas do trabalho \(p. 953\)](#)

Elementos da solicitação de trabalho de Operações em lote

Para criar um trabalho de operações em lote do S3, você deve fornecer as seguintes informações:

Operação

Especifique a operação que deseja que as operações em lote do S3 execute nos objetos no manifesto. Cada tipo de operação aceita parâmetros específicos para essa operação. Isso permite que você realize as mesmas tarefas que se executasse a operação individualmente em cada objeto.

Manifesto

O manifesto é uma lista de todos os objetos em que você deseja que as operações em lote do S3 execute a ação especificada. Use um relatório em formato CSV [Inventário do Amazon S3 \(p. 786\)](#) como manifesto ou a própria lista CSV personalizada de objetos.

Se os objetos no manifesto estiverem em um bucket versionado, você deverá especificar os IDs de versão dos objetos. Para obter mais informações, consulte [Especificar um manifesto \(p. 948\)](#).

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 só têm significado em relação às prioridades definidas para outros trabalhos na mesma conta e região. Você pode escolher qualquer sistema de numeração adequado para você. Por exemplo, talvez você queira atribuir prioridade 1 a 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 do S3 priorizam trabalhos em ordem numérica, mas não é garantida rigidez a essa ordem. Por isso, você não deve usar prioridades de trabalho para garantir que um trabalho comece ou termine antes de outro. Caso precise garantir uma ordem rígida, aguarde a conclusão de uma tarefa para iniciar a próxima.

RoleArn

Especifique uma função do AWS Identity and Access Management (IAM) para executar o trabalho. A função do IAM usada 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 funções do IAM, consulte [Funções do IAM](#) no Guia do usuário do IAM.

Para obter mais informações sobre as permissões do Amazon S3, consulte [Ações do Amazon S3 \(p. 441\)](#).

Relatório

Especifique se deseja que as operações em lote do S3 gerem um relatório de conclusão. Caso solicite um relatório de conclusão do trabalho, você deve fornecer os parâmetros para o relatório neste elemento. As informações necessárias incluem o seguinte:

- O bucket no qual você deseja armazenar o relatório.
- O formato do relatório
- Se você deseja que o relatório inclua os detalhes de todas as tarefas ou apenas das tarefas que apresentaram falha.
- Uma string de prefixo opcional.

Tags (opcional)

Você pode rotular e controlar o acesso aos trabalhos de operações em lote do S3 adicionando tags. As tags podem ser usadas para identificar quem é responsável por um trabalho de operações em lote. Você pode criar trabalhos com tags anexadas a eles e pode adicionar tags a trabalhos depois de criá-los. Por exemplo, você pode conceder a um usuário do IAM permissão para invocar `CreateJob`, desde que o trabalho seja criado com a tag "Department=Finance".

Para obter mais informações, consulte [the section called “Usar tags” \(p. 992\)](#).

Description (opcional)

Para rastrear e monitorar seu trabalho, você também pode fornecer uma descrição de até 256 caracteres. O Amazon S3 inclui essa descrição sempre que retorna informações sobre um trabalho ou exibe detalhes do trabalho 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.

Especificar um manifesto

Um manifesto é um objeto do Amazon S3 que contém as chaves de objeto em que você deseja que o Simple Storage Service (Amazon S3) atue. Para criar um manifesto de uma tarefa, especifique a chave de objeto, a ETag e o ID da versão opcional do manifesto. O conteúdo do manifesto deve estar codificado em URL. Os manifestos que usam a criptografia no servidor com chaves fornecidas pelo cliente (SSE-C) não são compatíveis. Manifestos que usam criptografia no servidor com chaves AWS KMS do AWS Key Management Service (SSE-KMS) só são compatíveis quando usam relatórios de inventário no formato CSV. O manifesto deve conter o nome do bucket, a chave de objeto e, opcionalmente, a versão de cada objeto. Qualquer outro campo no manifesto não é usado pelas operações em lote do S3.

É possível especificar um manifesto na solicitação de criação de um trabalho usando um dos seguintes formatos:

- Relatório do Amazon S3 Inventory: deve ser um relatório do Amazon S3 Inventory formatado em CSV. 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 \(p. 786\)](#). Se o relatório de inventário incluir IDs de versões, o recurso Operações em lote do S3 operará nas versões especificadas do objeto.

Note

O S3 Batch Operations é compatível com relatórios de inventário no formato CSV criptografados pelo AWS KMS.

- Arquivo CSV — Cada linha no arquivo deve incluir o nome do bucket, a chave do objeto e, opcionalmente, a versão do objeto. As chaves de objeto devem ser codificadas em URL conforme mostrado nos exemplos a seguir. O manifesto deve incluir ou omitir IDs de versão para todos os objetos. Para obter mais informações sobre o formato de manifesto CSV, consulte [JobManifestSpec](#) na Referência da API do Amazon Simple Storage Service.

Note

O S3 Batch Operations não oferece suporte a arquivos de manifesto no formato CSV criptografados pelo AWS KMS.

Veja a seguir um exemplo de manifesto no formato CSV sem os IDs de versão.

```
Examplebucket,objectkey1
Examplebucket,objectkey2
Examplebucket,objectkey3
Examplebucket,photos/jpgs/objectkey4
Examplebucket,photos/jpgs/newjersey/objectkey5
Examplebucket,object%20key%20with%20spaces
```

Veja a seguir um exemplo de manifesto no formato CSV incluindo os IDs de versão.

```
Examplebucket,objectkey1,PZ9ibn9D5lP6p298B7S9_ceqx1n5EJ0p
Examplebucket,objectkey2,YY_ouuAJByNW1LRBffMfxMge7XQWxMBF
Examplebucket,objectkey3,jbo9_jhdPEyB4Rrm0xWS0kU0EoNrU_oI
Examplebucket,photos/jpgs/objectkey4,6EqlikJJxLTsHsnbZbSRffn24_eh5Ny4
Examplebucket,photos/jpgs/newjersey/objectkey5,imHf3FAiRsvBW_EHB8GOu.NHunHO1gVs
Examplebucket,object%20key%20with%20spaces,9HkPvDaZY5MVbMhn6TMn1YTb5ArQAo3w
```

Important

Ao usar um manifesto fornecido pelo usuário e um bucket versionado, recomendamos especificar os IDs da versão dos objetos. Quando você cria um trabalho, o recurso Operações em lote do

S3 analisa todo o manifesto antes de executar o trabalho. 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 um trabalho estiver em execução e não tiver especificado o ID da versão do objeto, o Simple Storage Service (Amazon S3) realizará a operação na versão mais recente do objeto, não na versão que existia quando o trabalho foi criado. A única maneira de evitar esse comportamento é especificar o ID de versão do objeto listado no manifesto.

Note

O Simple Storage Service (Amazon S3) oferece a você a opção de criar um manifesto para o trabalho de replicação do S3 Batch Replication. A replicação em lote é uma operação sob demanda que replica objetos existentes. Para obter mais informações sobre replicação em lote, consulte [Replicar objetos existentes com o S3 Batch Replication \(p. 849\)](#).

Criar um trabalho

Você pode criar trabalhos do S3 Batch Operations usando o AWS Management Console, a AWS CLI, os Amazon SDKs ou a REST API.

Para obter mais informações sobre como criar uma solicitação de trabalho, consulte [Elementos da solicitação de trabalho de Operações em lote \(p. 946\)](#).

Pré-requisito

Antes de criar um trabalho do Batch Operations, confirme se você configurou as permissões relevantes. Para obter mais informações, consulte [Conceder permissões para operações em lote do Amazon S3 \(p. 939\)](#).

Uso do console do S3

Para criar um trabalho em lote

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Selecione Batch Operations (Operações em lote) no painel de navegação do console do Amazon S3.
3. Selecione Create job (Criar trabalho).
4. Selecione a Region (região) onde você deseja criar o trabalho.
5. Em Formato do manifesto, escolha o tipo de objeto do manifesto a ser usado.
 - Se você selecionar S3 inventory report (Relatório de inventário do S3), insira o caminho para o objeto manifest.json gerado pelo Amazon S3 como parte do relatório de inventário em formato CSV e, opcionalmente, o ID de versão do objeto do manifesto a ser usado, se não for a mais recente.
 - Se você selecionar CSV, insira o caminho para o objeto do manifesto formatado em CSV. O objeto do manifesto deve seguir o formato descrito no console. Opcionalmente, inclua o ID de versão do objeto do manifesto que deseja usar, se não for a mais recente.
6. Escolha Next (Próximo).
7. Em Operation (Operação), selecione a operação a ser executada em todos os objetos no manifesto. Preencha as informações da operação escolhida e, depois, selecione Next (Próximo).
8. Preencha as informações para Configure additional options (Configurar opções adicionais) e, depois, selecione Next (Próximo).
9. Em Review (Revisão), verifique as configurações. Se precisar fazer alterações, escolha Previous (Anterior). Caso contrário, selecione Create job (Criar trabalho).

Usar a AWS CLI

O exemplo a seguir cria um trabalho S3PutObjectTagging do S3 Batch Operations usando a AWS CLI.

Como criar uma tarefa **S3PutObjectTagging** de operações em lote

1. Crie uma função do AWS Identity and Access Management (IAM) e atribua permissões. A função concede permissão ao Amazon S3 para adicionar tags de objeto, para as quais você vai criar uma tarefa na próxima etapa.
 - a. Crie uma função do IAM da seguinte forma.

```
aws iam create-role \
--role-name S3BatchJobRole \
--assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "batchoperations.s3.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}'
```

Registre o nome de recurso da Amazon (ARN) da função. Você precisará do ARN 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 do Amazon S3 \(p. 939\)](#).

```
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}",
                "arn:aws:s3:::{ManifestBucket}/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
        "s3:PutObject",
        "s3:GetBucketLocation"
    ],
    "Resource": [
        "arn:aws:s3:::{${ReportBucket}}",
        "arn:aws:s3:::{${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 do 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.

Usar a AWS SDK for Java

O exemplo a seguir cria um trabalho do S3 Batch Operations usando a AWS SDK for Java.

Example

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.*;

import java.util.UUID;
import java.util.ArrayList;

import static com.amazonaws.regions.Regions.US_WEST_2;
```

```
public class CreateJob {  
    public static void main(String[] args) {  
        String accountId = "Account ID";  
        String iamRoleArn = "IAM Role ARN";  
        String reportBucketName = "arn:aws:s3:::bucket-where-completion-report-goes";  
        String uuid = UUID.randomUUID().toString();  
  
        ArrayList<S3Tag> tagSet = new ArrayList<S3Tag>();  
        tagSet.add(new S3Tag().withKey("keyOne").withValue("ValueOne"));  
  
        try {  
            JobOperation jobOperation = new JobOperation()  
                .withS3PutObjectTagging(new S3SetObjectTaggingOperation()  
                    .withTagSet(tagSet)  
                );  
  
            JobManifest manifest = new JobManifest()  
                .withSpec(new JobManifestSpec()  
                    .withFormat("S3BatchOperations_CSV_20180820")  
                    .withFields(new String[]{  
                        "Bucket", "Key"  
                    })  
                )  
                .withLocation(new JobManifestLocation()  
                    .withObjectArn("arn:aws:s3:::my_manifests/manifest.csv")  
                    .withETag("60e460c9d1046e73f7dde5043ac3ae85"));  
            JobReport jobReport = new JobReport()  
                .withBucket(reportBucketName)  
                .withPrefix("reports")  
                .withFormat("Report_CSV_20180820")  
                .withEnabled(true)  
                .withReportScope("AllTasks");  
  
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()  
                .withCredentials(new ProfileCredentialsProvider())  
                .withRegion(US_WEST_2)  
                .build();  
  
            s3ControlClient.createJob(new CreateJobRequest()  
                .withAccountId(accountId)  
                .withOperation(jobOperation)  
                .withManifest(manifest)  
                .withReport(jobReport)  
                .withPriority(42)  
                .withRoleArn(iamRoleArn)  
                .withClientRequestToken(uuid)  
                .withDescription("job description")  
                .withConfirmationRequired(false)  
            );  
  
        } catch (AmazonServiceException e) {  
            // The call was transmitted successfully, but Amazon S3 couldn't process  
            // it and returned an error response.  
            e.printStackTrace();  
        } catch (SdkClientException e) {  
            // Amazon S3 couldn't be contacted for a response, or the client  
            // couldn't parse the response from Amazon S3.  
            e.printStackTrace();  
        }  
    }  
}
```

Uso da API REST

Você pode usar a API REST para criar um trabalho do Batch Operations. Para obter mais informações, consulte [API REST CreateJob](#) na Referência de APIs do Amazon Simple Storage Service.

Respostas do trabalho

Se a solicitação `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 para que você possa identificar as operações em lote e monitorar o status do trabalho.

Quando você cria um trabalho via AWS CLI, Amazon SDKs ou API REST, é possível configurar o S3 Batch Operations para começar a processar o trabalho 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 AWS Management Console, você deve examinar os detalhes e confirmar se deseja executá-lo antes que o Batch Operations comece a processá-lo. Depois que você confirma que deseja executar o trabalho, ele avança como se tivesse sido criado usando um dos outros métodos. Se uma tarefa permanecer no estado suspenso por mais de 30 dias, ela apresentará falha.

Operações suportadas pelo S3 Batch Operations

As operações em lote do S3 são compatíveis com várias operações diferentes. Os tópicos desta seção descrevem cada uma dessas operações.

Copiar objetos

A operação `Copy` (Copiar) copia cada objeto especificado no manifesto. É possível copiar objetos em um bucket na mesma região da AWS ou em um bucket em outra região. O recurso Operações em lote do S3 é compatível com a maioria das opções disponíveis no Amazon S3 para copiar objetos. Essas opções incluem a configuração de metadados de objetos e permissões e a alteração da classe de armazenamento de um objeto.

É possível usar a operação `Copy` para copiar objetos não criptografados existentes e gravá-los de volta no mesmo bucket que os objetos criptografados. Para obter mais informações, consulte [Criptografia de objetos existentes com o Amazon S3 Batch Operations](#).

Ao copiar objetos, você pode alterar o algoritmo de soma de verificação usado para calcular a soma de verificação do objeto. Se os objetos não tiverem uma soma de verificação adicional calculada, você também poderá adicionar uma especificando o algoritmo de soma de verificação a ser usado pelo Amazon S3. Para obter mais informações, consulte [Verificar a integridade do objeto \(p. 229\)](#).

Para obter mais informações sobre como copiar objetos no Amazon S3, bem como parâmetros obrigatórios e opcionais, consulte [Cópia de objetos \(p. 215\)](#) neste guia e `CopyObject` na Referência de APIs do Amazon Simple Storage Service.

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.
- Ao tentar copiar objetos das classes S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive para a classe de armazenamento S3 Standard, primeiro você precisará restaurar esses objetos. Para obter mais informações, consulte [Restaurar um objeto arquivado \(p. 715\)](#).

- Os trabalhos de cópia devem ser criados na região de destino, que é a região para a qual você pretende copiar os objetos.
- Todas as opções de cópia são suportadas, exceto para verificações condicionais em ETags e criptografia do lado do servidor com chaves de criptografia fornecidas pelo cliente (SSE-C).
- Se os buckets não tiverem versões, você deverá substituir os objetos pelos mesmos nomes de chave.
- Os objetos não são necessariamente copiados na mesma ordem em que são listados no manifesto. Para buckets versionados, se preservar a ordem de versão atual/não atual for importante, você deverá copiar todas as versões não atuais primeiro. Em seguida, após a conclusão do primeiro trabalho, copie as versões atuais em um trabalho subsequente.
- A cópia de objetos na classe Reduced Redundancy Storage (RRS) não é compatível.

Copiar objetos usando o S3 Batch Operations

É possível usar o S3 Batch Operations para criar uma tarefa de cópia PUT para copiar objetos dentro da mesma conta ou para uma conta de destino diferente. As seções a seguir contêm exemplos de como armazenar e usar um manifesto que está em uma conta diferente. Na primeira seção, é possível usar o Amazon S3 Inventory para entregar o relatório de inventário à conta de destino para uso durante a criação do trabalho ou usar um manifesto de valores separados por vírgula (CSV) na conta de origem ou de destino, conforme mostrado no segundo exemplo. O terceiro exemplo mostra como usar a operação Copy para ativar a criptografia da chave do bucket do S3 em objetos existentes.

Exemplos de operação de cópia

- [Uso de um relatório de inventário entregue à conta de destino para copiar objetos entre Contas da AWS \(p. 954\)](#)
- [Uso de um manifesto CSV armazenado na conta de origem para copiar objetos entre Contas da AWS \(p. 957\)](#)
- [Uso do S3 Batch Operations para criptografar objetos com chaves de bucket do S3 \(p. 959\)](#)

Uso de um relatório de inventário entregue à conta de destino para copiar objetos entre Contas da AWS

Você pode usar o Amazon S3 Inventory para criar um relatório de inventário e usá-lo para criar uma lista de objetos a serem copiados com o S3 Batch Operations. Para obter mais informações sobre como usar um manifesto CSV na conta de origem ou destino, consulte [the section called “Uso de um manifesto CSV para copiar objetos entre Contas da AWS” \(p. 957\)](#).

O inventário do Amazon S3 gera inventários dos objetos em um bucket. A lista resultante é publicada em um arquivo de saída. O bucket do qual foi feito o inventário é chamado de bucket de origem e o bucket no qual o relatório de inventário é armazenado é chamado de bucket de destino.

O relatório do Amazon S3 Inventory pode ser configurado para ser entregue a outra Conta da AWS. Isso permite que o S3 Batch Operations leia o relatório de inventário quando o trabalho é criado na conta de destino.

Para obter mais informações sobre os buckets de origem e destino do inventário do Amazon S3, consulte [Buckets de origem e destino \(p. 787\)](#).

A maneira mais fácil de configurar um inventário é usando o AWS Management Console, mas você também pode usar a API REST, a AWS Command Line Interface (AWS CLI) ou AWS SDKs.

O procedimento de console a seguir contém as etapas de alto nível para configurar permissões para um trabalho de operações em lote do S3. Neste procedimento, você copia objetos de uma conta de origem para uma conta de destino, com o relatório de inventário armazenado na conta de destino.

Como configurar o Amazon S3 Inventory para buckets de origem e de destino que são de propriedade de diferentes contas

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Escolha um bucket de destino no qual armazenar o relatório de inventário.

Escolha um bucket de manifesto de destino para armazenar o relatório de inventário. Neste procedimento, a conta de destino é a conta que possui tanto o bucket de manifesto de destino como o bucket no qual os objetos são copiados.

3. Configure um inventário para relacionar objetos em um bucket de origem e publicar a lista em um bucket de manifesto de destino.

Configure uma lista de inventário para um bucket de origem. Ao fazer isso, você especifica o bucket de destino no qual deseja que a lista seja armazenada. O relatório de inventário para o bucket de origem é publicado no bucket de destino. Neste procedimento, a conta de origem é a conta que possui o bucket de origem.

Para obter informações sobre como usar o console para configurar um inventário, consulte [Configurar o inventário do Amazon S3 \(p. 789\)](#).

Selecione CSV para o formato de saída.

Ao inserir informações para o bucket de destino, selecione Buckets in another account (Buckets em outra conta). Em seguida, insira o nome do bucket do manifesto de destino. Se preferir, você poderá inserir o ID da conta de destino.

Após salvar a configuração do inventário, o console exibirá uma mensagem semelhante à seguinte:

O Amazon S3 não conseguiu criar uma política de bucket no bucket de destino. Peça para o proprietário do bucket de destino adicionar a política de bucket a seguir a fim de permitir que o Amazon S3 insira dados nele.

O console exibirá, então, uma política de bucket que pode ser usada para o bucket de destino.

4. Copie a política do bucket de destino que aparece no console.
5. Na conta de destino, adicione a política de bucket copiada ao bucket do manifesto de destino no qual o relatório de inventário está armazenado.
6. Crie uma função na conta de destino baseada na política de confiança do recurso Operações em lote do S3. Para obter mais informações sobre a política de confiança, consulte [Política de confiança \(p. 939\)](#).

Para obter mais informações sobre como criar uma função, consulte [Criação de uma função para delegar permissões a um serviço da AWS](#) no Guia do usuário do IAM.

Insira um nome para a função (o exemplo usa o nome BatchOperationsDestinationRoleCOPY). Escolha o serviço S3 e selecione o caso de uso S3 bucket Batch Operations (Operações em lote do bucket do S3), que aplica a política de confiança à função.

Em seguida, selecione Create policy (Criar política) para anexar a política a seguir à função.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowBatchOperationsDestinationObjectCOPY",  
            "Effect": "Allow",  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::  
                BatchOperationsDestinationBucketName/*"  
        }  
    ]  
}
```

```

    "Action": [
        "s3:PutObject",
        "s3:PutObjectVersionAcl",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionTagging",
        "s3:PutObjectTagging",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
    ],
    "Resource": [
        "arn:aws:s3:::ObjectDestinationBucket/*",
        "arn:aws:s3:::ObjectSourceBucket/*",
        "arn:aws:s3:::ObjectDestinationManifestBucket/*"
    ]
}
]
}
}

```

A função usa a política para conceder ao `batchoperations.s3.amazonaws.com` permissão para ler o manifesto no bucket de destino. Ela também concede permissões a objetos GET, listas de controle de acesso (ACLs), tags e versões no bucket do objeto de origem. Além disso, ela concede permissões a objetos PUT, ACLs, tags e versões no bucket do objeto de destino.

7. Na conta de origem, crie uma política de bucket para o bucket de origem que concede a função que você criou na etapa anterior para objetos GET, ACLs, tags e versões no bucket de origem. Esta etapa permite que as operações em lote do S3 obtenham objetos do bucket de origem por meio da função de confiança.

Veja a seguir um exemplo da política de bucket para a conta de origem.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowBatchOperationsSourceObjectCOPY",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::DestinationAccountNumber:role/BatchOperationsDestinationRoleCOPY"
            },
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:GetObjectAcl",
                "s3:GetObjectTagging",
                "s3:GetObjectVersionAcl",
                "s3:GetObjectVersionTagging"
            ],
            "Resource": "arn:aws:s3:::ObjectSourceBucket/*"
        }
    ]
}

```

8. Depois que o relatório de inventário estiver disponível, crie uma tarefa de cópia do objeto PUT de operações em lote do S3 na conta de destino escolhendo o relatório de inventário no bucket do manifesto de destino. Você precisa do ARN para a função que criou na conta de destino.

Para obter informações gerais sobre como criar uma tarefa, consulte [Criar um trabalho de operações em lote do S3 \(p. 946\)](#).

Para obter informações sobre a criação de um trabalho usando o console, consulte [Criar um trabalho de operações em lote do S3 \(p. 946\)](#).

Uso de um manifesto CSV armazenado na conta de origem para copiar objetos entre Contas da AWS

Você pode usar um arquivo CSV que está armazenado em outra Conta da AWS como um manifesto para um trabalho do S3 Batch Operations. Para usar um relatório de inventário do S3, consulte [the section called “Uso de um relatório de inventário para copiar objetos entre Contas da AWS” \(p. 954\)](#).

O procedimento a seguir mostra como configurar permissões ao usar uma tarefa de operações em lote do S3 para copiar objetos de uma conta de origem para uma conta de destino com arquivo de manifesto CSV armazenado na conta de origem.

Para configurar um manifesto CSV armazenado em outra Conta da AWS

1. Crie uma função na conta de destino baseada na política de confiança do recurso Operações em lote do S3. Neste procedimento, a conta de destino é a conta para a qual os objetos estão sendo copiados.

Para obter mais informações sobre a política de confiança, consulte [Política de confiança \(p. 939\)](#).

Para obter mais informações sobre como criar uma função, consulte [Criação de uma função para delegar permissões a um serviço da AWS](#) no Guia do usuário do IAM.

Se você criar a função usando o console, insira um nome para a função (a função do exemplo usa o nome `BatchOperationsDestinationRoleCOPY`). Escolha o serviço S3 e selecione o caso de uso S3 bucket Batch Operations (Operações em lote do bucket do S3), que aplica a política de confiança à função.

Em seguida, selecione Create policy (Criar política) para anexar a política a seguir à função.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowBatchOperationsDestinationObjectCOPY",  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject",  
                "s3:PutObjectVersionAcl",  
                "s3:PutObjectAcl",  
                "s3:PutObjectVersionTagging",  
                "s3:PutObjectTagging",  
                "s3:GetObject",  
                "s3:GetObjectVersion",  
                "s3:GetObjectAcl",  
                "s3:GetObjectTagging",  
                "s3:GetObjectVersionAcl",  
                "s3:GetObjectVersionTagging"  
            ],  
            "Resource": [  
                "arn:aws:s3:::ObjectDestinationBucket/*",  
                "arn:aws:s3:::ObjectSourceBucket/*",  
                "arn:aws:s3:::ObjectSourceManifestBucket/*"  
            ]  
        }  
    ]  
}
```

Com o uso da política, a função concede permissão ao `batchoperations.s3.amazonaws.com` para ler o manifesto no bucket do manifesto de origem. Ela concede permissões a objetos GET, ACLs, tags e versões no bucket do objeto de origem. Ela também concede permissões a objetos PUT, ACLs, tags e versões no bucket do objeto de destino.

2. Na conta de origem, crie uma política de bucket para o bucket que contém o manifesto para conceder a função que você criou na etapa anterior para objetos GET e versões no bucket do manifesto de origem.

Esta etapa permite que as operações em lote do S3 leiam o manifesto usando a função confiável. Aplique a política de bucket ao bucket que contém o manifesto.

Veja a seguir um exemplo da política de bucket para ser aplicada ao bucket do manifesto de origem.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowBatchOperationsSourceManifestRead",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": [  
                    "arn:aws:iam::DestinationAccountNumber:user/ConsoleUserCreatingJob",  
                    "arn:aws:iam::DestinationAccountNumber:role/  
BatchOperationsDestinationRoleCOPY"  
                ]  
            },  
            "Action": [  
                "s3:GetObject",  
                "s3:GetObjectVersion"  
            ],  
            "Resource": "arn:aws:s3:::ObjectSourceManifestBucket/*"  
        }  
    ]  
}
```

Esta política também concede ao usuário de um console que está criando uma tarefa na conta de destino as mesmas permissões no bucket do manifesto de origem por meio da mesma política de bucket.

3. Na conta de origem, crie uma política de bucket para o bucket de origem que concede a função que você criou para objetos GET, ACLs, tags e versões no bucket de origem. O Operações em lote do S3 pode então obter objetos do bucket de origem por meio da função confiável.

Veja a seguir um exemplo da política de bucket para o bucket que contém os objetos de origem.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowBatchOperationsSourceObjectCOPY",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::DestinationAccountNumber:role/  
BatchOperationsDestinationRoleCOPY"  
            },  
            "Action": [  
                "s3:GetObject",  
                "s3:GetObjectVersion",  
                "s3:GetObjectAcl",  
                "s3:GetObjectTagging",  
                "s3:GetObjectVersionAcl",  
                "s3:PutObjectAcl"  
            ],  
            "Resource": "arn:aws:s3:::ObjectSourceManifestBucket/*"  
        }  
    ]  
}
```

```
        "s3:GetObjectVersionTagging"
    ],
    "Resource": "arn:aws:s3:::ObjectSourceBucket/*"
}
]
```

- Crie um trabalho de operações em lote do S3 na conta de destino. Você precisa do nome de recurso da Amazon (ARN) para a função que criou na conta de destino.

Para obter informações gerais sobre como criar uma tarefa, consulte [Criar um trabalho de operações em lote do S3 \(p. 946\)](#).

Para obter informações sobre a criação de um trabalho usando o console, consulte [Criar um trabalho de operações em lote do S3 \(p. 946\)](#).

Uso do S3 Batch Operations para criptografar objetos com chaves de bucket do S3

Nesta seção, você usa a operação Copy do Amazon S3 Batch Operations para identificar e ativar a criptografia de chaves de bucket do S3 em objetos existentes. Para obter mais informações sobre chaves de buckets do S3, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3 \(p. 369\)](#) e [Configurando seu bucket para usar uma chave de bucket do S3 com SSE-KMS para novos objetos \(p. 372\)](#).

Os tópicos abordados neste exemplo incluem:

Tópicos

- [Pré-requisitos \(p. 959\)](#)
- [Etapa 1: obter a lista de objetos usando o Amazon S3 Inventory \(p. 959\)](#)
- [Etapa 2: Filtrar sua lista de objetos com S3 Select \(p. 960\)](#)
- [Etapa 3: Configurar e executar o trabalho do S3 Batch Operations \(p. 962\)](#)
- [Resumo \(p. 966\)](#)

Pré-requisitos

Para acompanhar as etapas deste procedimento, é necessário ter uma conta da AWS e pelo menos um bucket do S3 para manter os arquivos de trabalho e os resultados criptografados. Você também pode achar grande parte da documentação existente do S3 Batch Operations útil, incluindo os seguintes tópicos:

- [Conceitos básicos sobre operações em lote do S3 \(p. 938\)](#)
- [Criar um trabalho de operações em lote do S3 \(p. 946\)](#)
- [Operações suportadas pelo S3 Batch Operations \(p. 953\)](#)
- [Gerenciar trabalhos de operações em lote do S3 \(p. 979\)](#)

Etapa 1: obter a lista de objetos usando o Amazon S3 Inventory

Para começar, identifique o bucket do S3 que contém os objetos a serem criptografados e obtenha uma lista de seu conteúdo. Um relatório do Amazon S3 Inventory é a maneira mais conveniente e acessível de fazer isso. O relatório fornece a lista dos objetos em um bucket, juntamente com os metadados associados. O bucket de origem refere-se ao bucket inventariado, e o bucket de destino refere-se ao bucket onde você armazena o arquivo de relatório de inventário. Para obter mais informações sobre os buckets de origem e destino do inventário do Amazon S3, consulte [Inventário do Amazon S3 \(p. 786\)](#).

A maneira mais fácil de configurar um inventário é usando o AWS Management Console. Mas você também pode usar a API REST, AWS Command Line Interface (AWS CLI) ou AWS SDKs. Antes de seguir

essas etapas, faça login no console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>. Se você encontrar erros de permissão negada, adicione uma política de bucket ao bucket de destino. Para obter mais informações, consulte [Conceder permissões para o Amazon S3 Inventory e a análise do Simple Storage Service \(Amazon S3\) \(p. 526\)](#).

Para obter uma lista de objetos usando o S3 Inventory

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Buckets e, em seguida, escolha um bucket que contém objetos para criptografar.
3. Na guia Management (Gerenciamento), navegue para a seção Inventory configurations (Configurações de inventário) e escolha Create inventory configuration (Criar configuração de inventário).
4. Dê um nome ao seu novo inventário, insira o nome do bucket S3 de destino e, opcionalmente, crie um prefixo de destino para que o Amazon S3 atribua objetos nesse bucket.
5. Em Output format (Formato de saída), escolha CSV.
6. Na seção Additional fields - optional (Campos adicionais - opcional), escolha Encryption (Criptografia) e quaisquer outros campos de relatórios que interessem a você. Defina a frequência das entregas de relatório como Daily (Diariamente) para que o primeiro relatório seja entregue ao seu bucket mais cedo.
7. Selecione Create (Criar) para salvar suas configurações.

O Amazon S3 pode demorar até 48 horas para entregar o primeiro relatório, portanto, verifique quando o primeiro relatório chegar. Após receber seu primeiro relatório, vá para a próxima seção para filtrar o conteúdo do relatório do S3 Inventory. Se não desejar mais receber relatórios de inventário para esse bucket, exclua a configuração do S3 Inventory. Caso contrário, o S3 fornecerá relatórios em uma programação diária ou semanal.

Uma lista de inventário não é uma única visualização point-in-time de todos os objetos. As listas de inventário são um snapshot de contínuo de itens do bucket que são consistentes no final (ou seja, a lista pode não incluir os objetos adicionados ou excluídos recentemente). A combinação do S3 Inventory e S3 Batch Operations funciona melhor quando você trabalha com objetos estáticos ou com um conjunto de objetos criado há dois ou mais dias. Para trabalhar com dados mais recentes, use a operação da API [ListObjectSv2](#) (GET Bucket) para criar sua lista de objetos manualmente. Se necessário, repita o processo nos próximos dias ou até que o relatório de inventário mostre o status desejado para todas as chaves.

Etapa 2: Filtrar sua lista de objetos com S3 Select

Depois de receber o relatório do S3 Inventory, você poderá filtrar o conteúdo do relatório para listar somente os objetos que não estão criptografados com chaves de bucket. Se quiser que todos os objetos do bucket sejam criptografados com chaves de bucket, você poderá ignorar esta etapa. No entanto, filtrar seu relatório do S3 Inventory nesta fase economiza o tempo e as despesas de criptografar novamente os objetos que já foram criptografados.

Embora as etapas a seguir mostrem como filtrar usando o [Amazon S3 Select](#), você também pode usar o [Amazon Athena](#). Para decidir qual ferramenta usar, consulte o arquivo `manifest.json` de relatório do S3 Inventory. Esse arquivo lista o número de arquivos de dados associados a esse relatório. Se o número for grande, use o Amazon Athena porque ele é executado em vários objetos do S3, enquanto o S3 Select funciona em um objeto de cada vez. Para obter mais informações sobre como usar o Amazon S3 e o Athena em conjunto, consulte [Consulta do Amazon S3 Inventory com o Amazon Athena \(p. 797\)](#) e [Uso do Athena](#) na publicação do blog [Encrypting objects with Amazon S3 Batch Operations](#).

Para filtrar seu relatório do S3 Inventory usando o S3 Select

1. Abra o arquivo `manifest.json` do seu relatório de inventário e observe a seção `fileSchema` do JSON. Isso informa a consulta que você executa nos dados.

O seguinte JSON é um arquivo `manifest.json` de exemplo para um inventário em formato CSV em um bucket com versionamento habilitado. Dependendo de como você configurou seu relatório de inventário, seu manifesto pode parecer diferente.

```
{  
    "sourceBucket": "batchoperationsdemo",  
    "destinationBucket": "arn:aws:s3:::testbucket",  
    "version": "2021-05-22",  
    "creationTimestamp": "1558656000000",  
    "fileFormat": "CSV",  
    "fileSchema": "Bucket, Key, VersionId, IsLatest, IsDeleteMarker, BucketKeyStatus",  
    "files": [  
        {  
            "key": "demoinv/batchoperationsdemo/DemoInventory/data/009a40e4-f053-4c16-8c75-6100f8892202.csv.gz",  
            "size": 72691,  
            "MD5checksum": "c24c831717a099f0ebe4a9d1c5d3935c"  
        }  
    ]  
}
```

Se o versionamento não estiver ativado no bucket, ou se você optar por executar o relatório para as versões mais recentes, o `fileSchema` é `Bucket, Key e BucketKeyStatus`.

Se o versionamento estiver habilitado, dependendo de como você configurou o relatório de inventário, o `fileSchema` poderá incluir o seguinte: `Bucket, Key, VersionId, IsLatest, IsDeleteMarker, BucketKeyStatus`. Portanto, preste atenção às colunas 1, 2, 3 e 6 ao executar a consulta.

O S3 Batch Operations precisa do bucket, da chave e do ID da versão como entradas para executar o trabalho, além do campo pelo qual pesquisar, que é o status da chave do bucket. Você não precisa do campo de ID da versão, mas especificá-lo quando você opera em um bucket versionado é útil. Para obter mais informações, consulte [Trabalhar com objetos em um bucket com versionamento habilitado \(p. 689\)](#).

2. Localize os arquivos de dados para o relatório de inventário. O objeto `manifest.json` lista os arquivos de dados em `files` (arquivos).
3. Depois de localizar e selecionar o arquivo de dados no console do S3, escolha Actions (Ações) e, em seguida, Query with S3 Select (Consulta com o S3 Select).
4. Mantenha os campos predefinidos CSV, Comma (Vírgula) e GZIP selecionados e escolha Next (Avançar).
5. Para revisar o formato do relatório de inventário antes de prosseguir, escolha Show file preview (Mostrar visualização do arquivo).
6. Insira as colunas a serem referenciadas no campo de expressão SQL. Depois, escolha Run SQL (Executar SQL). A expressão a seguir retorna as colunas 1 – 3 para todos os objetos sem chave de bucket configurada.

```
select s._1, s._2, s._3 from s3object s where s._6 = 'DISABLED'
```

Estes são resultados de exemplo.

```
batchoperationsdemo,0100059%7Ethumb.jpg,lsrtIxksLu0R0ZkYPL.LhgD5caTYn6vu  
batchoperationsdemo,0100074%7Ethumb.jpg,sd2M60g6Fdazoi6D5kNARIE7KzUibmHR  
batchoperationsdemo,0100075%7Ethumb.jpg,TLYESLn1mXD5c4BwiOIinqFrktddkol  
batchoperationsdemo,0200147%7Ethumb.jpg,amufzfMi_fEw0Rs99rxR_HrDF1E.13Y0  
batchoperationsdemo,0301420%7Ethumb.jpg,9qGU2SEscL.C.c_SK89trmXYIwooABSh  
batchoperationsdemo,0401524%7Ethumb.jpg,ORnEWNuB1QhHrrYAGFsZhbyvEYJ3DUor  
batchoperationsdemo,200907200065HQ%7Ethumb.jpg,d8LgvIVjbDR5mUVwW6pu9ahTfReyn5V4  
batchoperationsdemo,200907200076HQ%7Ethumb.jpg,XUT25d7.gK40u_GmnupdaZg3BVx2jN40
```

batchoperationsdemo, 201103190002HQ%7Ethumb.jpg, z.2sVRh0myqVi0BuIrngWlsRPQdb7qOS

7. Faça download dos resultados, salve-os em um formato CSV e faça seu upload para o Amazon S3 como sua lista de objetos para o trabalho do S3 Batch Operations.
8. Se você tiver vários arquivos de manifesto, execute Query with S3 Select (Consulta com o S3 Select) neles também. Dependendo do tamanho dos resultados, você pode combinar as listas e executar um único trabalho do S3 Batch Operations ou executar cada lista como um trabalho separado.

Considere o [preço](#) de executar cada trabalho do S3 Batch Operations ao decidir o número de trabalhos a serem executados.

Etapa 3: Configurar e executar o trabalho do S3 Batch Operations

Agora que tem suas listas CSV filtradas de objetos do S3, você pode iniciar o trabalho de do S3 Batch Operations para criptografar os objetos com chaves de bucket do S3.

Um trabalho refere-se coletivamente à lista (manifesto) de objetos fornecidos, à operação executada e aos parâmetros especificados. A maneira mais fácil de criptografar esse conjunto de objetos é usando a operação de cópia `PUT` e especificando o mesmo prefixo de destino dos objetos listados no manifesto. Isso substitui os objetos existentes em um bucket sem versionamento ou, com o versionamento ativado, cria uma versão mais recente e criptografada dos objetos.

Como parte da cópia dos objetos, especifique que o Amazon S3 deve criptografar o objeto com criptografia SSE-KMS e S3. Esse trabalho copia os objetos para que todos os seus objetos mostrem uma data de criação atualizada após a conclusão, independentemente de quando você os adicionou originalmente ao S3. Especifique também as outras propriedades do seu conjunto de objetos como parte do trabalho do S3 Batch Operations, incluindo tags de objeto e classe de armazenamento.

Subetapas

- [Definir sua política do IAM \(p. 962\)](#)
- [Configurar sua função do IAM do Batch Operations \(p. 963\)](#)
- [Ative as chaves de bucket do S3 para um bucket existente \(p. 964\)](#)
- [Criar seu trabalho do Batch Operations \(p. 964\)](#)
- [Executar seu trabalho do Batch Operations \(p. 965\)](#)
- [Pontos importantes \(p. 965\)](#)

Definir sua política do IAM

1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, escolha Policy (Política) e, em seguida, Create Policy (Criar política).
3. Selecione a guia JSON. SelecioneEdit policy (Editar política) e adicione a política do IAM de exemplo que aparece no bloco de código a seguir.

Depois de copiar o exemplo de política no console do IAM, substitua o seguinte:

- a. Substitua `{SOURCE_BUCKET_FOR_COPY}` pelo nome do bucket de origem.
- b. Substitua `{DESTINATION_BUCKET_FOR_COPY}` pelo nome do bucket de destino.
- c. Substitua `{MANIFEST_KEY}` pelo nome do objeto de manifesto.
- d. Substitua `{REPORT_BUCKET}` pelo nome do bucket onde deseja salvar os relatórios.

```
{  
    "Version": "2012-10-17",
```

```
"Statement": [
    {
        "Sid": "CopyObjectsToEncrypt",
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:PutObjectTagging",
            "s3:PutObjectAcl",
            "s3:PutObjectVersionTagging",
            "s3:PutObjectVersionAcl",
            "s3:GetObject",
            "s3:GetObjectAcl",
            "s3:GetObjectTagging",
            "s3:GetObjectVersion",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
        ],
        "Resource": [
            "arn:aws:s3:::{SOURCE_BUCKET_FOR_COPY}/*",
            "arn:aws:s3:::{DESTINATION_BUCKET_FOR_COPY}/*"
        ]
    },
    {
        "Sid": "ReadManifest",
        "Effect": "Allow",
        "Action": [
            "s3:GetObject",
            "s3:GetObjectVersion"
        ],
        "Resource": "arn:aws:s3:::{MANIFEST_KEY}"
    },
    {
        "Sid": "WriteReport",
        "Effect": "Allow",
        "Action": [
            "s3:PutObject"
        ],
        "Resource": "arn:aws:s3:::{REPORT_BUCKET}/*"
    }
]
```

4. Escolha Next: Tags (Próximo: tags).
5. Adicione quaisquer tags desejadas (opcional) e escolha Next: Review (Próximo: Revisão).
6. Adicione um nome para a política e, opcionalmente, uma descrição. Em seguida, escolha Create policy (Criar política).
7. Selecione Review policy (Revisar política) e, em seguida, escolha, Save changes (Salvar alterações).
8. Com sua política do S3 Batch Operations agora concluída, o console retorna você à página Policies (Políticas) do IAM. Filtre o nome da política, escolha o botão à esquerda do nome da política, escolha o botão Policy actions (Ações da política) e escolha Attach (Anexar).

Para anexar a política recém-criada a uma função do IAM, selecione os usuários, grupos ou funções apropriados em sua conta e escolha Attach policy (Anexar política). Isso levará você de volta ao console do IAM.

Configurar sua função do IAM do Batch Operations

1. No painel de navegação do console do IAM, selecione Roles (Funções) e, em seguida, Create role (Criar função).
2. Escolha AWS service (Serviço da AWS), S3 e S3 Batch Operations (Operações em lote do S3). Então, escolha Próximo: Permissões.

3. Comece a inserir o nome da política do IAM que acabou de criar. Marque a caixa de seleção ao lado do nome da política quando ela for exibida e escolha Next: Tags (Próximo: Tags).
4. (Opcional) Adicione tags ou mantenha os campos de chave e valor em branco para este exercício. Selecione Next: Review.
5. Insira um nome de função e aceite a descrição padrão ou adicione a sua própria. Selecione Create role.
6. Certifique-se de que o usuário que está criando o trabalho tenha as permissões no exemplo a seguir.

Substitua `{ACCOUNT-ID}` por seu ID da conta da AWS e `{IAM_ROLE_NAME}` pelo nome que você planeja aplicar à função do IAM que criará na etapa de criação do trabalho do Batch Operations posteriormente. Para obter mais informações, consulte [Conceder permissões para operações em lote do Amazon S3 \(p. 939\)](#).

```
{  
    "Sid": "AddIamPermissions",  
    "Effect": "Allow",  
    "Action": [  
        "iam:GetRole",  
        "iam:PassRole"  
    ],  
    "Resource": "arn:aws:iam::{ACCOUNT-ID}:role/{IAM_ROLE_NAME}"  
}
```

Ative as chaves de bucket do S3 para um bucket existente

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o bucket para o qual você deseja ativar chaves de bucket do S3.
3. Escolha Properties (Propriedades).
4. Em Default encryption (Criptografia padrão), escolha Edit (Editar).
5. Nas opções de Server-side encryption (Criptografia no lado do servidor), escolha Enable (Habilitar).
6. Em Encryption key type (Tipo da chave de criptografia), escolha AWS KMS key (SSE-KMS) (Chave do KMS (SSE-KMS)) e escolha o formato da chave do AWS KMS preferido:
 - AWS chave gerenciada (aws/s3).
 - Escolha entre suas chaves do AWS KMS e uma chave do KMS de criptografia simétrica na mesma região de seu bucket.
 - ARN da chave do AWS KMS
7. Em Bucket key (Chave do bucket), escolha Enable (Habilitar) e, em seguida, escolha Save changes (Salvar alterações).

Agora que a chave do bucket está ativada no nível do bucket, os objetos carregados, modificados ou copiados nesse bucket herdarão essa configuração de criptografia por padrão. Isso inclui objetos copiados usando o Amazon S3 Batch Operations.

Criar seu trabalho do Batch Operations

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha S3 Batch Operations e, em seguida, escolha Create Job (Criar trabalho).
3. Selecione a Region (Região) onde você armazena seus objetos e escolha CSV como o tipo de manifesto.
4. Insira o caminho ou navegue para o arquivo de manifesto CSV criado anteriormente a partir dos resultados do S3 Select (ou Athena). Se o manifesto contiver IDs de versão, marque essa caixa. Escolha Next (Próximo).

5. Selecione a operação Copy (Copiar) e escolha o bucket de destino da cópia. Você pode manter a criptografia no lado do servidor desabilitada. Desde que o destino do bucket tenha a chave do bucket habilitada, a operação de cópia aplicará a chave do bucket no bucket de destino.
6. (Opcional) Escolha uma classe de armazenamento e os outros parâmetros conforme desejado. Os parâmetros especificados nesta etapa se aplicam a todas as operações executadas nos objetos listados no manifesto. Escolha Next (Próximo).
7. Dê uma descrição ao seu trabalho (ou mantenha o padrão), defina seu nível de prioridade, escolha um tipo de relatório e especifique o Path to completion report destination (Caminho para destino do relatório de conclusão).
8. Na seção Permissions (Permissões), certifique-se de escolher a função do IAM do Batch Operations definida anteriormente. Escolha Next (Próximo).
9. Em Review (Revisão), verifique as configurações. Se precisar fazer alterações, escolha Previous (Anterior). Depois de confirmar as configurações do Batch Operations, escolha Create job (Criar trabalho).

Para obter mais informações, consulte [Criar um trabalho de operações em lote do S3 \(p. 946\)](#).

Executar seu trabalho do Batch Operations

O assistente de configuração retorna você automaticamente à seção S3 Batch Operations do console do Amazon S3. Suas novas transições de trabalho do estado New (Novo) para o estado Preparing (Preparação) como S3 inicia o processo. Durante o estado de preparação, o S3 lê o manifesto do trabalho, verifica se há erros e calcula o número de objetos.

1. Escolha o botão de atualização no console do Amazon S3 para verificar o progresso. Dependendo do tamanho do manifesto, a leitura pode levar minutos ou horas.
2. Quando o S3 termina de ler o manifesto do trabalho, o trabalho é movido para o estado Awaiting your confirmation (Aguardando sua confirmação). Escolha o botão de opção à esquerda da ID do trabalho e escolha Run job (Executar trabalho).
3. Verifique as configurações do trabalho e escolha Run job (Executar trabalho) no canto inferior direito.

Depois que o trabalho começar a ser executado, você poderá escolher o botão de atualizar para verificar o andamento na exibição do painel do console ou selecionando o trabalho específico.

4. Quando o trabalho é concluído, você pode exibir as contagens de objetos Successful (Com êxito) e Failed (Com falha) para confirmar que tudo foi executado conforme esperado. Se você ativou relatórios de trabalhos, verifique o relatório do trabalho para saber a causa exata de qualquer operação com falha.

Você também pode executar essas etapas usando a AWS CLI, SDKs ou APIs. Para obter mais informações sobre como rastrear o status de um trabalho e os relatórios de conclusão, consulte [Rastreamento de relatórios de status e conclusão \(p. 982\)](#).

Pontos importantes

Considere os seguintes problemas ao usar o S3 Batch Operations para criptografar objetos com chaves de bucket:

- Você será cobrado por trabalhos, objetos e solicitações do S3 Batch Operations, além de quaisquer encargos associados à operação que o S3 Batch Operations executa em seu nome, incluindo transferência de dados, solicitações e outras cobranças. Para obter mais informações, consulte [Definição de preço do Amazon S3](#).
- Se você usar um bucket versionado, cada trabalho do S3 Batch Operations executado criará novas versões criptografadas de seus objetos. Ele também mantém as versões anteriores sem chave de bucket configurada. Para excluir as versões antigas, configure uma política de expiração do ciclo de

vida do S3 para versões não atuais, conforme descrito em [Elementos de configuração do ciclo de vida \(p. 767\)](#).

- A operação de cópia cria novos objetos com novas datas de criação, o que pode afetar ações do ciclo de vida, como arquivamento. Se você copiar todos os objetos no bucket, todas as novas cópias terão datas de criação idênticas ou semelhantes. Para identificar esses objetos e criar regras de ciclo de vida diferentes para vários subconjuntos de dados, considere o uso de tags de objeto.

Resumo

Nesta seção, você classificou objetos existentes para filtrar dados já criptografados. Em seguida, você aplicou o recurso de chave de bucket em objetos não criptografados usando o S3 Batch Operations para copiar dados existentes em um bucket com chave de bucket ativada. Esse processo pode economizar tempo e dinheiro, permitindo que você conclua operações como criptografar todos os objetos existentes.

Para obter mais informações sobre o S3 Batch Operations, consulte [Executar operações em lote de grande escala em objetos do Amazon S3 \(p. 938\)](#).

Para obter exemplos que mostram a operação de cópia com tags usando a AWS CLI e o AWS SDK for Java, consulte [Criar um trabalho do Operações em lote com tags de trabalho usadas para rotulagem \(p. 993\)](#).

Invocar função do AWS Lambda

A função Invoke AWS Lambda inicia as funções do AWS Lambda para executar ações personalizadas em objetos que estão listados em um manifesto. Esta seção descreve como criar uma função do Lambda para uso com as operações em lote do S3 e como criar um trabalho para chamar a função. O trabalho de Operações em lote do S3 usa a operação `LambdaInvoke` para executar uma função do Lambda em cada objeto listado em um manifesto.

Você pode trabalhar com o S3 Batch Operations para Lambda via AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDKs ou APIs REST. Para obter mais informações sobre como usar o Lambda, consulte [Conceitos básicos do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

As seções a seguir explicam como começar a usar as operações em lote do S3 com o Lambda.

Tópicos

- [Usar o Lambda com operações em lote do Amazon S3 \(p. 966\)](#)
- [Criar uma função do Lambda para uso com operações em lote do S3 \(p. 967\)](#)
- [Criar um trabalho de operações em lote do S3 que chama uma função do Lambda \(p. 971\)](#)
- [Fornecer informações em nível de tarefa em manifestos do Lambda \(p. 971\)](#)

Usar o Lambda com operações em lote do Amazon S3

Ao usar o S3 Batch Operations com o AWS Lambda, crie novas funções do Lambda especificamente para uso com o S3 Batch Operations. Não é possível reutilizar funções baseadas em eventos do Amazon S3 existentes com operações em lote do S3. As funções de eventos só podem receber mensagens; elas não retornam mensagens. As funções do Lambda usadas com operações em lote do S3 devem aceitar e retornar mensagens. Para obter mais informações sobre como usar o Lambda com eventos do Amazon S3, consulte [Uso do AWS Lambda com o Amazon S3](#) no Guia do desenvolvedor do AWS Lambda.

Crie um trabalho de operações em lote do S3 que chame a função do Lambda. O trabalho executa a mesma função do Lambda em todos os objetos listados no manifesto. É possível controlar quais versões da função do Lambda usar durante o processamento dos objetos no manifesto. As operações em lote

do S3 são compatíveis com nomes de recursos da Amazon (ARNs) não qualificados, aliases e versões específicas. Para obter mais informações, consulte [Introdução ao versionamento do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

Se você fornecer ao trabalho de operações em lote do S3 um ARN de função que usa um alias ou o qualificador `$LATEST`, e atualizar a versão indicada por um deles, as operações em lote do S3 começarão a chamar a nova versão dessa função do Lambda. Isso pode ser útil quando você quer atualizar a funcionalidade durante um trabalho grande. Para que as operações em lote do S3 não alterem a versão usada, forneça a versão específica no parâmetro `FunctionARN` ao criar o trabalho.

Códigos de resultado e resposta

Há dois níveis de código que as operações em lote do S3 esperam das funções do Lambda. O primeiro é o código de resposta para toda a solicitação, e o segundo é um código de resultado por tarefa. A tabela a seguir contém os códigos de resposta.

Código de resposta	Descrição
<code>Succeeded</code>	A tarefa foi concluída normalmente. Se você solicitou um relatório de conclusão do trabalho, a string do resultado da tarefa será incluída no relatório.
<code>TemporaryFailure</code>	A tarefa sofreu uma falha temporária e será redirecionada antes da conclusão do trabalho. A string de resultado é ignorada. Se esse for o redirecionamento final, a mensagem de erro será incluída no relatório final.
<code>PermanentFailure</code>	A tarefa sofreu uma falha permanente. Se você solicitou um relatório de conclusão do trabalho, a tarefa será marcada como <code>Failed</code> e incluirá a string da mensagem de erro. As strings de resultado de tarefas com falha são ignoradas.

Criar uma função do Lambda para uso com operações em lote do S3

Essa seção fornece um exemplo de permissões do AWS Identity and Access Management (IAM) que devem ser usadas com a função do Lambda. Ela também contém um exemplo de função do Lambda para uso com operações em lote do S3. Se você nunca criou uma função do Lambda antes, consulte [Tutorial: Como usar um trigger do Amazon S3 para invocar uma função do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

Crie funções do Lambda especificamente para uso com as operações em lote do S3. Não é possível reutilizar as funções existentes do Lambda baseadas em eventos do Amazon S3. Isso ocorre porque as funções do Lambda usadas para operações em lote do S3 devem aceitar e retornar campos de dados especiais.

Important

As funções do AWS Lambda escritas em Java aceitam as interfaces de manipulador `RequestHandler` ou `RequestStreamHandler`. No entanto, para oferecer suporte ao formato de solicitação e resposta do S3 Batch Operations, o AWS Lambda requer a interface `RequestStreamHandler` para serialização e desserialização personalizadas de uma solicitação e resposta. Essa interface permite que o Lambda passe um `InputStream` e um `OutputStream` para o método Java `handleRequest`.

Use a interface `RequestStreamHandler` ao usar funções do Lambda com operações em lote do S3. Se você usar uma interface `RequestHandler`, o trabalho em lote falhará com “Invalid JSON returned in Lambda payload (JSON inválido retornado no payload do Lambda)” no relatório de conclusão.

Para obter mais informações, consulte [Interfaces do manipulador](#) no Guia do usuário do AWS Lambda.

Exemplo de permissões do IAM

Veja a seguir exemplos das permissões do IAM necessárias para usar uma função do Lambda com operações em lote do S3.

Example Política de confiança de operações em lote do S3

Veja a seguir um exemplo da política de confiança que pode ser usada para a função do IAM de operações em lote. Essa função do IAM é especificada quando você cria o trabalho e concede às operações em lote permissão para assumir a função do IAM.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "batchoperations.s3.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

Example Política do IAM do Lambda

Veja a seguir um exemplo de uma política do IAM que concede permissão às operações em lote do S3 para chamar a função do Lambda e ler o manifesto de entrada.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "BatchOperationsLambdaPolicy",  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetObject",  
                "s3:GetObjectVersion",  
                "s3:PutObject",  
                "lambda:InvokeFunction"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Exemplo de solicitação e resposta

Esta seção fornece exemplos de solicitação e resposta para a função do Lambda.

Example Request

Veja a seguir um exemplo do JSON de uma solicitação para a função do Lambda.

```
{  
    "invocationSchemaVersion": "1.0",  
    "invocationId": "YXNkbGZqYWRmaiBhc2RmdW9hZHNmZGpmaGFzbGtkaGZza2RmaAo",  
    "job": {  
        "id": "f3cc4f60-61f6-4a2b-8a21-d07600c373ce"  
    },  
    "tasks": [  
        {  
            "taskId": "dGFza2lkZ29lc2hlcUK",  
            "s3Key": "customerImage1.jpg",  
            "s3VersionId": "1",  
            "s3BucketArn": "arn:aws:s3:us-east-1:0123456788:awsexamplebucket1"  
        }  
    ]  
}
```

Example Response

Veja a seguir um exemplo do JSON de uma resposta para a função do Lambda.

```
{  
    "invocationSchemaVersion": "1.0",  
    "treatMissingKeysAs" : "PermanentFailure",  
    "invocationId" : "YXNkbGZqYWRmaiBhc2RmdW9hZHNmZGpmaGFzbGtkaGZza2RmaAo",  
    "results": [  
        {  
            "taskId": "dGFza2lkZ29lc2hlcUK",  
            "resultCode": "Succeeded",  
            "resultString": "[\"Mary Major\", \"John Stiles\"]"  
        }  
    ]  
}
```

Exemplo de função do Lambda para operações em lote do S3

O exemplo Lambda em Python a seguir remove um marcador de exclusão de um objeto versionado.

Como mostra o exemplo, as chaves das operações em lote do S3 são codificadas por URL. Para usar o Amazon S3 com outros produtos da AWS, é importante que seu URL decodifique a chave passada pelo S3 Batch Operations.

```
import logging  
from urllib import parse  
import boto3  
from botocore.exceptions import ClientError  
  
logger = logging.getLogger(__name__)  
logger.setLevel('INFO')  
  
s3 = boto3.client('s3')  
  
def lambda_handler(event, context):  
    """  
    Removes a delete marker from the specified versioned object.  
    :param event: The S3 batch event that contains the ID of the delete marker  
                 to remove.  
    :param context: Context about the event.  
    :return: A result structure that Amazon S3 uses to interpret the result of the  
            operation. When the result code is TemporaryFailure, S3 retries the  
    """
```

```
        operation.  
    """  
    # Parse job parameters from Amazon S3 batch operations  
    invocation_id = event['invocationId']  
    invocation_schema_version = event['invocationSchemaVersion']  
  
    results = []  
    result_code = None  
    result_string = None  
  
    task = event['tasks'][0]  
    task_id = task['taskId']  
  
    try:  
        obj_key = parse.unquote(task['s3Key'], encoding='utf-8')  
        obj_version_id = task['s3VersionId']  
        bucket_name = task['s3BucketArn'].split(':')[ -1]  
  
        logger.info("Got task: remove delete marker %s from object %s.",  
                   obj_version_id, obj_key)  
  
        try:  
            # If this call does not raise an error, the object version is not a delete  
            # marker and should not be deleted.  
            response = s3.head_object(  
                Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id)  
            result_code = 'PermanentFailure'  
            result_string = f"Object {obj_key}, ID {obj_version_id} is not "  
                           f"a delete marker."  
  
            logger.debug(response)  
            logger.warning(result_string)  
        except ClientError as error:  
            delete_marker = error.response['ResponseMetadata']['HTTPHeaders'] \  
                           .get('x-amz-delete-marker', 'false')  
            if delete_marker == 'true':  
                logger.info("Object %s, version %s is a delete marker.",  
                           obj_key, obj_version_id)  
                try:  
                    s3.delete_object(  
                        Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id)  
                    result_code = 'Succeeded'  
                    result_string = f"Successfully removed delete marker "  
                                   f"{obj_version_id} from object {obj_key}."  
                    logger.info(result_string)  
                except ClientError as error:  
                    # Mark request timeout as a temporary failure so it will be retried.  
                    if error.response['Error']['Code'] == 'RequestTimeout':  
                        result_code = 'TemporaryFailure'  
                        result_string = f"Attempt to remove delete marker from "  
                                       f"object {obj_key} timed out."  
                        logger.info(result_string)  
                    else:  
                        raise  
                else:  
                    raise ValueError(f"The x-amz-delete-marker header is either not "  
                                    f"present or is not 'true'.")  
            except Exception as error:  
                # Mark all other exceptions as permanent failures.  
                result_code = 'PermanentFailure'  
                result_string = str(error)  
                logger.exception(error)  
        finally:  
            results.append({  
                'taskId': task_id,  
                'resultCode': result_code,
```

```
        'resultString': result_string
    })
return {
    'invocationSchemaVersion': invocation_schema_version,
    'treatMissingKeysAs': 'PermanentFailure',
    'invocationId': invocation_id,
    'results': results
}
```

Criar um trabalho de operações em lote do S3 que chama uma função do Lambda

Ao criar um trabalho de operações em lote do S3 para chamar uma função do Lambda, forneça o seguinte:

- O ARN da função do Lambda (que pode incluir o alias da função ou o número de uma versão específica)
- Uma função do IAM com permissão para invocar a função
- O parâmetro de ação `LambdaInvokeFunction`

Para obter mais informações sobre como criar um trabalho de operações em lote do S3, consulte [Criar um trabalho de operações em lote do S3 \(p. 946\)](#) e [Operações suportadas pelo S3 Batch Operations \(p. 953\)](#).

O exemplo a seguir cria um trabalho do S3 Batch Operations que invoca uma função do Lambda usando a AWS CLI.

```
aws s3control create-job
--account-id <AccountID>
--operation '{"LambdaInvoke": { "FunctionArn":
"arn:aws:lambda:Region:AccountID:function:LambdaFunctionName" } }'
--manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820","Fields":
["Bucket","Key"]}}, "Location":
{"ObjectArn":"arn:aws:s3:::ManifestLocation", "ETag":"ManifestETag"}'
--report
'{"Bucket":"arn:aws:s3:::awsexamplebucket1", "Format":"Report_CSV_20180820", "Enabled":true, "Prefix": "Re
--priority 2
--role-arn arn:aws:iam::AccountID:role/BatchOperationsRole
--region Region
--description "Lambda Function"
```

Fornecer informações em nível de tarefa em manifestos do Lambda

Ao usar funções do AWS Lambda com o S3 Batch Operations, talvez você queira que dados adicionais acompanhem cada tarefa/chave em que são operados. Por exemplo, você pode querer fornecer uma chave de objeto de origem e uma nova chave de objeto. A função do Lambda pode copiar a chave de origem em um novo bucket do S3 com um novo nome. Por padrão, as operações em lote do Amazon S3 permitem especificar apenas o bucket de destino e uma lista de chaves de origem no manifesto de entrada do trabalho. O processo a seguir descreve como você pode incluir dados adicionais em seu manifesto para poder executar funções do Lambda mais complexas.

Para especificar parâmetros por chave no manifesto de operações em lote do S3 para usar no código da função do Lambda, use o seguinte formato JSON codificado por URL. O campo `key` é passado para a função do Lambda como se fosse uma chave de objeto do Amazon S3. Porém, ele pode ser interpretado pela função do Lambda como contendo outros valores ou várias chaves, como mostrado a seguir.

Note

O número máximo de caracteres para o campo `key` no manifesto é 1.024.

Example manifesto que substitui as "chaves do Amazon S3" por strings JSON

A versão codificada por URL deve ser fornecida para operações em lote do S3.

```
my-bucket,{"origKey": "object1key", "newKey": "newObject1Key"}  
my-bucket,{"origKey": "object2key", "newKey": "newObject2Key"}  
my-bucket,{"origKey": "object3key", "newKey": "newObject3Key"}
```

Example Manifesto codificado por URL

Essa versão codificada por URL deve ser fornecida para operações em lote do S3. A versão não codificada em URL não funciona.

```
my-bucket,%7B%22origKey%22%3A%20%22object1key%22%2C%20%22newKey%22%3A%20%22newObject1Key  
%22%7D  
my-bucket,%7B%22origKey%22%3A%20%22object2key%22%2C%20%22newKey%22%3A%20%22newObject2Key  
%22%7D  
my-bucket,%7B%22origKey%22%3A%20%22object3key%22%2C%20%22newKey%22%3A%20%22newObject3Key  
%22%7D
```

Example função do Lambda com formato de manifesto que grava resultados no relatório de trabalho

Esta função do Lambda mostra como analisar uma tarefa delimitada por pipe que é codificada no manifesto das operações em lote do S3. A tarefa indica qual operação de revisão é aplicada ao objeto especificado.

```
import logging  
from urllib import parse  
import boto3  
from botocore.exceptions import ClientError  
  
logger = logging.getLogger(__name__)  
logger.setLevel('INFO')  
  
s3 = boto3.resource('s3')  
  
  
def lambda_handler(event, context):  
    """  
    Applies the specified revision to the specified object.  
  
    :param event: The Amazon S3 batch event that contains the ID of the object to  
                 revise and the revision type to apply.  
    :param context: Context about the event.  
    :return: A result structure that Amazon S3 uses to interpret the result of the  
            operation.  
    """  
    # Parse job parameters from Amazon S3 batch operations  
    invocation_id = event['invocationId']  
    invocation_schema_version = event['invocationSchemaVersion']  
  
    results = []  
    result_code = None  
    result_string = None  
  
    task = event['tasks'][0]  
    task_id = task['taskId']  
    # The revision type is packed with the object key as a pipe-delimited string.  
    obj_key, revision = \
```

```
parse.unquote(task['s3Key']), encoding='utf-8').split('|')
bucket_name = task['s3BucketArn'].split(':')[ -1]

logger.info("Got task: apply revision %s to %s.", revision, obj_key)

try:
    stanza_obj = s3.Bucket(bucket_name).Object(obj_key)
    stanza = stanza_obj.get()['Body'].read().decode('utf-8')
    if revision == 'lower':
        stanza = stanza.lower()
    elif revision == 'upper':
        stanza = stanza.upper()
    elif revision == 'reverse':
        stanza = stanza[::-1]
    elif revision == 'delete':
        pass
    else:
        raise TypeError(f"Can't handle revision type '{revision}'.")

    if revision == 'delete':
        stanza_obj.delete()
        result_string = f"Deleted stanza {stanza_obj.key}."

    else:
        stanza_obj.put(Body=bytes(stanza, 'utf-8'))
        result_string = f"Applied revision type '{revision}' to " \
                        f"stanza {stanza_obj.key}."

    logger.info(result_string)
    result_code = 'Succeeded'
except ClientError as error:
    if error.response['Error']['Code'] == 'NoSuchKey':
        result_code = 'Succeeded'
        result_string = f"Stanza {obj_key} not found, assuming it was deleted " \
                        f"in an earlier revision."
        logger.info(result_string)
    else:
        result_code = 'PermanentFailure'
        result_string = f"Got exception when applying revision type '{revision}' " \
                        f"to {obj_key}: {error}."
        logger.exception(result_string)
finally:
    results.append({
        'taskId': task_id,
        'resultCode': result_code,
        'resultString': result_string
    })
return {
    'invocationSchemaVersion': invocation_schema_version,
    'treatMissingKeysAs': 'PermanentFailure',
    'invocationId': invocation_id,
    'results': results
}
```

Substituir todas as tags de objeto

A operação Substituir todas as tags de objeto substitui as tags de objeto do Amazon S3 em cada objeto listado no manifesto. A tag de objeto do Amazon S3 é um par de chave-valor de string que pode ser usada para armazenar metadados sobre um objeto.

Para criar um trabalho Substituir todas as tags de objeto, forneça um conjunto de tags que pretende aplicar. As Operações em lote do Amazon S3 aplica o mesmo conjunto de tags a cada objeto. O conjunto de tags que você fornece substitui os conjuntos de tags já associados aos objetos no manifesto. As

Operações em lote do S3 não são compatíveis com a adição de tags a objetos, deixando as tags existentes no lugar.

Se os objetos em seu manifesto estiverem em um bucket versionado, você pode 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 um ID de versão para nenhum objeto, as Operações em lote do S3 aplicam o conjunto de tags à versão mais recente de cada objeto.

Restrições e limitações

- A função do AWS Identity and Access Management (IAM) que você especificar para executar o trabalho do Batch Operations deve ter permissões para executar a operação Replace all object tags (Substituir todas as tags de objeto) do Amazon S3 subjacente. Para obter mais informações sobre as permissões necessárias, consulte [PutObjectTagging](#) na Referência da API do Amazon Simple Storage Service.
- As Operações em lote do S3 usam a operação [PutObjectTagging](#) do Amazon S3 para aplicar tags a cada objeto no manifesto. Todas as restrições e limitações que se aplicam à operação subjacente também se aplicam a trabalhos das Operações em lote do S3.

Para obter mais informações sobre como usar o console para criar trabalhos, consulte [Criar um trabalho de operações em lote do S3](#).

Para obter mais informações sobre a marcação de objetos, consulte [Categorizando seu armazenamento usando tags \(p. 878\)](#) neste guia, além de [PutObjectTagging](#), [GetObjectTagging](#), e [DeleteObjectTagging](#) na Amazon Simple Storage Service API Reference (Referência de APIs do Amazon Simple Storage Service).

Excluir todas as tags de objeto

A operação Excluir todas as tags de objeto remove todos os conjuntos de tags de objeto do Amazon S3 atualmente associados aos objetos listados no manifesto. As Operações em lote do S3 não são compatíveis com a exclusão de tags de objetos enquanto mantém outras tags no lugar.

Se os objetos no manifesto estiverem em um bucket versionado, você poderá remover os conjuntos de tags de uma versão específica de um objeto. Para fazer isso, especifique um ID de versão para cada objeto no manifesto. Se você não incluir um ID de versão para um objeto, as Operações em lote do S3 removerão o conjunto de tags da versão mais recente de cada objeto.

Para obter mais informações sobre o recurso Operações em lote, consulte [Especificar um manifesto \(p. 948\)](#).

Warning

A execução deste trabalho remove todos os conjuntos de tags de objeto em cada objeto listado no manifesto.

Restrições e limitações

- A função do AWS Identity and Access Management (IAM) que você especificar para executar o trabalho deverá ter permissões para executar a operação Delete object tagging (Excluir marcação de objetos) do Amazon S3 subjacente. Para obter mais informações, consulte [DeleteObjectTagging](#) na Referência da API do Amazon Simple Storage Service.
- As Operações em lote do S3 usam a operação [DeleteObjectTagging](#) do Amazon S3 para remover os conjuntos de tags de cada objeto no manifesto. Todas as restrições e limitações que se aplicam à operação subjacente também se aplicam a trabalhos das Operações em lote do S3.

Para obter mais informações sobre como criar trabalhos, consulte [Criar um trabalho de operações em lote do S3 \(p. 946\)](#).

Para obter mais detalhes sobre a marcação de objetos, consulte [Substituir todas as tags de objeto \(p. 973\)](#) neste guia e [PutObjectTagging](#), [GetObjectTagging](#) e [DeleteObjectTagging](#) na Referência da API do Amazon Simple Storage Service.

Substituir lista de controle de acesso

A operação Substituir lista de controle de acesso 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.

As operações em lote do S3 são compatíveis com ACLs que você define e com ACLs pré-configuradas fornecidas pelo Amazon S3 com um conjunto predefinido de permissões de acesso.

Se os objetos em seu manifesto estiverem em um bucket versionado, é possível 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 um ID de versão para um objeto, as operações em lote do S3 aplicarão a ACL à versão mais recente do objeto.

Para obter mais informações sobre ACLs no Amazon S3, consulte [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#).

Bloqueio de acesso público do S3

Para limitar o acesso público a todos os objetos em um bucket, use o Amazon S3 Block Public Access em vez do S3 Batch Operations. O Block Public Access 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. Usar as Operações em lote do S3 quando precisar aplicar uma ACL personalizada a cada objeto no manifesto. Para obter mais informações sobre o S3 Block Public Access, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

Propriedade de objetos do S3

Se os objetos no manifesto estiverem em um bucket que usa a configuração aplicada pelo proprietário do bucket para Object Ownership, a operação Replace access control list (ACL) (Substituir lista de controle de acesso (ACL)) poderá especificar apenas ACLs do objeto que concedem controle total ao proprietário do bucket. A operação não pode conceder permissões de ACL de objeto a outros grupos ou Contas da AWS. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Restrições e limitações

- A função especificada para executar o trabalho Substituir lista de controle de acesso deve ter permissões para executar a operação subjacente do Amazon S3 `PutObjectAcl`. Para obter mais informações sobre as permissões necessárias, consulte [PutObjectAcl](#) na Referência da API do Amazon Simple Storage Service.
- O S3 Batch Operations usa a operação `PutObjectAcl` do Amazon S3 para aplicar a ACL especificada a cada objeto no manifesto. Portanto, todas as restrições e limitações que se aplicam à operação `PutObjectAcl` subjacente também se aplicam a trabalhos Substituir lista de controle de acesso das Operações em lote do S3.

Restaurar objetos

A operação Restore inicia solicitações de restauração para objetos arquivados em uma lista de objetos do Amazon S3 que você especificar. Os seguintes objetos devem ser restaurados com um trabalho [Iniciar a restauração de um objeto do S3](#) antes que possam ser acessados em tempo real:

- Objetos arquivados nas classes de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive
- Objetos arquivados por meio da classe de armazenamento do S3 Intelligent-Tiering nos níveis Acesso de arquivamento ou Acesso de arquivamento profundo

O uso de uma operação Iniciar a restauração de um objeto do S3 no trabalho das Operações em lote do S3 resulta em uma solicitação de restauração para cada objeto especificado no manifesto.

Important

O trabalho Iniciar a restauração de um objeto do S3 apenas inicia a solicitação para restaurar objetos. As Operações em lote do S3 relatam o trabalho como concluído para cada objeto depois que a solicitação é iniciada para aquele objeto. O Amazon S3 não atualiza o trabalho nem o notificará quando os objetos tiverem sido restaurados. No entanto, é possível usar notificações de eventos para receber alertas quando os objetos estiverem disponíveis no Amazon S3. Para obter mais informações, consulte [Notificações de eventos do Amazon S3 \(p. 1083\)](#).

A restauração de arquivos arquivados das classes de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive difere da restauração de arquivos da classe de armazenamento S3 Intelligent-Tiering nos níveis Acesso de arquivamento ou Acesso de arquivamento profundo.

- Quando você restaura do S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, uma cópia temporária do objeto é criada. O Amazon S3 exclui essa cópia após `ExpirationInDays` dias. Depois que essa cópia for excluída, é preciso enviar uma solicitação de restauração adicional para acessá-la.
- Quando você restaura a partir dos níveis Acesso de arquivamento ou Acesso de arquivamento profundo do S3 Intelligent-Tiering, o objeto faz a transição de volta para o nível Acesso frequente do S3 Intelligent-Tiering. O objeto transita automaticamente para o nível de Acesso de arquivamento após um mínimo de 90 dias consecutivos sem acesso. Ele passa para o nível Acesso de arquivamento profundo após um mínimo de 180 dias consecutivos sem acesso. Não especifique o argumento `ExpirationInDays` ao restaurar objetos arquivados do S3 Intelligent-Tiering.
- Os trabalhos do Batch Operations podem operar em objetos de classe de armazenamento S3 Glacier e S3 Glacier Deep Archive ou em objetos de nível de armazenamento Acesso de arquivamento e Acesso de arquivamento profundo do S3 Intelligent-Tiering. Eles não podem operar em ambos os tipos no mesmo trabalho. Para restaurar objetos de ambos os tipos, você deve criar trabalhos de Operações em lote separados.

Para criar um trabalho Iniciar a restauração de um objeto do S3, os seguintes argumentos estão disponíveis:

ExpirationInDays

Esse argumento especifica por quanto tempo o objeto do S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive permanecerá disponível no Amazon S3. Os trabalhos do `Initiate Restore Object` voltados a objetos do S3 Glacier Flexible Retrieval e do S3 Glacier Deep Archive exigem `ExpirationInDays` definido como 1 ou superior.

Por outro lado, não defina `ExpirationInDays` ao criar trabalhos de operação Iniciar a restauração de um objeto do S3 que visam objetos do nível Acesso de arquivamento e Acesso de arquivamento profundo do S3 Intelligent-Tiering. Os objetos nos níveis de acesso de arquivamento do S3 Intelligent-Tiering não estão sujeitos à expiração da restauração, portanto, especificar `ExpirationInDays` resulta na falha na solicitação de restauração.

GlacierJobTier

O Amazon S3 pode restaurar objetos usando um dos três níveis de recuperação diferentes: `EXPEDITED`, `STANDARD` e `BULK`. No entanto, o recurso Operações em lote do S3 suporta apenas os níveis de recuperação `STANDARD` e `BULK`. Para obter mais informações sobre as diferenças entre os níveis de recuperação, consulte [Opções de recuperação de arquivamento \(p. 713\)](#). Para obter mais

informações sobre definição de preço para cada nível, consulte a seção [Solicitações e recuperações de dados](#) na seção [Definição de preço do Amazon S3](#).

Sobrepor restaurações

Se o trabalho [Iniciar a restauração de um objeto do S3](#) tentar restaurar um objeto que já esteja em processo de restauração, o comportamento das Operações em lote do S3 será o seguinte.

A operação de restauração 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 de restauração anterior já foi concluída e o objeto está disponível no momento. Nesse caso, as Operações em lote atualizam a data de expiração do objeto restaurado para corresponder ao `ExpirationInDays` especificado na solicitação de restauração em andamento.

A operação apresenta falha para o objeto se uma das seguintes condições for verdadeira:

- A solicitação de restauração já em andamento ainda não foi concluída e a duração da restauração para este trabalho (especificada por `ExpirationInDays`) é diferente da duração da restauração especificada na solicitação de restauração 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.

Limitations

Os trabalhos Iniciar a restauração de um objeto do S3 têm as seguintes limitações:

- Você deve criar o trabalho na mesma região que os objetos arquivados.
- As Operações em lote do S3 não são compatíveis com o nível de recuperação `EXPEDITED`.
- O S3 Batch Operations não é compatível com a restauração de subconjuntos de objetos S3 Intelligent-Tiering ou S3 Glacier Flexible Retrieval. Você deve chamar [RestoreObject](#) para este fim.

Para obter mais informações sobre a restauração de objetos, consulte [Restaurar um objeto arquivado](#) (p. 715).

Retenção do Bloqueio de objetos do S3

A operação de retenção de bloqueio de objetos permite que você aplique datas de retenção para seus objetos usando o modo de governança ou de conformidade. Esses modos de retenção aplicam níveis diferentes de proteção. É possível aplicar qualquer modo de retenção a qualquer versão de objeto. Datas de retenção, como retenções legais, impedem que um objeto seja substituído ou excluído. O Amazon S3 armazena a opção `Retain Until Date` (Reter até uma determinada data) especificada nos metadados do objeto e protege a versão especificada da versão do objeto até que o período de retenção expire.

É possível usar as operações em lote do S3 com bloqueio de objetos para gerenciar as datas de retenção de muitos objetos do Amazon S3 de uma só vez. Especifique a lista de objetos de destino no manifesto e envie-a para as operações em lote para conclusão. Para obter mais informações, consulte [Bloqueio de objetos do S3](#) [the section called “Períodos de retenção”](#) (p. 724).

As datas de retenção de trabalhos de operações em lote do S3 com datas de retenção é executado até que a conclusão, o cancelamento ou um estado de falha seja atingido. Use as operações em lote do S3 e a retenção de bloqueio de objetos do S3 ao adicionar, alterar ou remover a data de retenção de muitos objetos com uma única solicitação.

As operações em lote verificam se o bloqueio de objetos está habilitado no bucket antes de processar qualquer chave no manifesto. Para executar as operações e a validação, as operações em lote precisam das permissões `s3:GetBucketObjectLockConfiguration` e `s3:PutObjectRetention` em uma função do IAM para permitir que as operações em lote chamem o bloqueio de objetos em seu nome. Para obter mais informações, consulte [the section called “Gerenciar o bloqueio de objetos” \(p. 728\)](#).

Para obter informações sobre como usar essa operação com a API REST, consulte `S3PutObjectRetention` na operação [CreateJob](#) na Referência da API do Amazon Simple Storage Service.

Para obter um exemplo de AWS Command Line Interface de uso dessa operação, consulte [the section called “Usar operações em lote com retenção de bloqueio de objetos” \(p. 1007\)](#). Para ver um exemplo de AWS SDK for Java, consulte [the section called “Usar operações em lote com retenção de bloqueio de objetos” \(p. 1007\)](#).

Restrições e limitações

- As operações em lote do S3 não fazem alterações em nível de bucket.
- O versionamento e o bloqueio de objetos do S3 devem ser configurados no bucket em que o trabalho é executado.
- Todos os objetos listados no manifesto devem estar no mesmo bucket.
- A operação funciona na versão mais recente do objeto, a menos que uma versão seja explicitamente especificada no manifesto.
- É necessária a permissão `s3:PutObjectRetention` na função do IAM para usar isso.
- `s3:GetBucketObjectLockConfiguration` A permissão do IAM é necessária para confirmar que o bloqueio de objetos está habilitado para o bucket do S3.
- Só é possível estender o período de retenção de objetos com datas de retenção do modo `COMPLIANCE` aplicadas, e não é possível reduzi-lo.

Retenção legal do Bloqueio de objetos do S3

A operação de retenção legal do Bloqueio de Objetos permite que você coloque uma retenção legal em uma versão de objeto. Assim como a definição de 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.

Você pode usar o Operações em lote do S3 com o Bloqueio de objetos para adicionar retenções legais a muitos objetos do Amazon S3 de uma só vez. Você pode fazer isso listando os objetos de destino no manifesto e enviando essa lista às operações em lote. O trabalho de operações em lote do S3 com retenção legal de bloqueio de objetos é executado até a conclusão, até o cancelamento ou até que um estado de falha seja atingido.

As operações em lote do S3 verificam se o bloqueio de objetos está habilitado no bucket do S3 antes de processar uma chave no manifesto. Para executar as operações de objeto e a validação em nível de bucket, as Operações em lote do S3 precisam de `s3:PutObjectLegalHold` e `s3:GetBucketObjectLockConfiguration` em uma função do IAM, que permitem que as Operações em lote do S3 chamem o Bloqueio de objetos do S3 em seu nome.

Ao criar o trabalho de operações em lote do S3 para remover a retenção legal, é necessário especificar `Off` (Desativado) como o status de retenção legal. Para obter mais informações, consulte [the section called “Gerenciar o bloqueio de objetos” \(p. 728\)](#).

Para obter informações sobre como usar essa operação com a API REST, consulte `S3PutObjectLegalHold` na operação [CreateJob](#) na Referência da API do Amazon Simple Storage Service.

Para obter um exemplo de uso desta operação, consulte [Usar o AWS SDK for Java \(p. 1018\)](#).

Restrições e limitações

- As operações em lote do S3 não fazem alterações em nível de bucket.
 - Todos os objetos listados no manifesto devem estar no mesmo bucket.
 - O versionamento e o bloqueio de objetos do S3 devem ser configurados no bucket em que o trabalho é executado.
 - A operação funciona na versão mais recente do objeto, a menos que uma versão seja explicitamente especificada no manifesto.
 - `s3:PutObjectLegalHold`A permissão é necessária na função do IAM para adicionar ou remover a retenção legal de objetos.
 - `s3:GetBucketObjectLockConfiguration`A permissão do IAM é necessária para confirmar se o bloqueio de objetos do S3 está habilitado para o bucket do S3.
-
- [Copiar objetos \(p. 953\)](#)
 - [Invocar função do AWS Lambda \(p. 966\)](#)
 - [Substituir todas as tags de objeto \(p. 973\)](#)
 - [Excluir todas as tags de objeto \(p. 974\)](#)
 - [Substituir lista de controle de acesso \(p. 975\)](#)
 - [Restaurar objetos \(p. 975\)](#)
 - [Retenção do Bloqueio de objetos do S3 \(p. 977\)](#)
 - [Retenção legal do Bloqueio de objetos do S3 \(p. 978\)](#)
 - [Replicar objetos existentes com o S3 Batch Replication \(p. 849\)](#)

Gerenciar trabalhos de operações em lote do S3

O Amazon S3 oferece um conjunto de ferramentas robusto para ajudar a gerenciar trabalhos do S3 Batch Operations após a criação deles. Esta seção descreve as operações que você pode usar para gerenciar e rastrear os trabalhos usando o AWS Management Console, a AWS CLI, os AWS SDKs ou a API REST.

Tópicos

- [Usar o console do Simple Storage Service \(Amazon S3\) para gerenciar trabalhos do S3 Batch Operations \(p. 979\)](#)
- [Listar trabalhos \(p. 980\)](#)
- [Visualizar detalhes do trabalho \(p. 980\)](#)
- [Atribuir prioridade aos trabalhos \(p. 980\)](#)

Usar o console do Simple Storage Service (Amazon S3) para gerenciar trabalhos do S3 Batch Operations

Usar o console para gerenciar seus trabalhos do S3 Batch Operations. Por exemplo, é possível:

- Exibir trabalhos ativos e enfileirados
- Alterar a prioridade de um trabalho
- Confirmar e executar um trabalho
- Clonar um trabalho
- Cancelar um trabalho

Para gerenciar trabalhos do S3 Batch Operations usando o console

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação esquerdo, escolha Batch Operations (Operações em lote).
3. Escolha o trabalho específico que você deseja gerenciar.

Listar trabalhos

Você pode recuperar uma lista dos seus trabalhos de operações em lote do S3. 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.

Obter uma lista de trabalhos Ativos e Concluídos

O exemplo de AWS CLI a seguir obtém uma 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
```

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 retornadas em uma lista de trabalhos, os detalhes de um trabalho individual incluem outros itens, como:

- Os parâmetros da operação
- Detalhes sobre o manifesto
- Informações sobre o relatório de conclusão (se você configurou um ao criar o trabalho)
- O nome do recurso da Amazon (ARN) da função de usuário à qual você atribuiu a execução do trabalho

Exibindo os detalhes de um trabalho individual, você acessa toda a configuração de um trabalho.

Obter a descrição de um trabalho de operações em lote do S3

O exemplo a seguir obtém a descrição de um trabalho do S3 Batch Operations usando a AWS CLI.

```
aws s3control describe-job \
--region us-west-2 \
--account-id acct-id \
--job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

Atribuir prioridade aos trabalhos

Você pode atribuir uma prioridade numérica a cada trabalho, que pode ser qualquer inteiro positivo. As operações em lote do S3 priorizam trabalhos de acordo com a prioridade 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.

A alteração da prioridade do trabalho não afeta a velocidade de processamento do trabalho.

Note

As operações em lote do S3 respeitam as prioridades de trabalho com o 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.

Uso do console do S3

Como atualizar a prioridade do trabalho no AWS Management Console

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação esquerdo, escolha Batch Operations (Operações em lote).
3. Selecione o trabalho específico que você deseja gerenciar.
4. Escolha Actions (Ações). Na lista suspensa, escolha Update priority (Atualizar prioridade).

Usar a AWS CLI

O exemplo a seguir atualiza a prioridade do trabalho usando a AWS CLI. Um número maior indica uma prioridade de execução mais alta.

```
aws s3control update-job-priority \
  --region us-west-2 \
  --account-id acct-id \
  --priority 98 \
  --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

Usar a AWS SDK for Java

O exemplo a seguir atualiza a prioridade de um trabalho do S3 Batch Operations usando o AWS SDK for Java.

Para obter mais informações sobre prioridade de trabalhos, consulte [Atribuir prioridade aos trabalhos \(p. 980\)](#).

Example

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.UpdateJobPriorityRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;
```

```
public class UpdateJobPriority {  
    public static void main(String[] args) {  
        String accountId = "Account ID";  
        String jobId = "00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c";  
  
        try {  
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()  
                .withCredentials(new ProfileCredentialsProvider())  
                .withRegion(US_WEST_2)  
                .build();  
  
            s3ControlClient.updateJobPriority(new UpdateJobPriorityRequest()  
                .withAccountId(accountId)  
                .withJobId(jobId)  
                .withPriority(98));  
  
        } catch (AmazonServiceException e) {  
            // The call was transmitted successfully, but Amazon S3 couldn't process  
            // it and returned an error response.  
            e.printStackTrace();  
        } catch (SdkClientException e) {  
            // Amazon S3 couldn't be contacted for a response, or the client  
            // couldn't parse the response from Amazon S3.  
            e.printStackTrace();  
        }  
    }  
}
```

Rastreamento de relatórios de status e conclusão

Com o S3 Batch Operations, você pode exibir e atualizar o status do trabalho, adicionar notificações e registro em log, rastrear falhas de trabalho e gerar relatórios de conclusão.

Tópicos

- [Status de trabalhos \(p. 982\)](#)
- [Atualizar status do trabalho \(p. 985\)](#)
- [Notificações e registro em log \(p. 986\)](#)
- [Monitorar falhas nos trabalhos \(p. 986\)](#)
- [Relatórios de conclusão \(p. 987\)](#)
- [Monitorar um trabalho de operações em lote do S3 no Amazon EventBridge por meio do AWS CloudTrail \(p. 987\)](#)
- [Exemplos: relatórios de conclusão de operações em lote do S3 \(p. 990\)](#)

Status de trabalhos

Depois de criar um trabalho, ele passa por diversos status. A tabela a seguir descreve os status e as possíveis transições entre eles.

Status	Descrição	Transições
New	Ao ser criado, o trabalho ganha o status New.	Um trabalho passa automaticamente para o estado Preparing quando o Amazon S3 começa a processar o objeto do manifesto.

Status	Descrição	Transições
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 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 Suspended imediatamente depois de Preparing. Depois de confirmar que deseja executar o trabalho e ele entra no estado Ready, ele não retorna mais a Suspended.	Depois de confirmar que deseja executar o trabalho, o status muda para Ready.
Ready	O Amazon S3 está pronto para executar as operações do objeto solicitadas.	Um trabalho passa automaticamente para Active quando o Amazon S3 começa a executá-lo. O tempo que o trabalho permanece no estado Ready depende do fato de outros trabalhos de maior prioridade estarem sendo executados e do tempo necessários para conclui-los.

Status	Descrição	Transições
Active	O Amazon S3 está executando a operação solicitada nos objetos listados no manifesto. Enquanto um trabalho está no status Active, é possível monitorar seu progresso usando o console do Amazon S3 ou a operação <code>DescribeJob</code> por meio da API REST, AWS CLI ou AWS SDKs.	O trabalho deixa o estado Active 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 Paused.	O trabalho passa automaticamente para Paused quando o estado Pausing é concluído.
Paused	O trabalho pode assumir o estado Paused se você enviar outro trabalho de maior prioridade enquanto o atual está sendo executado.	O trabalho Paused volta automaticamente para Active depois que os trabalhos de prioridade mais alta que estavam bloqueando sua execução são concluídos, suspensos ou falham.
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 S3 Batch Operations 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.

Status	Descrição	Transições
Failed	O trabalho apresentou falha e não está mais em execução. Para obter mais informações sobre falhas em trabalhos, consulte Monitorar falhas nos trabalhos (p. 986) .	Failed é um estado terminal. Quando o trabalho atinge Failed, ele não muda mais de estado.

Atualizar status do trabalho

Os exemplos da AWS CLI e do SDK for Java a seguir atualizam o status de um trabalho do Batch Operations. Para obter mais informações sobre como usar o console do S3 para gerenciar trabalhos do Operações em lote, consulte [Usar o console do Simple Storage Service \(Amazon S3\) para gerenciar trabalhos do S3 Batch Operations \(p. 979\)](#).

Usar a AWS CLI

- Se você não tiver especificado o parâmetro `--no-confirmation-required` no exemplo `create-job` anterior, o trabalho permanecerá em um estado suspenso até você confirmar o trabalho definindo o status dele como Ready. O Amazon S3 torna o trabalho qualificado 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'
```

- 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
```

Usar o AWS SDK for Java

O exemplo a seguir atualiza o status de um trabalho do S3 Batch Operations usando o AWS SDK for Java.

Para obter mais informações sobre o status de trabalhos, consulte [Rastreamento de relatórios de status e conclusão \(p. 982\)](#).

Example

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSControl;
import com.amazonaws.services.s3control.AWSControlClient;
import com.amazonaws.services.s3control.model.UpdateJobStatusRequest;
```

```
import static com.amazonaws.regions.Regions.US_WEST_2;

public class UpdateJobStatus {
    public static void main(String[] args) {
        String accountId = "Account ID";
        String jobId = "00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            s3ControlClient.updateJobStatus(new UpdateJobStatusRequest()
                .withAccountId(accountId)
                .withJobId(jobId)
                .withRequestedJobStatus("Ready"));

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

Notificações e registro em log

Além de solicitar relatórios de conclusão, você também pode capturar, revisar e auditar as atividades de operações em lote usando o AWS CloudTrail. Como as operações em lote usam 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, consulte os exemplos nas seções a seguir.

Note

As Operações em lote do Amazon S3 geram eventos de gerenciamento e dados no CloudTrail durante a execução do trabalho. O volume desses eventos é dimensionado com o número de chaves no manifesto de cada trabalho. Consulte a página [Definição de preço do CloudTrail](#) para obter detalhes com exemplos de como a definição de preço muda dependendo do número de trilhas que você configurou em sua conta. Para saber como configurar e registrar em log eventos para atender às suas necessidades, consulte [Criar sua primeira trilha](#) no Guia do usuário do AWS CloudTrail.

Para obter mais informações sobre eventos do Amazon S3, consulte [Notificações de eventos do Amazon S3 \(p. 1083\)](#).

Monitorar falhas nos trabalhos

Se uma operação em lote do S3 encontrar um problema que o impeça de ser executado com êxito, como não ser capaz de ler o manifesto especificado, o trabalho falhará. Quando um trabalho falha, ele gera um ou mais códigos de falha ou motivos de falha. As operações em lote do S3 armazenam os códigos e os motivos da falha com o trabalho para que você possa visualizá-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 de 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 excede 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

As operações em lote do S3 funcionam 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.

Relatórios de conclusão

Ao criar um trabalho, você solicita um relatório de conclusão. Desde que as operações em lote do S3 invoquem com êxito pelo menos uma tarefa, o Amazon S3 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. Os relatórios de conclusão oferecem uma maneira fácil de visualizar os resultados das tarefas em um formato consolidado, sem a necessidade de configurações adicionais. Para obter um exemplo de um relatório de conclusão, consulte [Exemplos: relatórios de conclusão de operações em lote do S3 \(p. 990\)](#).

Mesmo que você não configure um relatório de conclusão, ainda poderá monitorar e auditar o trabalho e suas tarefas usando o CloudTrail e o Amazon CloudWatch. Para obter mais informações, consulte a seção a seguir:

Tópicos

- [Monitorar um trabalho de operações em lote do S3 no Amazon EventBridge por meio do AWS CloudTrail \(p. 987\)](#)
- [Exemplos: relatórios de conclusão de operações em lote do S3 \(p. 990\)](#)

Monitorar um trabalho de operações em lote do S3 no Amazon EventBridge por meio do AWS CloudTrail

A atividade do trabalho de operações em lote do Amazon S3 é registrada como eventos no AWS CloudTrail. Você pode criar uma regra personalizada no Amazon EventBridge e enviar esses eventos para o recurso de notificação de destino de sua escolha, como o Amazon Simple Notification Service (Amazon SNS).

Note

O Amazon EventBridge é a maneira preferida de gerenciar seus eventos. O Amazon CloudWatch Events e o EventBridge são o mesmo serviço subjacente e API, mas o EventBridge oferece mais recursos. As alterações feitas no CloudWatch ou no EventBridge aparecem em cada console. Para obter mais informações, consulte o [Guia do usuário do Amazon EventBridge](#).

Exemplos de rastreamento

- [Eventos de operações em lote do S3 registrados no CloudTrail \(p. 988\)](#)
- [Regra do EventBridge para rastrear eventos de trabalhos de operações em lote do S3 \(p. 988\)](#)

Eventos de operações em lote do S3 registrados no CloudTrail

Quando um trabalho de operações em lote é criado, ele é gravado como um evento JobCreated no CloudTrail. À medida que o trabalho é executado, ele muda de estado durante o processamento e outros eventos JobStatusChanged são registrados no CloudTrail. Você pode visualizar esses eventos no [console do CloudTrail](#). Para obter mais informações sobre o CloudTrail, consulte o [Guia do usuário do AWS CloudTrail](#).

Note

Somente eventos status-change de trabalho de operações em lote do S3 são registrados no CloudTrail.

Example Evento de conclusão de trabalho de operações em lote do S3 registrado pelo CloudTrail

```
{  
    "eventVersion": "1.05",  
    "userIdentity": {  
        "accountId": "123456789012",  
        "invokedBy": "s3.amazonaws.com"  
    },  
    "eventTime": "2020-02-05T18:25:30Z",  
    "eventSource": "s3.amazonaws.com",  
    "eventName": "JobStatusChanged",  
    "awsRegion": "us-west-2",  
    "sourceIPAddress": "s3.amazonaws.com",  
    "userAgent": "s3.amazonaws.com",  
    "requestParameters": null,  
    "responseElements": null,  
    "eventId": "f907577b-bf3d-4c53-b9ed-8a83a118a554",  
    "readOnly": false,  
    "eventType": "AwsServiceEvent",  
    "recipientAccountId": "123412341234",  
    "serviceEventDetails": {  
        "jobId": "d6e58ec4-897a-4b6d-975f-10d7f0fb63ce",  
        "jobArn": "arn:aws:s3:us-west-2:181572960644:job/d6e58ec4-897a-4b6d-975f-10d7f0fb63ce",  
        "status": "Complete",  
        "jobEventId": "b268784cf0a66749f1a05bce259804f5",  
        "failureCodes": [],  
        "statusChangeReason": []  
    }  
}
```

Regra do EventBridge para rastrear eventos de trabalhos de operações em lote do S3

O exemplo a seguir mostra como criar uma regra no Amazon EventBridge para capturar eventos do S3 Batch Operations registrados pelo AWS CloudTrail para um destino escolhido por você.

Para fazer isso, você cria uma regra seguindo todas as etapas em [Criar uma regra do EventBridge que é acionada em uma chamada de API da AWS usando o CloudTrail](#). Cole a política de padrão de evento personalizada de operações em lote do S3, quando aplicável, e selecione o serviço de destino de sua escolha.

Política de padrão de evento personalizado de operações em lote do S

```
{  
    "source": [  
        "aws.s3"
```

```
],
  "detail-type": [
    "AWS Service Event via CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "s3.amazonaws.com"
    ],
    "eventName": [
      "JobCreated",
      "JobStatusChanged"
    ]
  }
}
```

Os exemplos a seguir são dois eventos de operações em lote que foram enviados para o Amazon Simple Queue Service (Amazon SQS) a partir de uma regra de evento do EventBridge. Um trabalho de operações em lote passa por vários estados diferentes durante o processamento (New, Preparing, Active etc.), portanto, você pode esperar receber várias mensagens para cada trabalho.

Example Evento de exemplo JobCreated

```
{
  "version": "0",
  "id": "51dc8145-541c-5518-2349-56d7dffdf2d8",
  "detail-type": "AWS Service Event via CloudTrail",
  "source": "aws.s3",
  "account": "123456789012",
  "time": "2020-02-27T15:25:49Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "eventVersion": "1.05",
    "userIdentity": {
      "accountId": "11112223334444",
      "invokedBy": "s3.amazonaws.com"
    },
    "eventTime": "2020-02-27T15:25:49Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "JobCreated",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "s3.amazonaws.com",
    "userAgent": "s3.amazonaws.com",
    "eventID": "7c38220f-f80b-4239-8b78-2ed867b7d3fa",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "serviceEventDetails": {
      "jobId": "e849b567-5232-44be-9a0c-40988f14e80c",
      "jobArn": "arn:aws:s3:us-east-1:181572960644:job/e849b567-5232-44be-9a0c-40988f14e80c",
      "status": "New",
      "jobEventId": "f177ff24f1f097b69768e327038f30ac",
      "failureCodes": [],
      "statusChangeReason": []
    }
  }
}
```

Example Evento de conclusão de trabalho JobStatusChanged

```
{
  "version": "0",
```

```
"id": "c8791abf-2af8-c754-0435-fd869ce25233",
"detail-type": "AWS Service Event via CloudTrail",
"source": "aws.s3",
"account": "123456789012",
"time": "2020-02-27T15:26:42Z",
"region": "us-east-1",
"resources": [],
"detail": {
    "eventVersion": "1.05",
    "userIdentity": {
        "accountId": "1111222233334444",
        "invokedBy": "s3.amazonaws.com"
    },
    "eventTime": "2020-02-27T15:26:42Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "JobStatusChanged",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "s3.amazonaws.com",
    "userAgent": "s3.amazonaws.com",
    "eventID": "0238c1f7-c2b0-440b-8dbd-1ed5e5833afb",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "serviceEventDetails": {
        "jobId": "e849b567-5232-44be-9a0c-40988f14e80c",
        "jobArn": "arn:aws:s3:us-east-1:181572960644:job/e849b567-5232-44be-9a0c-40988f14e80c",
        "status": "Complete",
        "jobEventId": "51f5ac17dba408301d56cd1b2c8d1e9e",
        "failureCodes": [],
        "statusChangeReason": []
    }
}
```

Exemplos: relatórios de conclusão de operações em lote do S3

Ao criar uma trabalho de operações em lote do S3, você pode solicitar um relatório de conclusão para todas as tarefas ou somente para aquelas que apresentaram falha. Contanto que pelo menos uma tarefa tenha sido invocada com êxito, as operações em lote do S3 geram um relatório para tarefas que foram concluídas, que falharam ou que foram canceladas.

O relatório de conclusão contém informações adicionais de cada tarefa, inclusive a chave e a versão do objeto, status, códigos de erro e descrições de qualquer erro. A descrição dos erros para cada tarefa com falha pode ser usada para diagnosticar problemas durante a criação de trabalhos, como permissões.

Example arquivo de resultado do manifesto de nível superior

O arquivo `manifest.json` de nível superior contém os locais de cada relatório bem-sucedido e (se a tarefa apresentou qualquer falha) o local de relatórios com falha, conforme mostrado no exemplo a seguir.

```
{
    "Format": "Report_CSV_20180820",
    "ReportCreationDate": "2019-04-05T17:48:39.725Z",
    "Results": [
        {
            "TaskExecutionStatus": "succeeded",
            "Bucket": "my-job-reports",
            "MD5Checksum": "83b1c4cbe93fc893f54053697e10fd6e",
            "Key": "job-f8fb9d89-a3aa-461d-bddc-ea6a1b131955/results/6217b0fab0de85c408b4be96aeaca9b195a7daa5.csv"
        },
        {
            "TaskExecutionStatus": "failed",
            "Bucket": "my-job-reports",
            "MD5Checksum": "83b1c4cbe93fc893f54053697e10fd6e",
            "Key": "job-f8fb9d89-a3aa-461d-bddc-ea6a1b131955/results/6217b0fab0de85c408b4be96aeaca9b195a7daa5.csv"
        }
    ]
}
```

```
        "TaskExecutionStatus": "failed",
        "Bucket": "my-job-reports",
        "MD5Checksum": "22ee037f3515975f7719699e5c416eaa",
        "Key": "job-f8fb9d89-a3aa-461d-bddc-ea6a1b131955/results/
b2ddad417e94331e9f37b44f1faf8c7ed5873f2e.csv"
    }
],
"ReportSchema": "Bucket, Key, VersionId, TaskStatus, ErrorCode, HttpStatusCode,
ResultMessage"
}
```

Example relatórios de tarefas com falha

Relatórios de tarefas com falha contêm as seguintes informações para todas as tarefas com falha:

- Bucket
- Key
- VersionId
- TaskStatus
- ErrorCode
- HttpStatusCode
- ResultMessage

O exemplo de relatório a seguir mostra um caso no qual a função do AWS Lambda atingiu o tempo limite, fazendo com que as falhas excedessem o limite de falhas. Ela foi então marcado como um PermanentFailure.

```
awsexamplebucket1,image_14975,,failed,200,PermanentFailure,"Lambda returned function error:
{""errorMessage"":""2019-04-05T17:35:21.155Z 2845ca0d-38d9-4c4b-abcf-379dc749c452 Task
timed out after 3.00 seconds""}
awsexamplebucket1,image_15897,,failed,200,PermanentFailure,"Lambda returned function error:
{""errorMessage"":""2019-04-05T17:35:29.610Z 2d0a330b-de9b-425f-b511-29232fde5fe4 Task
timed out after 3.00 seconds""}
awsexamplebucket1,image_14819,,failed,200,PermanentFailure,"Lambda returned function error:
{""errorMessage"":""2019-04-05T17:35:22.362Z fcf5efde-74d4-4e6d-b37a-c7f18827f551 Task
timed out after 3.00 seconds""}
awsexamplebucket1,image_15930,,failed,200,PermanentFailure,"Lambda returned function error:
{""errorMessage"":""2019-04-05T17:35:29.809Z 3dd5b57c-4a4a-48aa-8a35-cbf027b7957e Task
timed out after 3.00 seconds""}
awsexamplebucket1,image_17644,,failed,200,PermanentFailure,"Lambda returned function error:
{""errorMessage"":""2019-04-05T17:35:46.025Z 10a764e4-2b26-4d8c-9056-1e1072b4723f Task
timed out after 3.00 seconds""}
awsexamplebucket1,image_17398,,failed,200,PermanentFailure,"Lambda returned function error:
{""errorMessage"":""2019-04-05T17:35:44.661Z 1e306352-4c54-4eba-aee8-4d02f8c0235c Task
timed out after 3.00 seconds""}
```

Example relatório de tarefas bem-sucedidas

Os relatórios de tarefas bem-sucedidas contêm o seguinte para as tarefas concluídas:

- Bucket
- Key
- VersionId
- TaskStatus
- ErrorCode

- `HTTPStatusCode`
- `ResultMessage`

No exemplo a seguir, a função do Lambda copiou com êxito o objeto do Amazon S3 para outro bucket. A resposta retornada do Amazon S3 é passada de volta para operações em lote do S3 e, em seguida, é gravada no relatório de conclusão final.

```
awsexamplebucket1,image_17775,,succeeded,200,,"{u'CopySourceVersionId':  
'xVR78haVKlRnurYofbTfYr3ufYbktF8h', u'CopyObjectResult': {u'LastModified':  
datetime.datetime(2019, 4, 5, 17, 35, 39, tzinfo=tzlocal()), u'ETag':  
'"fe66f4390c50f29798f040d7aae72784"'}, 'ResponseMetadata': {'HTTPStatusCode':  
200, 'RetryAttempts': 0, 'HostId': 'nXNaCLIMxEJzWNmeMNQV2KpjbaCJLn0OGoXWZpuVOFS/  
iQYWxb3QtTvzX9S9Vfx21A3oTKLwImKw', 'RequestId': '3ED5852152014362', 'HTTPHeaders':  
{'content-length': '234', 'x-amz-id-2': 'nXNaCLIMxEJzWNmeMNQV2KpjbaCJLn0OGoXWZpuVOFS/  
iQYWxb3QtTvzX9S9Vfx21A3oTKLwImKw', 'x-amz-copy-source-version-id':  
'xVR78haVKlRnurYofbTfYr3ufYbktF8h', 'server': 'AmazonS3', 'x-amz-request-id':  
'3ED5852152014362', 'date': 'Fri, 05 Apr 2019 17:35:39 GMT', 'content-type': 'application/  
xml'}}}"  
awsexamplebucket1,image_17763,,succeeded,200,,"{u'CopySourceVersionId':  
'6HjOUSim4Wj6BTcbxToXW44pSZ.40pwq', u'CopyObjectResult': {u'LastModified':  
datetime.datetime(2019, 4, 5, 17, 35, 39, tzinfo=tzlocal()),  
u'ETag': '"fe66f4390c50f29798f040d7aae72784"'}, 'ResponseMetadata':  
{'HTTPStatusCode': 200, 'RetryAttempts': 0, 'HostId': 'GiCZNYr8LHd/  
Thyk6beTRP96IGZk2sYxujLe13TuuLpq6U2RD3we0Yoluuidm1PRvkMwnEW1aFc', 'RequestId':  
'1BC9F5B1B95D7000', 'HTTPHeaders': {'content-length': '234', 'x-amz-id-2': 'GiCZNYr8LHd/  
Thyk6beTRP96IGZk2sYxujLe13TuuLpq6U2RD3we0Yoluuidm1PRvkMwnEW1aFc', 'x-amz-copy-source-  
version-id': '6HjOUSim4Wj6BTcbxToXW44pSZ.40pwq', 'server': 'AmazonS3', 'x-amz-request-id':  
'1BC9F5B1B95D7000', 'date': 'Fri, 05 Apr 2019 17:35:39 GMT', 'content-type': 'application/  
xml'}}}"  
awsexamplebucket1,image_17860,,succeeded,200,,"{u'CopySourceVersionId':  
'm.MDD0g_QsUnYZ8TBzVFrp.TmjN8PJyX', u'CopyObjectResult': {u'LastModified':  
datetime.datetime(2019, 4, 5, 17, 35, 40, tzinfo=tzlocal()), u'ETag':  
'"fe66f4390c50f29798f040d7aae72784"'}, 'ResponseMetadata': {'HTTPStatusCode':  
200, 'RetryAttempts': 0, 'HostId': 'F9ooZOgpE5g9sNgBzxjdiPHqB4+0DNWgj3qbsir  
+sKai4fv7rQEcf2FBN1VeeFc2WH45a9ygb2g', 'RequestId': '8D9CA56A56813DF3', 'HTTPHeaders':  
{'content-length': '234', 'x-amz-id-2': 'F9ooZOgpE5g9sNgBzxjdiPHqB4+0DNWgj3qbsir  
+sKai4fv7rQEcf2FBN1VeeFc2WH45a9ygb2g', 'x-amz-copy-source-version-id':  
'm.MDD0g_QsUnYZ8TBzVFrp.TmjN8PJyX', 'server': 'AmazonS3', 'x-amz-request-id':  
'8D9CA56A56813DF3', 'date': 'Fri, 05 Apr 2019 17:35:40 GMT', 'content-type': 'application/  
xml'}}}"
```

Controlar o acesso e rotular trabalhos usando tags

Você pode rotular e controlar o acesso aos trabalhos de operações em lote do S3 adicionando tags. As tags podem ser usadas para identificar quem é responsável por um trabalho de operações em lote. A presença de marcações de trabalho pode conceder ou limitar a capacidade do usuário de cancelar um trabalho, ativar um trabalho no estado de confirmação ou alterar o nível de prioridade de um trabalho. Você pode criar trabalhos com tags anexadas a eles e pode adicionar tags aos trabalhos depois que eles são criados. Cada tag é um par de chave-valor que pode ser incluído quando você cria o trabalho ou atualizado posteriormente.

Warning

As tags de trabalho não devem conter informações confidenciais ou dados pessoais.

Considere o seguinte exemplo de marcação: suponha que você deseja que seu departamento financeiro crie um trabalho de operações em lote. Você pode escrever uma política do AWS Identity and Access Management (IAM) que permita que um usuário invoque `CreateJob`, desde que o trabalho seja criado com a tag `Department` atribuída ao valor `Finance`. Além disso, você pode anexar essa política a todos os usuários que são membros do departamento de Finanças.

Continuando com este exemplo, você pode escrever uma política que permita que um usuário atualize a prioridade de qualquer trabalho que tenha as tags desejadas ou cancele qualquer trabalho que tenha essas tags. Para obter mais informações, consulte [the section called “Controlar permissões” \(p. 998\)](#).

Você pode adicionar tags a novos trabalhos de operações em lote do S3 ao criá-los ou adicionar tags a trabalhos existentes.

Observe as seguintes restrições de tag:

- Você pode associar até 50 tags a um trabalho, desde que elas tenham 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.

Para obter mais informações sobre restrições de tags, consulte [Restrições de tags definidas pelo usuário no Guia do usuário do AWS Billing and Cost Management](#).

Operações de API relacionadas à marcação de trabalhos de operações em lote do S3

O Amazon S3 oferece suporte às seguintes operações de API específicas para a marcação de trabalhos de operações em lote do S3:

- [GetJobTagging](#) — retorna o conjunto de tags associado a um trabalho de operações em lote.
- [PutJobTagging](#) — substitui o conjunto de tags associado a um trabalho. Há dois cenários distintos de gerenciamento de tags de trabalho de operações em lote do S3 usando essa ação de API:
 - O trabalho não tem tags — você pode adicionar um conjunto de tags a um trabalho (o trabalho não tem tags anteriores).
 - O trabalho tem um conjunto de tags existentes — para modificar o conjunto de tags existente, é possível substituir completamente o conjunto de tags existente ou fazer alterações nele recuperando o conjunto de tags existente usando [GetJobTagging](#), modificar esse conjunto de tags e usar essa ação de API para substituir a tag definida pela que você modificou.

Note

Se você enviar essa solicitação com o conjunto de tags vazio, o Operações em lote do S3 excluirá o conjunto de tags existente no objeto. Se você usar esse método, será cobrado por uma solicitação de nível 1 (PUT). Para obter mais informações, consulte [Definição de preço do Amazon S3](#).

Para excluir tags existentes do trabalho de Operações em lote, a ação [DeleteJobTagging](#) é preferida porque ela obtém o mesmo resultado sem incorrer em encargos.

- [DeleteJobTagging](#) — exclui o conjunto de tags associado a um trabalho de Operações em lote.

Criar um trabalho do Operações em lote com tags de trabalho usadas para rotulagem

Você pode rotular e controlar o acesso aos trabalhos de operações em lote do S3 adicionando tags. As tags podem ser usadas para identificar quem é responsável por um trabalho de operações em lote. Você pode criar trabalhos com tags anexadas a eles e pode adicionar tags aos trabalhos depois que eles são criados. Para obter mais informações, consulte [the section called “Usar tags” \(p. 992\)](#).

Usar a AWS CLI

O exemplo da AWS CLI a seguir cria um trabalho `S3PutObjectCopy` do S3 Batch Operations usando tags de trabalho como rótulos para o trabalho.

1. Selecione a ação ou OPERATION que deseja que o trabalho de operações em lote execute e escolha TargetResource.

```
read -d '' OPERATION <<EOF
{
    "S3PutObjectCopy": {
        "TargetResource": "arn:aws:s3:::destination-bucket"
    }
}
EOF
```

2. Identifique as TAGS que você deseja para o trabalho. Nesse caso, você aplica duas tags, department e FiscalYear, com os valores Marketing e 2020, respectivamente.

```
read -d '' TAGS <<EOF
[
    {
        "Key": "department",
        "Value": "Marketing"
    },
    {
        "Key": "FiscalYear",
        "Value": "2020"
    }
]
EOF
```

3. Especifique o MANIFEST para o trabalho de operações em lote.

```
read -d '' MANIFEST <<EOF
{
    "Spec": {
        "Format": "EXAMPLE_S3BatchOperations_CSV_20180820",
        "Fields": [
            "Bucket",
            "Key"
        ]
    },
    "Location": {
        "ObjectArn": "arn:aws:s3:::example-bucket/example_manifest.csv",
        "ETag": "example-5dc7a8bfb90808fc5d546218"
    }
}
EOF
```

4. Configure o REPORT para o trabalho de operações em lote.

```
read -d '' REPORT <<EOF
{
    "Bucket": "arn:aws:s3:::example-report-bucket",
    "Format": "Example_Report_CSV_20180820",
    "Enabled": true,
    "Prefix": "reports/copy-with-replace-metadata",
    "ReportScope": "AllTasks"
}
EOF
```

5. Execute a ação create-job para criar o trabalho de operações em lote com entradas definidas nas etapas anteriores.

```
aws \
    s3control create-job \
```

```
--account-id 123456789012 \
--manifest "${MANIFEST//$/\n}" \
--operation "${OPERATION//$/\n/}" \
--report "${REPORT//$/\n}" \
--priority 10 \
--role-arn arn:aws:iam::123456789012:role/batch-operations-role \
--tags "${TAGS//$/\n/}" \
--client-request-token "$(uuidgen)" \
--region us-west-2 \
--description "Copy with Replace Metadata";
```

Usar o AWS SDK for Java

Example

O exemplo a seguir cria um trabalho do S3 Batch Operations com tags usando o AWS SDK for Java.

```
public String createJob(final AWSS3ControlClient awss3ControlClient) {
    final String manifestObjectArn = "arn:aws:s3:::example-manifest-bucket/
manifests/10_manifest.csv";
    final String manifestObjectVersionId = "example-5dc7a8bfb90808fc5d546218";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
        .withObjectArn(manifestObjectArn)
        .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
        new
    JobManifestSpec().withFormat(JobManifestFormat.S3InventoryReport_CSV_20161130);

    final JobManifest manifestToPublicApi = new JobManifest()
        .withLocation(manifestLocation)
        .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::example-report-bucket";
    final String jobReportPrefix = "example-job-reports";

    final JobReport jobReport = new JobReport()
        .withEnabled(true)
        .withReportScope(JobReportScope.AllTasks)
        .withBucket(jobReportBucketArn)
        .withPrefix(jobReportPrefix)
        .withFormat(JobReportFormat.Report_CSV_20180820);

    final String lambdaFunctionArn = "arn:aws:lambda:us-
west-2:123456789012:function:example-function";

    final JobOperation jobOperation = new JobOperation()
        .withLambdaInvoke(new
    LambdaInvokeOperation().withFunctionArn(lambdaFunctionArn));

    final S3Tag departmentTag = new S3Tag().withKey("department").withValue("Marketing");
    final S3Tag fiscalYearTag = new S3Tag().withKey("FiscalYear").withValue("2020");

    final String roleArn = "arn:aws:iam::123456789012:role/example-batch-operations-role";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
        .withAccountId("123456789012")
        .withDescription("Test lambda job")
        .withManifest(manifestToPublicApi)
        .withOperation(jobOperation)
```

```
.withPriority(priority)
.withRoleArn(roleArn)
.withReport(jobReport)
.withTags(departmentTag, fiscalYearTag)
.withConfirmationRequired(requiresConfirmation);

final CreateJobResult result = awss3ControlClient.createJob(request);

return result.get jobId();
}
```

Excluir as tags de um trabalho do Operações em lote do S3

Você pode usar esses exemplos para excluir as tags de um trabalho do Operações em lote.

Usar a AWS CLI

O exemplo a seguir exclui as tags de um trabalho do Batch Operations usando a AWS CLI.

```
aws \
    s3control delete-job-tagging \
    --account-id 123456789012 \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
    --region us-east-1;
```

Excluir as tags de um trabalho de operações em lote

Example

O exemplo a seguir exclui as tags de um trabalho do S3 Batch Operations usando o AWS SDK for Java.

```
public void deleteJobTagging(final AWSS3ControlClient awss3ControlClient,
    final String jobId) {
    final DeleteJobTaggingRequest deleteJobTaggingRequest = new DeleteJobTaggingRequest()
        .withJobId(jobId);

    final DeleteJobTaggingResult deleteJobTaggingResult =
        awss3ControlClient.deleteJobTagging(deleteJobTaggingRequest);
}
```

Colocar tags de trabalho para um trabalho existente do Operações em lote do S3

Você pode usar [PutJobTagging](#) para adicionar tags de trabalho aos trabalhos de operações em lote do S3 existentes. Para obter mais informações, veja os exemplos a seguir:

Usar a AWS CLI

Veja a seguir um exemplo de uso de `s3control put-job-tagging` para adicionar tags de trabalho aos seus trabalhos do S3 Batch Operations usando a AWS CLI.

Note

Se você enviar essa solicitação com o conjunto de tags vazio, o Operações em lote do S3 excluirá o conjunto de tags existente no objeto. Além disso, se você usar esse método, será cobrado por uma solicitação de nível 1 (PUT). Para obter mais informações, consulte [Definição de preço do Amazon S3](#).

Para excluir tags existentes do trabalho de Operações em lote, a ação `DeleteJobTagging` é preferida porque ela obtém o mesmo resultado sem incorrer em encargos.

- Identifique as TAGS que você deseja para o trabalho. Nesse caso, você aplica duas tags, `department` e `FiscalYear`, com os valores `Marketing` e `2020`, respectivamente.

```
read -d '' TAGS <<EOF
[
  {
    "Key": "department",
    "Value": "Marketing"
  },
  {
    "Key": "FiscalYear",
    "Value": "2020"
  }
]
EOF
```

- Execute a ação `put-job-tagging` com os parâmetros necessários.

```
aws \
  s3control put-job-tagging \
  --account-id 123456789012 \
  --tags "${TAGS//$/\n/}" \
  --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
  --region us-east-1;
```

Usar o AWS SDK for Java

Example

O exemplo a seguir coloca as tags de um trabalho do S3 Batch Operations usando o AWS SDK for Java.

```
public void putJobTagging(final AWSS3ControlClient awss3ControlClient,
                           final String jobId) {
    final S3Tag departmentTag = new S3Tag().withKey("department").withValue("Marketing");
    final S3Tag fiscalYearTag = new S3Tag().withKey("FiscalYear").withValue("2020");

    final PutJobTaggingRequest putJobTaggingRequest = new PutJobTaggingRequest()
        .withJobId(jobId)
        .withTags(departmentTag, fiscalYearTag);

    final PutJobTaggingResult putJobTaggingResult =
        awss3ControlClient.putJobTagging(putJobTaggingRequest);
}
```

Obter as tags de um trabalho do Operações em lote S3

Você pode usar `GetJobTagging` para retornar as tags de um trabalho de operações em lote do S3. Para obter mais informações, veja os exemplos a seguir:

Usar a AWS CLI

O exemplo a seguir obtém as tags de um trabalho do Batch Operations usando a AWS CLI.

```
aws \
  s3control get-job-tagging \
```

```
--account-id 123456789012 \
--job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
--region us-east-1;
```

Usar o AWS SDK for Java

Example

O exemplo a seguir obtém as tags de um trabalho do S3 Batch Operations usando a AWS SDK for Java.

```
public List<S3Tag> getJobTagging(final AWSS3ControlClient awss3ControlClient,
                                    final String jobId) {
    final GetJobTaggingRequest getJobTaggingRequest = new GetJobTaggingRequest()
        .withJobId(jobId);

    final GetJobTaggingResult getJobTaggingResult =
        awss3ControlClient.getJobTagging(getJobTaggingRequest);

    final List<S3Tag> tags = getJobTaggingResult.getTags();

    return tags;
}
```

Controlar permissões para o recurso Operações em lote do S3 usando tags de trabalho

Para ajudá-lo a gerenciar seus trabalhos de operações em lote do S3, você pode adicionar tags de trabalho. Com tags de trabalho, você pode controlar o acesso aos seus trabalhos de operações em lote e impor que as tags sejam aplicadas quando qualquer trabalho for criado.

Você pode aplicar até 50 tags de trabalho a cada trabalho de operações em lote. Isso permite que você defina políticas muito granulares restringindo o conjunto de usuários que podem editar o trabalho. As tags de trabalho podem conceder ou limitar a capacidade do usuário de cancelar um trabalho, ativar um trabalho no estado de confirmação ou alterar o nível de prioridade de um trabalho. Além disso, você pode impor que as tags sejam aplicadas a todos os novos trabalhos e especificar os pares de chave-valor permitidos para as tags. Você pode expressar todas essas condições usando a mesma [linguagem de política do IAM](#). Para obter mais informações, consulte [Ações, recursos e chaves de condição do Amazon S3 \(p. 458\)](#).

O exemplo a seguir mostra como você pode usar tags de trabalho de operações em lote do S3 para conceder aos usuários permissão para criar e editar somente os trabalhos executados em um departamento específico (por exemplo, o departamento Finanças ou Conformidade). Você também pode atribuir trabalhos com base no estágio de desenvolvimento ao qual eles estão relacionados, como QA ou Produção.

Neste exemplo, você usa tags de trabalho do S3 Batch Operations nas políticas do AWS Identity and Access Management (IAM) para conceder aos usuários permissão para criar e editar apenas os trabalhos que estão sendo executados em seu departamento. Você atribui trabalhos com base no estágio de desenvolvimento ao qual eles estão relacionados, como QA ou Produção.

Este exemplo usa os seguintes departamentos, com cada um usando operações em lote de maneiras diferentes:

- Finanças
- Conformidade
- Business Intelligence

- Engenharia

Tópicos

- [Controlar o acesso atribuindo tags a usuários e recursos \(p. 999\)](#)
- [Marcar trabalhos de operações em lote por estágio e impor limites na prioridade do trabalho \(p. 1000\)](#)

Controlar o acesso atribuindo tags a usuários e recursos

Nesse cenário, os administradores estão usando o [controle de acesso baseado em atributos \(ABAC\)](#). ABAC é uma estratégia de autorização do IAM que define permissões anexando tags a usuários do IAM e recursos da AWS.

Os usuários e os trabalhos recebem uma das seguintes tags de departamento:

Chave: valor

- department : Finance
- department : Compliance
- department : BusinessIntelligence
- department : Engineering

Note

As chaves e os valores de tags de trabalho diferenciam maiúsculas de minúsculas.

Usando a estratégia de controle de acesso ABAC, você concede a um usuário no departamento de Finanças permissão para criar e gerenciar trabalhos de operações em lote do S3 em seu departamento associando a tag department=Finance ao usuário do IAM.

Além disso, você pode anexar uma política gerenciada ao usuário do IAM que permite que qualquer usuário em sua empresa crie ou modifique trabalhos de operações em lote do S3 dentro de seus respectivos departamentos.

A política neste exemplo inclui três instruções de política:

- A primeira instrução na política permite que o usuário crie um trabalho de operações em lote desde que a solicitação de criação de trabalho inclua uma tag de trabalho que corresponda ao respectivo departamento. Isso é expresso usando a sintaxe "\${aws:PrincipalTag/department}", que é substituída pela tag de departamento do usuário do IAM no momento da avaliação da política. A condição é satisfeita quando o valor fornecido para a tag de departamento na solicitação ("aws:RequestTag/department") corresponde ao departamento do usuário.
- A segunda instrução na política permite que os usuários alterem a prioridade dos trabalhos ou atualizem o status de um trabalho, desde que o trabalho que o usuário está atualizando corresponda ao departamento do usuário.
- A terceira instrução permite que um usuário atualize as tags de um trabalho de operações em lote a qualquer momento por meio de uma solicitação PutJobTagging, desde que (1) sua tag de departamento seja preservada e (2) o trabalho que está atualizando esteja dentro de seu departamento.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "batch:CreateJobs",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "batch:UpdateJobPriority",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "batch:UpdateJobStatus",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "batch:PutJobTagging",  
            "Resource": "*"  
        }  
    ]  
}
```

```
        "Action": "s3:CreateJob",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/department": "${aws:PrincipalTag/department}"
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:UpdateJobPriority",
                "s3:UpdateJobStatus"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "s3:PutJobTagging",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/department": "${aws:PrincipalTag/
department}",
                    "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                }
            }
        }
    ]
}
```

Marcar trabalhos de operações em lote por estágio e impor limites na prioridade do trabalho

Todos os trabalhos de operações em lote do S3 têm uma prioridade numérica, que o Amazon S3 usa para decidir em que ordem executar os trabalhos. Para este exemplo, você restringe a prioridade máxima que a maioria dos usuários pode atribuir a trabalhos, com intervalos de prioridade mais altos reservados para um conjunto limitado de usuários privilegiados, da seguinte forma:

- Intervalo de prioridade do estágio de QA (baixa): 1–100
- Intervalo de prioridade do estágio de produção (alta): 1–300

Para fazer isso, introduza um novo conjunto de tags representando o estágio do trabalho:

Chave: valor

- stage : QA
- stage : Production

Criar e atualizar trabalhos de baixa prioridade em um departamento

Esta política introduz duas novas restrições à criação e atualização de trabalhos de operações em lote do S3, além da restrição baseada em departamento:

- Ele permite que os usuários criem ou atualizem trabalhos em seu departamento com uma nova condição que requer que o trabalho inclua a tag `stage=QA`.
- Ele permite que os usuários criem ou atualizem a prioridade de um trabalho até uma nova prioridade máxima de 100.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "s3:CreateJob",  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/department": "${aws:PrincipalTag/department}",  
                    "aws:RequestTag/stage": "QA"  
                },  
                "NumericLessThanEquals": {  
                    "s3:RequestJobPriority": 100  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:UpdateJobStatus"  
            ],  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:ResourceTag/department": "${aws:PrincipalTag/department}"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "s3:UpdateJobPriority",  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:ResourceTag/department": "${aws:PrincipalTag/department}",  
                    "aws:ResourceTag/stage": "QA"  
                },  
                "NumericLessThanEquals": {  
                    "s3:RequestJobPriority": 100  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "s3:PutJobTagging",  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/department" : "${aws:PrincipalTag/department}",  
                    "aws:ResourceTag/department": "${aws:PrincipalTag/department}",  
                    "aws:RequestTag/stage": "QA",  
                    "aws:ResourceTag/stage": "QA"  
                }  
            }  
        }  
    ]  
}
```

```
        },
        {
            "Effect": "Allow",
            "Action": "s3:GetJobTagging",
            "Resource": "*"
        }
    ]
}
```

Criar e atualizar trabalhos de alta prioridade em um departamento

Um pequeno número de usuários pode exigir a capacidade de criar trabalhos de alta prioridade em QA ou Produção. Para dar suporte a essa necessidade, crie uma política gerenciada adaptada da política de baixa prioridade na seção anterior.

Essa política faz o seguinte:

- Permite que os usuários criem ou atualizem trabalhos em seu departamento com a tag `stage=QA` ou `stage=Production`.
- Permite que os usuários criem ou atualizem a prioridade de um trabalho até um máximo de 300.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:CreateJob",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:RequestTag/stage": [
                        "QA",
                        "Production"
                    ]
                },
                "StringEquals": {
                    "aws:RequestTag/department": "${aws:PrincipalTag/department}"
                },
                "NumericLessThanEquals": {
                    "s3:RequestJobPriority": 300
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:UpdateJobStatus"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/department": "${aws:PrincipalTag/department}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "s3:UpdateJobPriority",
            "Resource": "*",

```

```
"Condition": {
    "ForAnyValue:StringEquals": {
        "aws:ResourceTag/stage": [
            "QA",
            "Production"
        ]
    },
    "StringEquals": {
        "aws:ResourceTag/department": "${aws:PrincipalTag/department}"
    },
    "NumericLessThanEquals": {
        "s3:RequestJobPriority": 300
    }
},
{
    "Effect": "Allow",
    "Action": "s3:PutJobTagging",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/department": "${aws:PrincipalTag/department}",
            "aws:ResourceTag/department": "${aws:PrincipalTag/department}"
        },
        "ForAnyValue:StringEquals": {
            "aws:RequestTag/stage": [
                "QA",
                "Production"
            ],
            "aws:ResourceTag/stage": [
                "QA",
                "Production"
            ]
        }
    }
}
]
```

Gerenciar o Bloqueio de objetos do S3 usando o Operações em lote do S3

Com o bloqueio de objetos do S3, você também pode colocar uma retenção legal em uma versão de objeto. Assim como a definição de 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. Para obter mais informações, consulte [Retenção legal do Bloqueio de objetos do S3 \(p. 978\)](#).

Para obter informações sobre o uso de operações em lote do S3 com o bloqueio de objetos para adicionar retenções legais a muitos objetos do Amazon S3 de uma só vez, consulte as seções a seguir.

Tópicos

- [Habilitar o Bloqueio de objetos do S3 usando o Operações em lote do S3 \(p. 1004\)](#)
- [Definir a retenção do Bloqueio de objetos usando o Operações em lote \(p. 1007\)](#)
- [Usar operações em lote do S3 com o modo de conformidade de retenção do bloqueio de objetos do S3 \(p. 1008\)](#)
- [Usar operações em lote do S3 com modo de governança de retenção de bloqueio de objetos do S3 \(p. 1012\)](#)

- Usar operações em lote do S3 para desativar a retenção legal do bloqueio de objetos do S3 (p. 1016)

Habilitar o Bloqueio de objetos do S3 usando o Operações em lote do S3

Você pode usar operações em lote do S3 com bloqueio de objeto do S3 para gerenciar a retenção ou habilitar uma retenção legal para muitos objetos do Amazon S3 de uma só vez. Especifique a lista de objetos de destino no manifesto e envie-a para as operações em lote para conclusão. Para obter mais informações, consulte [the section called “Retenção do Bloqueio de objetos” \(p. 977\)](#) e [the section called “Retenção legal do Bloqueio de objetos” \(p. 978\)](#).

Os exemplos a seguir mostram como criar uma função do IAM com permissões do Operações em lote do S3 e atualizar as permissões de função para criar trabalhos que habilitem o Bloqueio de objetos. Nos exemplos, substitua todos os valores de variável por valores adequados às suas necessidades. Você também deve ter um manifesto CSV identificando os objetos para o trabalho de operações em lote do S3. Para obter mais informações, consulte [the section called “Especificificar um manifesto” \(p. 948\)](#).

Usar a AWS CLI

1. Crie uma função do IAM e atribua permissões de operações em lote do S3 para executar.

Esta etapa é necessária para todos os trabalhos de operações em lote do S3.

```
export AWS_PROFILE='aws-user'

read -d '' bops_trust_policy <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "batchoperations.s3.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF
aws iam create-role --role-name bops-objectlock --assume-role-policy-document
"${bops_trust_policy}"
```

2. Configure operações em lote do S3 com bloqueio de objeto do S3 para executar.

Nesta etapa, permita que a função faça o seguinte:

- a. Execute o bloqueio de objeto no bucket do S3 que contém os objetos de destino nos quais você deseja que as operações em lote sejam executadas.
- b. Leia o bucket do S3 no qual estão localizados o arquivo CSV de manifesto e os objetos.
- c. Grave os resultados do trabalho de operações em lote do S3 no bucket de relatórios.

```
read -d '' bops_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {

```

```
        "Effect": "Allow",
        "Action": "s3:GetBucketObjectLockConfiguration",
        "Resource": [
            "arn:aws:s3:::{ManifestBucket}"
        ],
    },
{
    "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}/*"
    ]
}
]
}
EOF

aws iam put-role-policy --role-name bops-objectlock --policy-name object-lock-permissions --policy-document "${bops_permissions}"
```

Usar o AWS SDK for Java

Os exemplos a seguir mostram como criar uma função do IAM com permissões do S3 Batch Operations e atualizar as permissões de função para criar trabalhos que habilitem o bloqueio de objetos usando o AWS SDK for Java. No código, substitua todos os valores de variável por valores adequados às suas necessidades. Você também deve ter um manifesto CSV identificando os objetos para o trabalho de operações em lote do S3. Para obter mais informações, consulte [the section called “Especificar um manifesto” \(p. 948\)](#).

Você executa as seguintes etapas:

1. Crie uma função do IAM e atribua permissões de operações em lote do S3 para executar. Esta etapa é necessária para todos os trabalhos de operações em lote do S3.
2. Configure operações em lote do S3 com bloqueio de objeto do S3 para executar.

Permita que a função faça o seguinte:

1. Execute o bloqueio de objeto no bucket do S3 que contém os objetos de destino nos quais você deseja que as operações em lote sejam executadas.
2. Leia o bucket do S3 no qual estão localizados o arquivo CSV de manifesto e os objetos.
3. Grave os resultados do trabalho de operações em lote do S3 no bucket de relatórios.

```
public void createObjectLockRole() {
    final String roleName = "bops-object-lock";

    final String trustPolicy = "{" +
```

```
"  \"Version\": \"2012-10-17\", " +
"  \"Statement\": [ " +
"    { " +
"      \"Effect\": \"Allow\", " +
"      \"Principal\": { " +
"        \"Service\": [ " +
"          \"batchoperations.s3.amazonaws.com\" +
"        ]" +
"      }, " +
"      \"Action\": \"sts:AssumeRole\" " +
"    } " +
"  ]" +
"}";
```

```
final String bopsPermissions = "{" +
"  \"Version\": \"2012-10-17\", " +
"  \"Statement\": [ " +
"    { " +
"      \"Effect\": \"Allow\", " +
"      \"Action\": \"s3:GetBucketObjectLockConfiguration\", " +
"      \"Resource\": [ " +
"        \"arn:aws:s3:::ManifestBucket\" +
"      ]" +
"    }, " +
"    { " +
"      \"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/*\" +
"      ]" +
"    }" +
"  ]" +
"}";
```

```
final AmazonIdentityManagement iam =
    AmazonIdentityManagementClientBuilder.defaultClient();
```

```
final CreateRoleRequest createRoleRequest = new CreateRoleRequest()
    .withAssumeRolePolicyDocument(bopsPermissions)
    .withRoleName(roleName);
```

```
final CreateRoleResult createRoleResult = iam.createRole(createRoleRequest);
```

```
final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
    .withPolicyDocument(bopsPermissions)
    .withPolicyName("bops-permissions")
    .withRoleName(roleName);
```

```
final PutRolePolicyResult putRolePolicyResult =
iam.putRolePolicy(putRolePolicyRequest);
}
```

Definir a retenção do Bloqueio de objetos usando o Operações em lote

O exemplo a seguir permite que a regra defina a retenção de bloqueio de objetos do S3 para seus objetos no bucket de manifestos.

Atualize a função para incluir permissões `s3:PutObjectRetention` para que você possa executar a retenção de bloqueio de objetos nos objetos no bucket.

Usar a AWS CLI

```
export AWS_PROFILE='aws-user'

read -d '' retention_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectRetention"
            ],
            "Resource": [
                "arn:aws:s3:::{ManifestBucket}/*"
            ]
        }
    ]
}
EOF

aws iam put-role-policy --role-name bops-objectlock --policy-name retention-permissions --policy-document "${retention_permissions}"
```

Usar o AWS SDK for Java

```
public void allowPutObjectRetention() {
    final String roleName = "bops-object-lock";

    final String retentionPermissions = "{" +
        "    \"Version\": \"2012-10-17\", " +
        "    \"Statement\": [ " +
        "        { " +
        "            \"Effect\": \"Allow\", " +
        "            \"Action\": [ " +
        "                \"s3:PutObjectRetention\" " +
        "            ], " +
        "            \"Resource\": [ " +
        "                \"arn:aws:s3:::ManifestBucket*\" " +
        "            ] " +
        "        } " +
        "    ] " +
    "}";

    final AmazonIdentityManagement iam =
        AmazonIdentityManagementClientBuilder.defaultClient();

    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
        .withPolicyDocument(retentionPermissions)
        .withPolicyName("retention-permissions")
        .withRoleName(roleName);
```

```
    final PutRolePolicyResult putRolePolicyResult =
    iam.putRolePolicy(putRolePolicyRequest);
}
```

Usar operações em lote do S3 com o modo de conformidade de retenção do bloqueio de objetos do S3

O exemplo a seguir baseia-se nos exemplos anteriores de criação de uma política de confiança e na definição de permissões de configuração do Operações em lote do S3 e Bloqueio de objetos do S3 em seus objetos. Este exemplo define o modo de retenção como COMPLIANCE e retain until date como 1º de janeiro de 2020. Ele cria um trabalho que visa objetos no bucket de manifesto e relata os resultados no bucket de relatórios identificado.

Usar a AWS CLI

Example Definir conformidade com menções em vários objetos

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/bops-objectlock'

read -d '' OPERATION <<EOF
{
    "S3PutObjectRetention": {
        "Retention": {
            "RetainUntilDate": "2025-01-01T00:00:00",
            "Mode": "COMPLIANCE"
        }
    }
}
EOF

read -d '' MANIFEST <<EOF
{
    "Spec": {
        "Format": "S3BatchOperations_CSV_20180820",
        "Fields": [
            "Bucket",
            "Key"
        ]
    },
    "Location": {
        "ObjectArn": "arn:aws:s3:::ManifestBucket/compliance-objects-manifest.csv",
        "ETag": "Your-manifest-ETag"
    }
}
EOF

read -d '' REPORT <<EOF
{
    "Bucket": "arn:aws:s3:::ReportBucket",
    "Format": "Report_CSV_20180820",
    "Enabled": true,
    "Prefix": "reports/compliance-objects-bops",
    "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
```

```
--account-id "${ACCOUNT_ID}" \
--manifest "${MANIFEST//$/\n}" \
--operation "${OPERATION//$/\n/}" \
--report "${REPORT//$/\n}" \
--priority 10 \
--role-arn "${ROLE_ARN}" \
--client-request-token "$(uuidgen)" \
--region "${AWS_DEFAULT_REGION}" \
--description "Set compliance retain-until to 1 Jul 2030";
```

Example Estender a **COMPLIANCE** do modo **retain until date** para 15 de janeiro de 2020

O exemplo a seguir estende COMPLIANCE do modo **retain until date** para 15 de janeiro de 2025.

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/bops-objectlock'

read -d '' OPERATION <<EOF
{
    "S3PutObjectRetention": {
        "Retention": {
            "RetainUntilDate": "2025-01-15T00:00:00",
            "Mode": "COMPLIANCE"
        }
    }
}
EOF

read -d '' MANIFEST <<EOF
{
    "Spec": {
        "Format": "S3BatchOperations_CSV_20180820",
        "Fields": [
            "Bucket",
            "Key"
        ],
        "Location": {
            "ObjectArn": "arn:aws:s3:::ManifestBucket/compliance-objects-manifest.csv",
            "ETag": "Your-manifest-ETag"
        }
    }
}
EOF

read -d '' REPORT <<EOF
{
    "Bucket": "arn:aws:s3:::ReportBucket",
    "Format": "Report_CSV_20180820",
    "Enabled": true,
    "Prefix": "reports/compliance-objects-bops",
    "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$/\n}" \
    --operation "${OPERATION//$/\n/}" \
    --report "${REPORT//$/\n}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
```

```
--client-request-token "$(uuidgen)" \
--region "${AWS_DEFAULT_REGION}" \
--description "Extend compliance retention to 15 Jan 2020";
```

Usar o AWS SDK for Java

Example Defina o modo de retenção como COMPLIANCE e a retenção até à data de 1º de janeiro de 2020.

```
public String createComplianceRetentionJob(final AWSS3ControlClient awss3ControlClient)
    throws ParseException {
    final String manifestObjectArn = "arn:aws:s3:::ManifestBucket/compliance-objects-manifest.csv";
    final String manifestObjectVersionId = "your-object-version-Id";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
        .withObjectArn(manifestObjectArn)
        .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
        new JobManifestSpec()
            .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
            .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
        .withLocation(manifestLocation)
        .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::ReportBucket";
    final String jobReportPrefix = "reports/compliance-objects-bops";

    final JobReport jobReport = new JobReport()
        .withEnabled(true)
        .withReportScope(JobReportScope.AllTasks)
        .withBucket(jobReportBucketArn)
        .withPrefix(jobReportPrefix)
        .withFormat(JobReportFormat.Report_CSV_20180820);

    final SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
    final Date janFirst = format.parse("01/01/2020");

    final JobOperation jobOperation = new JobOperation()
        .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
            .withRetention(new S3Retention()
                .withMode(S3ObjectLockRetentionMode.COMPLIANCE)
                .withRetainUntilDate(janFirst)));

    final String roleArn = "arn:aws:iam::123456789012:role/bops-object-lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
        .withAccountId("123456789012")
        .withDescription("Set compliance retain-until to 1 Jan 2020")
        .withManifest(manifestToPublicApi)
        .withOperation(jobOperation)
        .withPriority(priority)
        .withRoleArn(roleArn)
        .withReport(jobReport)
        .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
```

}

Example Estender a **COMPLIANCE** do modo **retain until date**

O exemplo a seguir estende a **COMPLIANCE** do modo **retain until date** para 15 de janeiro de 2020.

```
public String createExtendComplianceRetentionJob(final AWSS3ControlClient awss3ControlClient) throws ParseException {
    final String manifestObjectArn = "arn:aws:s3:::ManifestBucket/compliance-objects-manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
        .withObjectArn(manifestObjectArn)
        .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
        new JobManifestSpec()
            .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
            .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
        .withLocation(manifestLocation)
        .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::ReportBucket";
    final String jobReportPrefix = "reports/compliance-objects-bops";

    final JobReport jobReport = new JobReport()
        .withEnabled(true)
        .withReportScope(JobReportScope.AllTasks)
        .withBucket(jobReportBucketArn)
        .withPrefix(jobReportPrefix)
        .withFormat(JobReportFormat.Report_CSV_20180820);

    final SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
    final Date jan15th = format.parse("15/01/2020");

    final JobOperation jobOperation = new JobOperation()
        .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
            .withRetention(new S3Retention()
                .withMode(S3ObjectLockRetentionMode.COMPLIANCE)
                .withRetainUntilDate(jan15th)));

    final String roleArn = "arn:aws:iam::123456789012:role/bops-object-lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
        .withAccountId("123456789012")
        .withDescription("Extend compliance retention to 15 Jan 2020")
        .withManifest(manifestToPublicApi)
        .withOperation(jobOperation)
        .withPriority(priority)
        .withRoleArn(roleArn)
        .withReport(jobReport)
        .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Usar operações em lote do S3 com modo de governança de retenção de bloqueio de objetos do S3

O exemplo a seguir baseia-se nos exemplos anteriores de criação de uma política de confiança e na definição de permissões de configuração de operações em lote do S3 e bloqueio de objetos do S3. Ele mostra como aplicar a governança de retenção do Bloqueio de objetos do S3 com a `retain until date` de 30 de janeiro de 2025, em vários objetos. Ele cria um trabalho de operações em lote que usa o bucket de manifesto e relata os resultados no bucket de relatórios.

Usar a AWS CLI

Example Aplicar a governança de retenção do Bloqueio de objetos do S3 em vários objetos com a retenção até a data de 30 de janeiro de 2020

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/bops-objectlock'

read -d '' OPERATION <<EOF
{
    "S3PutObjectRetention": {
        "Retention": {
            "RetainUntilDate": "2025-01-30T00:00:00",
            "Mode": "GOVERNANCE"
        }
    }
}
EOF

read -d '' MANIFEST <<EOF
{
    "Spec": {
        "Format": "S3BatchOperations_CSV_20180820",
        "Fields": [
            "Bucket",
            "Key"
        ],
        "Location": {
            "ObjectArn": "arn:aws:s3:::ManifestBucket/governance-objects-manifest.csv",
            "ETag": "Your-manifest-ETag"
        }
    }
}
EOF

read -d '' REPORT <<EOF
{
    "Bucket": "arn:aws:s3:::ReportBucketT",
    "Format": "Report_CSV_20180820",
    "Enabled": true,
    "Prefix": "reports/governance-objects",
    "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST}://${'\n'}" \
    --operation "${OPERATION}://${'\n'}/}" \
    --report "${REPORT}://${'\n'}" \
```

```
--priority 10 \
--role-arn "${ROLE_ARN}" \
--client-request-token "$(uuidgen)" \
--region "${AWS_DEFAULT_REGION}" \
--description "Put governance retention";
```

Example Ignorar a governança de retenção em vários objetos

O exemplo a seguir baseia-se nos exemplos anteriores de criação de uma política de confiança e na definição de permissões de configuração de operações em lote do S3 e bloqueio de objetos do S3. Ele mostra como ignorar a governança de retenção em vários objetos e cria um trabalho de operações em lote que usa o bucket de manifestos e relata os resultados no bucket de relatórios.

```
export AWS_PROFILE='aws-user'

read -d '' bypass_governance_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:BypassGovernanceRetention"
            ],
            "Resource": [
                "arn:aws:s3:::${ManifestBucket}/*"
            ]
        }
    ]
}
EOF

aws iam put-role-policy --role-name bops-objectlock --policy-name bypass-governance-
permissions --policy-document "${bypass_governance_permissions}"

export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/bops-objectlock'

read -d '' OPERATION <<EOF
{
    "S3PutObjectRetention": {
        "BypassGovernanceRetention": true,
        "Retention": {}
    }
}
EOF

read -d '' MANIFEST <<EOF
{
    "Spec": {
        "Format": "S3BatchOperations_CSV_20180820",
        "Fields": [
            "Bucket",
            "Key"
        ],
        "Location": {
            "ObjectArn": "arn:aws:s3:::${ManifestBucket}/governance-objects-manifest.csv",
            "ETag": "Your-manifest-ETag"
        }
    }
}
```

```
EOF

read -d '' REPORT <<EOF
{
    "Bucket": "arn:aws:s3:::REPORT_BUCKET",
    "Format": "Report_CSV_20180820",
    "Enabled": true,
    "Prefix": "reports/bops-governance",
    "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Remove governance retention";
```

Usar o AWS SDK for Java

O exemplo a seguir baseia-se nos exemplos anteriores de criação de uma política de confiança e na definição de permissões de configuração de operações em lote do S3 e bloqueio de objetos do S3. Ele mostra como aplicar a governança de retenção de bloqueio de objetos do S3 com o `retain until date` definido como 30 de janeiro de 2020 em vários objetos. Ele cria um trabalho de operações em lote que usa o bucket de manifesto e relata os resultados no bucket de relatórios.

Example Aplicar a governança de retenção do Bloqueio de objetos do S3 em vários objetos com a retenção até a data de 30 de janeiro de 2020

```
public String createGovernanceRetentionJob(final AWSS3ControlClient awss3ControlClient)
    throws ParseException {
    final String manifestObjectArn = "arn:aws:s3:::ManifestBucket/governance-objects-manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
        .withObjectArn(manifestObjectArn)
        .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
        new JobManifestSpec()
            .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
            .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
        .withLocation(manifestLocation)
        .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::ReportBucket";
    final String jobReportPrefix = "reports/governance-objects";

    final JobReport jobReport = new JobReport()
        .withEnabled(true)
        .withReportScope(JobReportScope.AllTasks)
        .withBucket(jobReportBucketArn)
        .withPrefix(jobReportPrefix)
        .withFormat(JobReportFormat.Report_CSV_20180820);
```

```
final SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
final Date jan30th = format.parse("30/01/2020");

final JobOperation jobOperation = new JobOperation()
    .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
        .withRetention(new S3Retention()
            .withMode(S3ObjectLockRetentionMode.GOVERNANCE)
            .withRetainUntilDate(jan30th)));

final String roleArn = "arn:aws:iam::123456789012:role/bops-object-lock";
final Boolean requiresConfirmation = true;
final int priority = 10;

final CreateJobRequest request = new CreateJobRequest()
    .withAccountId("123456789012")
    .withDescription("Put governance retention")
    .withManifest(manifestToPublicApi)
    .withOperation(jobOperation)
    .withPriority(priority)
    .withRoleArn(roleArn)
    .withReport(jobReport)
    .withConfirmationRequired(requiresConfirmation);

final CreateJobResult result = awss3ControlClient.createJob(request);

return result.getJobId();
}
```

Example Ignorar a governança de retenção em vários objetos

O exemplo a seguir baseia-se nos exemplos anteriores de criação de uma política de confiança e na definição de permissões de configuração de operações em lote do S3 e bloqueio de objetos do S3. Ele mostra como ignorar a governança de retenção em vários objetos e cria um trabalho de operações em lote que usa o bucket de manifestos e relata os resultados no bucket de relatórios.

```
public void allowBypassGovernance() {
    final String roleName = "bops-object-lock";

    final String bypassGovernancePermissions = "{" +
        "    \"Version\": \"2012-10-17\", " +
        "    \"Statement\": [ " +
        "        { " +
        "            \"Effect\": \"Allow\", " +
        "            \"Action\": [ " +
        "                \"s3:BypassGovernanceRetention\" " +
        "            ], " +
        "            \"Resource\": [ " +
        "                \"arn:aws:s3:::ManifestBucket/*\" " +
        "            ] " +
        "        } " +
        "    ] " +
    "}";

    final AmazonIdentityManagement iam =
        AmazonIdentityManagementClientBuilder.defaultClient();

    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
        .withPolicyDocument(bypassGovernancePermissions)
        .withPolicyName("bypass-governance-permissions")
        .withRoleName(roleName);

    final PutRolePolicyResult putRolePolicyResult =
        iam.putRolePolicy(putRolePolicyRequest);
```

```
}

public String createRemoveGovernanceRetentionJob(final AWSS3ControlClient
awss3ControlClient) {
    final String manifestObjectArn = "arn:aws:s3:::ManifestBucket/governance-objects-manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
        .withObjectArn(manifestObjectArn)
        .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
        new JobManifestSpec()
            .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
            .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
        .withLocation(manifestLocation)
        .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::ReportBucket";
    final String jobReportPrefix = "reports/bops-governance";

    final JobReport jobReport = new JobReport()
        .withEnabled(true)
        .withReportScope(JobReportScope.AllTasks)
        .withBucket(jobReportBucketArn)
        .withPrefix(jobReportPrefix)
        .withFormat(JobReportFormat.Report_CSV_20180820);

    final JobOperation jobOperation = new JobOperation()
        .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
            .withRetention(new S3Retention()));

    final String roleArn = "arn:aws:iam::123456789012:role/bops-object-lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
        .withAccountId("123456789012")
        .withDescription("Remove governance retention")
        .withManifest(manifestToPublicApi)
        .withOperation(jobOperation)
        .withPriority(priority)
        .withRoleArn(roleArn)
        .withReport(jobReport)
        .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Usar operações em lote do S3 para desativar a retenção legal do bloqueio de objetos do S3

O exemplo a seguir baseia-se nos exemplos anteriores de criação de uma política de confiança e na definição de permissões de configuração de operações em lote do S3 e bloqueio de objetos do S3. Ele mostra como desativar a retenção legal de bloqueio de objetos em objetos usando operações em lote.

O exemplo primeiro atualiza a função para conceder permissões s3:PutObjectLegalHold, cria um trabalho de operações em lote que desativa (remove) a retenção legal dos objetos identificados no manifesto e, depois, cria um relatório.

Usar a AWS CLI

Example Atualiza a função para conceder permissões **s3:PutObjectLegalHold**

```
export AWS_PROFILE='aws-user'

read -d '' legal_hold_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectLegalHold"
            ],
            "Resource": [
                "arn:aws:s3:::ManifestBucket/*"
            ]
        }
    ]
}

EOF

aws iam put-role-policy --role-name bops-objectlock --policy-name legal-hold-permissions --policy-document "${legal_hold_permissions}"
```

Example Desativar retenção legal

O exemplo a seguir desativa a retenção legal.

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/bops-objectlock'

read -d '' OPERATION <<EOF
{
    "S3PutObjectLegalHold": {
        "LegalHold": {
            "Status": "OFF"
        }
    }
}
EOF

read -d '' MANIFEST <<EOF
{
    "Spec": {
        "Format": "S3BatchOperations_CSV_20180820",
        "Fields": [
            "Bucket",
            "Key"
        ]
    },
    "Location": {
        "ObjectArn": "arn:aws:s3:::ManifestBucket/legalhold-object-manifest.csv",
        "ETag": "Your-manifest-ETag"
    }
}
EOF

read -d '' REPORT <<EOF
{
```

```
"Bucket": "arn:aws:s3:::ReportBucket",  
"Format": "Report_CSV_20180820",  
"Enabled": true,  
"Prefix": "reports/legalhold-objects-bops",  
"ReportScope": "AllTasks"  
}  
EOF  
  
aws \  
    s3control create-job \  
    --account-id "${ACCOUNT_ID}" \  
    --manifest "${MANIFEST//$/\n}" \  
    --operation "${OPERATION//$/\n/}" \  
    --report "${REPORT//$/\n}" \  
    --priority 10 \  
    --role-arn "${ROLE_ARN}" \  
    --client-request-token "$(uuidgen)" \  
    --region "${AWS_DEFAULT_REGION}" \  
    --description "Turn off legal hold";
```

Usar o AWS SDK for Java

Example Atualiza a função para conceder permissões **s3:PutObjectLegalHold**

```
public void allowPutObjectLegalHold() {  
    final String roleName = "bops-object-lock";  
  
    final String legalHoldPermissions = "{" +  
        "    \"Version\": \"2012-10-17\", " +  
        "    \"Statement\": [ " +  
        "        {" +  
        "            \"Effect\": \"Allow\", " +  
        "            \"Action\": [ " +  
        "                \"s3:PutObjectLegalHold\" " +  
        "            ], " +  
        "            \"Resource\": [ " +  
        "                \"arn:aws:s3:::ManifestBucket/*\" " +  
        "            ]" +  
        "        }" +  
        "    ]" +  
    "};";  
  
    final AmazonIdentityManagement iam =  
        AmazonIdentityManagementClientBuilder.defaultClient();  
  
    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()  
        .withPolicyDocument(legalHoldPermissions)  
        .withPolicyName("legal-hold-permissions")  
        .withRoleName(roleName);  
  
    final PutRolePolicyResult putRolePolicyResult =  
        iam.putRolePolicy(putRolePolicyRequest);  
}
```

Example Desativar retenção legal

Use o exemplo abaixo se quiser desativar a retenção legal.

```
public String createLegalHoldOffJob(final AWSS3ControlClient awss3ControlClient) {  
    final String manifestObjectArn = "arn:aws:s3:::ManifestBucket/legalhold-object-manifest.csv";  
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";
```

```
final JobManifestLocation manifestLocation = new JobManifestLocation()
    .withObjectArn(manifestObjectArn)
    .withETag(manifestObjectVersionId);

final JobManifestSpec manifestSpec =
    new JobManifestSpec()
        .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
        .withFields("Bucket", "Key");

final JobManifest manifestToPublicApi = new JobManifest()
    .withLocation(manifestLocation)
    .withSpec(manifestSpec);

final String jobReportBucketArn = "arn:aws:s3:::ReportBucket";
final String jobReportPrefix = "reports/legalhold-objects-bops";

final JobReport jobReport = new JobReport()
    .withEnabled(true)
    .withReportScope(JobReportScope.AllTasks)
    .withBucket(jobReportBucketArn)
    .withPrefix(jobReportPrefix)
    .withFormat(JobReportFormat.Report_CSV_20180820);

final JobOperation jobOperation = new JobOperation()
    .withS3PutObjectLegalHold(new S3SetObjectLegalHoldOperation()
        .withLegalHold(new S3ObjectLockLegalHold()
            .withStatus(S3ObjectLockLegalHoldStatus.OFF)));
}

final String roleArn = "arn:aws:iam::123456789012:role/bops-object-lock";
final Boolean requiresConfirmation = true;
final int priority = 10;

final CreateJobRequest request = new CreateJobRequest()
    .withAccountId("123456789012")
    .withDescription("Turn off legal hold")
    .withManifest(manifestToPublicApi)
    .withOperation(jobOperation)
    .withPriority(priority)
    .withRoleArn(roleArn)
    .withReport(jobReport)
    .withConfirmationRequired(requiresConfirmation);

final CreateJobResult result = awss3ControlClient.createJob(request);

return result.getJobId();
}
```

Tutorial Operações em lote do S3

Os tutoriais a seguir apresentam procedimentos completos para algumas tarefas de operações em lote.

- Tutorial: Vídeos de transcodificação em lote com operações em lote do S3, AWS Lambda e o AWS Elemental MediaConvert (p. 70)

Monitorar o Amazon S3

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e a performance do Amazon S3 e das soluções da AWS. Recomendamos coletar dados de monitoramento de todas as partes de sua solução da AWS para que possa depurar mais facilmente uma falha multiponto, caso ocorra. Antes de começar a monitorar o Amazon S3, crie um plano de monitoramento que inclua as respostas para as 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?

Para obter mais informações sobre registro em log e monitoramento no Amazon S3, consulte os tópicos a seguir.

Tópicos

- [Ferramentas de monitoramento \(p. 1020\)](#)
- [Opções de registro em log para o Amazon S3 \(p. 1021\)](#)
- [Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail \(p. 1023\)](#)
- [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#)
- [Monitoramento de métricas com o Amazon CloudWatch \(p. 1067\)](#)
- [Notificações de eventos do Amazon S3 \(p. 1083\)](#)

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

É possível usar as seguintes ferramentas automatizadas de monitoramento para supervisionar o Amazon S3 e gerar relatórios quando algo estiver errado:

- [Amazon CloudWatch Alarms: 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 limite ao longo de vários períodos. A ação é uma notificação enviada para um tópico do Amazon Simple Notification Service \(Amazon SNS\) ou uma política do Amazon EC2 Auto Scaling. Os alarmes do CloudWatch não invocam ações só porque estão em um determinado estado. O estado deve ter sido alterado e mantido por uma quantidade especificada de períodos. Para obter mais informações, consulte \[Monitoramento de métricas com o Amazon CloudWatch \\(p. 1067\\)\]\(#\).](#)

- AWS CloudTrailMonitoramento de log: compartilhe arquivos de log entre contas, monitore os arquivos de log do CloudTrail em tempo real enviando-os para o CloudWatch Logs, escreva aplicações 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 [Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail \(p. 1023\)](#).

Ferramentas de monitoramento manual

Outra parte importante do monitoramento do Amazon S3 é o monitoramento manual dos itens que os alarmes do CloudWatch não abrangem. O Amazon S3, o CloudWatch, o Trusted Advisor e outros painéis do AWS Management Console fornecem uma visão rápida do estado do ambiente da AWS. Você pode permitir o registro em log de acesso 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 [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#).

- O painel do Amazon S3 mostra o seguinte:
 - Seus buckets, objetos e propriedades que contêm
- A página inicial do CloudWatch mostra o seguinte:
 - Alertas e status atual
 - Gráficos de alertas e recursos
 - Estado de integridade do serviço

Além disso, é possível 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.
- Pesquise e procure todas as métricas de recursos da AWS.
- Criar e editar alertas para ser notificado sobre problemas.
- O AWS Trusted Advisor pode ajudar a monitorar os recursos da AWS para melhorar a performance, 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 mais informações, consulte [AWS Trusted Advisor](#).

O Trusted Advisor tem essas verificações que se referem ao Amazon S3:

- Verificações da configuração de registro em 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 buckets do Amazon S3 que não têm versionamento habilitado ou têm versionamento suspenso.

Opções de registro em log para o Amazon S3

É possível registrar as ações que são realizadas por usuários, funções ou produtos da AWS em recursos do Amazon S3 e manter registros de log para fins de auditoria e conformidade. Para fazer isso, você pode usar o registro em log de acesso ao servidor, o registro em log do AWS CloudTrail ou uma combinação de ambos. Recomendamos o uso do AWS CloudTrail para registrar em log as ações no nível de objeto e de bucket para os recursos do Amazon S3. Para obter mais informações sobre cada opção, consulte as seguintes seções:

- [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#)
- [Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail \(p. 1023\)](#)

A tabela a seguir lista as principais propriedades de logs do AWS CloudTrail e de logs de acesso ao servidor do Amazon S3. Analise a tabela e as observações para garantir que o AWS CloudTrail atenda aos seus requisitos de segurança.

Propriedades de log	AWS CloudTrail	Logs do servidor do Amazon S3
Pode ser encaminhado para outros sistemas (CloudWatch Logs, CloudWatch Events)	Sim	
Entregue logs para mais de um destino (por exemplo, envie os mesmos logs para dois buckets diferentes)	Sim	
Ative logs para um subconjunto de objetos (prefixo)	Sim	
Entrega de logs entre contas (bucket de origem e de destino pertencentes a contas diferentes)	Sim	
Validação de integridade do arquivo de log usando assinatura digital/hashing	Sim	
Padrão/opção de criptografia para arquivos de log	Sim	
Operações de objeto (usando APIs do Amazon S3)	Sim	Sim
Operações de bucket (usando APIs do Amazon S3)	Sim	Sim
UI pesquisável para logs	Sim	
Campos para parâmetros de bloqueio de objetos, propriedades selecionadas do Amazon S3 para registros de log	Sim	
Campos de Object Size, Total Time, Turn-Around Time e HTTP Referer para registros de log		Sim
Transições, expirações e restaurações do ciclo de vida		Sim
Registro de chaves em uma operação de exclusão em lote		Sim
Falhas de autenticação ¹		Sim
Contas em que logs são entregues	Proprietário do bucket ² e solicitante	Somente proprietário do bucket
Performance and Cost	AWS CloudTrail	Amazon S3 Server Logs

Propriedades de log	AWS CloudTrail	Logs do servidor do Amazon S3
Preço	Os eventos de gerenciamento (primeira entrega) são gratuitos. Os eventos de dados incorrem em uma taxa, além do armazenamento de logs	Nenhum custo adicional além do armazenamento de logs
Velocidade da entrega de logs	Eventos de dados a cada 5 minutos. Eventos de gerenciamento a cada 15 minutos	Em algumas horas
Formato do log	JSON	Arquivo de log com registros delimitados por novas linhas e separados por espaços

Observações:

1. O CloudTrail não entrega logs para solicitações com falha de autenticação (nas quais as credenciais fornecidas não são válidas). No entanto, ele inclui logs para solicitações nas quais a autorização falha (`AccessDenied`) e as solicitações são feitas por usuários anônimos.
2. O proprietário do bucket do S3 recebe logs do CloudTrail somente se o conta possuir ou tiver acesso total ao objeto na solicitação. Para obter mais informações, consulte [Ações do nível de objetos em cenários entre contas \(p. 1028\)](#).

Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail

O Amazon S3 é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações tomadas por um usuário, uma função ou um serviço da AWS no Amazon S3. O CloudTrail captura um subconjunto de chamadas de API para o Amazon S3 como eventos, incluindo chamadas do console do Amazon S3 e chamadas de código para as 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 você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history (Histórico de eventos). Usando as informações coletadas pelo CloudTrail, é possível determinar 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 ativá-lo, consulte o [Guia do usuário do AWS CloudTrail](#).

Usar logs do CloudTrail com logs de acesso ao servidor do Amazon S3 e CloudWatch Logs

Os logs do AWS CloudTrail fornecem os registros das ações tomadas por um usuário, uma função ou um produto da AWS no Amazon S3, enquanto os logs de acesso do servidor do Amazon S3 oferecem registros das solicitações feitas a um bucket do S3. Para obter mais informações sobre como os diferentes registros funcionam e as propriedades, a performance e os custos deles, consulte [the section called "Opções de registro em log" \(p. 1021\)](#).

Você pode usar os logs do AWS CloudTrail em conjunto com os logs de acesso ao servidor do Amazon S3. Os logs do CloudTrail fornecem rastreamento detalhado de API para operações no nível de bucket e objeto do Amazon S3. Os logs de acesso ao servidor do Amazon S3 fornecem visibilidade em operações no nível de objeto em seus dados no Amazon S3. Para obter mais informações sobre logs de acesso ao servidor, consulte [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#).

Você também pode usar logs do CloudTrail junto com o CloudWatch para Amazon S3. A integração do CloudTrail com o CloudWatch Logs entrega as atividades de API no nível do bucket do S3 capturadas pelo CloudTrail a um stream de logs do CloudWatch no grupo de logs do CloudWatch especificado. 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 alarmes do CloudWatch para monitorar atividades específicas da API, consulte o [Guia do usuário do AWS CloudTrail](#). Para obter mais informações sobre como usar o CloudWatch com o Amazon S3, consulte [Monitoramento de métricas com o Amazon CloudWatch \(p. 1067\)](#).

Rastreamento do CloudTrail com chamadas de API SOAP do Amazon S3

O CloudTrail rastreia as chamadas de 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: Usar a API SOAP \(p. 1279\)](#).

Important

Os novos recursos do Amazon S3 não são compatíveis com o SOAP. Recomendamos usar a API REST ou os AWS SDKs.

Ações SOAP do Amazon S3 rastreadas pelo registro em log do CloudTrail

Nome da API SOAP	Nome do evento da API usado no log do CloudTrail
ListAllMyBuckets	ListBuckets
CreateBucket	CreateBucket
DeleteBucket	DeleteBucket
GetBucketAccessControlPolicy	GetBucketAcl
SetBucketAccessControlPolicy	PutBucketAcl
GetBucketLoggingStatus	GetBucketLogging
SetBucketLoggingStatus	PutBucketLogging

Para obter mais informações sobre o CloudTrail e o Amazon S3, consulte os seguintes tópicos:

Tópicos

- [Eventos do CloudTrail no Amazon S3 \(p. 1025\)](#)
- [Entradas de arquivo de log do CloudTrail para Amazon S3 e S3 on Outposts \(p. 1029\)](#)
- [Habilitar o log de eventos do CloudTrail para buckets e objetos do S3 \(p. 1033\)](#)
- [Identificar solicitações do Amazon S3 usando o CloudTrail \(p. 1035\)](#)

Eventos do CloudTrail no Amazon S3

O CloudTrail é habilitado em sua Conta da AWS quando ela é criada. Quando a atividade do evento compatível ocorrer no Amazon S3, ela será registrada em um evento do CloudTrail juntamente com outros eventos de serviços da AWS no Event history (Histórico de eventos). Você pode visualizar, pesquisar e baixar 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 de eventos em andamento na sua Conta da AWS, incluindo eventos do Amazon S3, crie uma trilha. Uma trilha permite que o CloudTrail entregue 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 da . A trilha registra em log eventos de todas as regiões na partição da AWS e entrega os arquivos de log para o bucket do Amazon S3 especificado por você. Além disso, é possível configurar outros serviços da AWS para analisar mais ainda mais e agir com base nos dados de eventos coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- [Criar uma trilha para a sua Conta da AWS](#)
- [Integrações de serviços da AWS com CloudTrail Logs](#)
- [Configurar notificações do Amazon SNS para o CloudTrail](#)
- [Recebimento de arquivos de log do CloudTrail de várias regiões](#)
- [Recebimento de 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-raiz ou de usuário 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 AWS service (Serviço da AWS).

Para obter mais informações, consulte o [Elemento userIdentity do CloudTrail](#).

Você pode armazenar arquivos de log no bucket pelo tempo que desejar, mas também pode definir regras do ciclo de vida do Amazon S3 para arquivar ou excluir os arquivos de log automaticamente. Por padrão, os arquivos de log são criptografados usando-se Server-Side Encryption (SSE - Criptografia do lado do servidor) do Amazon S3.

Como o CloudTrail captura solicitações feitas para o Amazon S3

Por padrão, o CloudTrail registra em log chamadas de API no nível de buckets do S3 que foram feitas nos últimos 90 dias, mas não registra em log solicitações feitas a objetos. As chamadas no nível de buckets incluem eventos como `CreateBucket`, `DeleteBucket`, `PutBucketLifecycle`, `PutBucketPolicy`, etc. Você pode ver eventos no nível do bucket no console do CloudTrail. No entanto, não é possível visualizar eventos de dados (chamadas no nível do objeto do Amazon S3). Você deve analisar ou consultar os logs do CloudTrail para eles.

Ações no nível da conta do Amazon S3 rastreadas pelo registro em log do CloudTrail

O CloudTrail registra ações no nível da conta. Os registros do Amazon S3 são gravados com outros registros de serviço da AWS em um arquivo de log. O CloudTrail determina quando criar e gravar em um novo arquivo de acordo com o período e o tamanho do arquivo.

As tabelas nesta seção listam as ações no nível da conta do Amazon S3 que são compatíveis com o registro do CloudTrail.

Ações de API no nível da conta do Amazon S3 rastreadas pelo registro em log do CloudTrail aparecem como os seguintes nomes de eventos:

- [DeleteAccountPublicAccessBlock](#)
- [GetAccountPublicAccessBlock](#)
- [PutAccountPublicAccessBlock](#)

Ações no nível do bucket do Amazon S3 rastreadas pelo registro em log do CloudTrail

Por padrão, o CloudTrail registra ações no nível do bucket. Os registros do Amazon S3 são gravados com outros registros de serviço da AWS em um arquivo de log. O CloudTrail determina quando criar e gravar em um novo arquivo de acordo com o período e o tamanho do arquivo.

As tabelas nesta seção listam as ações do nível do bucket do Amazon S3 que são compatíveis com o registro em log do CloudTrail.

Ações de API no nível do bucket do Amazon S3 rastreadas pelo registro em log do CloudTrail aparecerão como os seguintes nomes de eventos:

- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteBucketCors](#)
- [DeleteBucketEncryption](#)
- [DeleteBucketLifecycle](#)
- [DeleteBucketPolicy](#)
- [DeleteBucketReplication](#)
- [DeleteBucketTagging](#)
- [DeleteBucketPublicAccessBlock](#)
- [GetBucketAcl](#)
- [GetBucketCors](#)
- [GetBucketEncryption](#)
- [GetBucketLifecycle](#)
- [GetBucketLocation](#)
- [GetBucketLogging](#)
- [GetBucketNotification](#)
- [GetObjectLockConfiguration](#)
- [GetBucketPolicy](#)
- [GetBucketReplication](#)
- [GetBucketRequestPayment](#)
- [GetBucketTagging](#)
- [GetBucketVersioning](#)
- [GetBucketWebsite](#)
- [GetBucketPublicAccessBlock](#)

- [ListBuckets](#)
- [PutBucketAcl](#)
- [PutBucketCors](#)
- [PutBucketEncryption](#)
- [PutBucketLifecycle](#)
- [PutBucketLogging](#)
- [PutBucketNotification](#)
- [PutBucketPolicy](#)
- [PutBucketReplication](#)
- [PutBucketRequestPayment](#)
- [PutBucketTagging](#)
- [PutBucketVersioning](#)
- [PutBucketWebsite](#)
- [PutBucketPublicAccessBlock](#)

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 em log do CloudTrail, pois verifica a configuração CORS de um bucket.

Ações no nível de objetos do Amazon S3 rastreadas pelo registro em log do AWS CloudTrail

Você também pode obter logs do CloudTrail para ações do Amazon S3 no nível do objeto. Para fazer isso, ative eventos de dados para o bucket do S3 ou todos os buckets em sua conta. 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 [Habilitar o log de eventos do CloudTrail para buckets e objetos do S3 \(p. 1033\)](#) e [Log de eventos de dados para trilhas](#) no Guia do usuário do AWS CloudTrail.

As seguintes ações de API no nível do objeto são registradas como eventos do CloudTrail:

- [AbortMultipartUpload](#)
- [CompleteMultipartUpload](#)
- [DeleteObjects](#)
- [DeleteObject](#)
- [GetObject](#)
- [GetObjectAcl](#)
- [GetObjectAttributes](#)
- [GetObjectTagging](#)
- [HeadObject](#)
- [CreateMultipartUpload](#)
- [ListParts](#)
- [PostObject](#)
- [RestoreObject](#)
- [PutObject](#)
- [PutObjectAcl](#)

- [PutObjectTagging](#)
- [CopyObject](#)
- [UploadPart](#)
- [UploadPartCopy](#)

Além dessas operações, você pode usar as seguintes operações do nível do bucket para obter logs do CloudTrail como ações do nível do objeto do Amazon S3 sob certas condições:

- [GET Bucket Object \(List Objects\)](#)(GET Bucket Object (Listar objetos) versão 2: selecione um prefixo especificado na trilha.
- [GET Bucket Object Versions \(List Object Versions\)](#) (GET Bucket Object Versions (Listar versões do objeto): 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.

Note

O CloudTrail não registra em log nomes de chaves para as chaves excluídas usando a operação Excluir vários objetos.

Ações do nível de objetos em cenários entre contas

Os seguintes são casos de uso especiais que envolvem chamadas de API do nível do objeto em cenários entre contas e como os logs do CloudTrail são relatados. O CloudTrail entrega sempre logs ao solicitante (quem fez a chamada da 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.
- A conta C é proprietária do objeto. Pode ser a mesma conta que a conta A.

Note

O CloudTrail entrega sempre os logs de acesso da API do nível do objeto ao solicitante (conta B). Além disso, o CloudTrail também entrega os mesmos logs ao proprietário do bucket (conta A) somente se o proprietário do bucket (conta C) tiver permissões para as mesmas ações da API sobre esse objeto. Caso contrário, o proprietário do bucket deverá obter permissões, via ACL do objeto, para obter os logs da API em nível de 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 mais informações sobre ACLs, consulte [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#).

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.

Entradas de arquivo de log do CloudTrail para Amazon S3 e S3 on Outposts

Uma trilha é uma configuração que permite a entrega de eventos como registros de log a um bucket do Amazon S3 especificado. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma solicitação única de qualquer fonte. Isso inclui informações sobre a ação solicitada, a data e hora da ação, os parâmetros de solicitação, e assim por diante. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada das chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

Para obter mais informações, veja os exemplos a seguir:

Tópicos

- [Exemplo: entrada de arquivo de log do CloudTrail para o Amazon S3 \(p. 1029\)](#)
- [Exemplo: entradas de arquivo de log do Amazon S3 no Outposts \(p. 1031\)](#)

Exemplo: entrada de arquivo de log do CloudTrail para o Amazon S3

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra as ações [Serviço GET](#), [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": "2019-02-01T03:18:19Z",  
            "eventSource": "s3.amazonaws.com",  
            "eventName": "ListBuckets",  
            "awsRegion": "us-west-2",  
            "sourceIPAddress": "127.0.0.1",  
            "userAgent": "[ ]",  
            "requestParameters": {  
                "host": [  
                    "s3.us-west-2.amazonaws.com"  
                ]  
            },  
            "responseElements": {}  
        }  
    ]  
}
```

```
        "responseElements": null,
        "additionalEventData": {
            "SignatureVersion": "SigV2",
            "AuthenticationMethod": "QueryString"
        },
        "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": "2019-02-01T03:22:33Z",
        "eventSource": "s3.amazonaws.com",
        "eventName": "PutBucketAcl",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "",
        "userAgent": "[ ]",
        "requestParameters": {
            "bucketName": "",
            "AccessControlPolicy": {
                "AccessControlList": {
                    "Grant": {
                        "Grantee": {
                            "xsi:type": "CanonicalUser",
                            "xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
                            "ID": "d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
                        },
                        "Permission": "FULL_CONTROL"
                    }
                },
                "xmlns": "http://s3.amazonaws.com/doc/2006-03-01/",
                "Owner": {
                    "ID": "d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
                }
            }
        },
        "host": [
            "s3.us-west-2.amazonaws.com"
        ],
        "acl": [
            ""
        ]
    },
    "responseElements": null,
    "additionalEventData": {
        "SignatureVersion": "SigV4",
        "CipherSuite": "ECDHE-RSA-AES128-SHA",
        "AuthenticationMethod": "AuthHeader"
    },
    "requestID": "BD8798EACDD16751",
    "eventID": "607b9532-1423-41c7-b048-ec2641693c47",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
},
{
    "eventVersion": "1.03",
    "userIdentity": {
```

```
        "type": "IAMUser",
        "principalId": "111122223333",
        "arn": "arn:aws:iam::111122223333:user/myUserName",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "myUserName"
    },
    "eventTime": "2019-02-01T03:26:37Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "GetBucketVersioning",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "",
    "userAgent": "[ ]",
    "requestParameters": {
        "host": [
            "s3.us-west-2.amazonaws.com"
        ],
        "bucketName": "DOC-EXAMPLE-BUCKET1",
        "versioning": [
            ""
        ]
    },
    "responseElements": null,
    "additionalEventData": {
        "SignatureVersion": "SigV4",
        "CipherSuite": "ECDHE-RSA-AES128-SHA",
        "AuthenticationMethod": "AuthHeader",
    },
    "requestID": "07D681279BD94AED",
    "eventID": "f2b287f3-0df1-4961-a2f4-c4bdfed47657",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
]
```

Exemplo: entradas de arquivo de log do Amazon S3 no Outposts

Os eventos de gerenciamento do Amazon S3 no Outposts estão disponíveis por meio do AWS CloudTrail. Para obter mais informações, consulte [Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail \(p. 1023\)](#). Além disso, opcionalmente é possível [habilitar o registro em log para eventos de dados no AWS CloudTrail](#).

Uma trilha é uma configuração que permite a entrega de eventos como registros de log a um bucket do S3 em uma região que você especifica. Os logs do CloudTrail para buckets do Outposts incluem um novo campo `edgeDeviceDetails`, que identifica o Outpost em que o bucket especificado está localizado.

Campos de log adicionais incluem a ação solicitada, a data e a hora da ação e os parâmetros de solicitação. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada das chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `PutObject` em `s3-outposts`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "111122223333",
        "arn": "arn:aws:iam::111122223333:user/yourUserName",
        "accountId": "222222222222",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```
        "userName": "yourUserName"
    },
    "eventTime": "2020-11-30T15:44:33Z",
    "eventSource": "s3-outposts.amazonaws.com",
    "eventName": "PutObject",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "26.29.66.20",
    "userAgent": "aws-cli/1.18.39 Python/3.4.10 Darwin/18.7.0 botocore/1.15.39",
    "requestParameters": {
        "expires": "Wed, 21 Oct 2020 07:28:00 GMT",
        "Content-Language": "english",
        "x-amz-server-side-encryption-customer-key-MD5": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "ObjectCannedACL": "BucketOwnerFullControl",
        "x-amz-server-side-encryption": "Aes256",
        "Content-Encoding": "gzip",
        "Content-Length": "10",
        "Cache-Control": "no-cache",
        "Content-Type": "text/html; charset=UTF-8",
        "Content-Disposition": "attachment",
        "Content-MD5": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
        "x-amz-storage-class": "Outposts",
        "x-amz-server-side-encryption-customer-algorithm": "Aes256",
        "bucketName": "DOC-EXAMPLE-BUCKET1",
        "Key": "path/upload.sh"
    },
    "responseElements": {
        "x-amz-server-side-encryption-customer-key-MD5": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "x-amz-server-side-encryption": "Aes256",
        "x-amz-version-id": "001",
        "x-amz-server-side-encryption-customer-algorithm": "Aes256",
        "ETag": "d41d8cd98f00b204e9800998ecf8427f"
    },
    "additionalEventData": {
        "CipherSuite": "ECDHE-RSA-AES128-SHA",
        "bytesTransferredIn": 10,
        "x-amz-id-2": "29xXQBV2O+xOHKItvzY1suLv1i6A52E0zOX159fpfsItYd58JhXwKxXAXI4IQkp6",
        "SignatureVersion": "SigV4",
        "bytesTransferredOut": 20,
        "AuthenticationMethod": "AuthHeader"
    },
    "requestID": "8E96D972160306FA",
    "eventID": "ee3b4e0c-ab12-459b-9998-0a5a6f2e4015",
    "readOnly": false,
    "resources": [
        {
            "accountId": "222222222222",
            "type": "AWS::S3Outposts::Object",
            "ARN": "arn:aws:s3-outposts:us-east-1:YYYY:outpost/op-01ac5d28a6a232904/bucket/path/upload.sh"
        },
        {
            "accountId": "222222222222",
            "type": "AWS::S3Outposts::Bucket",
            "ARN": "arn:aws:s3-outposts:us-east-1:YYYY:outpost/op-01ac5d28a6a232904/bucket/"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "444455556666",
    "sharedEventID": "02759a4c-c040-4758-b84b-7cbaaf17747a",
    "edgeDeviceDetails": {
        "type": "outposts",
        "deviceID": "op-01ac5d28a6a232904"
    }
},
```

```
    "eventCategory": "Data"  
}
```

Habilitar o log de eventos do CloudTrail para buckets e objetos do S3

Você pode usar eventos de dados do CloudTrail para obter informações sobre solicitações no nível do bucket e do objeto no Amazon S3. Para habilitar eventos de dados do CloudTrail para todos os seus buckets ou para uma lista de buckets específicos, você deve [criar uma trilha manualmente no CloudTrail](#).

Note

- A configuração padrão para o CloudTrail é encontrar somente eventos de gerenciamento. Verifique se os eventos de dados estão habilitados para sua conta.
- Com um bucket do S3 que está gerando uma workload elevada, você poderia rapidamente gerar milhares de logs em um curto período. Esteja atento a quanto tempo você escolhe para habilitar eventos de dados do CloudTrail para um bucket ocupado.

O CloudTrail armazena logs de eventos de dados do Amazon S3 em um bucket do S3 de sua escolha. Você deve considerar a possibilidade de usar um bucket em uma Conta da AWS separada para organizar melhor os eventos de vários buckets que você venha a ter em um local central para facilitar a consulta e a análise. O AWS Organizations facilita a criação de uma Conta da AWS vinculada à conta que é proprietária do bucket que está sendo monitorado. Para obter mais informações, consulte [O que é o AWS Organizations](#) no Guia do usuário do AWS Organizations.

Ao criar uma trilha no CloudTrail, na seção de eventos de dados, você pode marcar a caixa de seleção **Select all S3 buckets in your account** (Selecionar todos os buckets do S3 em sua conta) para registrar em log todos os eventos no nível de objetos.

Note

- É uma prática recomendada criar uma política de ciclo de vida para o bucket de evento de dados do AWS CloudTrail. Configure a política de ciclo de vida para remover periodicamente arquivos de log após o período que você acredita ser necessário auditá-los. Fazer isso reduz a quantidade de dados que o Athena analisa para cada consulta. Para obter mais informações, consulte [Definir a configuração do ciclo de vida em um bucket \(p. 753\)](#).
- Para obter mais informações sobre o formato do registro em log, consulte [Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail \(p. 1023\)](#).
- Para obter exemplos de como consultar logs do CloudTrail, consulte a publicação do Blog sobre big data da AWS [Análise da segurança, compatibilidade e atividade operacional usando o AWS CloudTrail e o Amazon Athena](#).

Habilitar o registro em log de objetos em um bucket usando o console

Você pode usar o console do Amazon S3 para configurar uma trilha do AWS CloudTrail e registrar em log eventos de dados de objetos em um bucket do S3. O CloudTrail oferece suporte ao registro em log de operações de API no nível do objeto do Amazon S3, como `GetObject`, `DeleteObject` e `PutObject`. Esses eventos são chamados de eventos de dados.

Por padrão, as trilhas do CloudTrail não registram em log eventos de dados, mas é possível configurar as trilhas para registrar eventos de dados para buckets do S3 que você especificar, ou para registrar eventos

de dados para todos os buckets do Amazon S3 em sua Conta da AWS. Para obter mais informações, consulte [Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail \(p. 1023\)](#).

O CloudTrail não preenche eventos de dados no histórico de eventos do CloudTrail. Além disso, nem todas as ações no nível do bucket são preenchidas no histórico de eventos do CloudTrail. Para obter mais informações sobre ações de API no nível do bucket do Amazon S3 monitoradas pelo registro em log do CloudTrail, consulte [Ações no nível do bucket do Amazon S3 rastreadas pelo registro em log do CloudTrail \(p. 1026\)](#). Para obter mais informações sobre como consultar os logs do CloudTrail, consulte o artigo da Central de Conhecimento da AWS que fala sobre o [uso de padrões de filtro do Amazon CloudWatch Logs e o Amazon Athena para consultar logs do CloudTrail](#).

Para configurar uma trilha para registrar em log eventos de dados para um bucket do S3, você pode usar o console do AWS CloudTrail ou o console do Amazon S3. Se você estiver configurando uma trilha para registrar em log eventos de dados para todos os buckets do Amazon S3 na sua Conta da AWS, será mais fácil usar o console do CloudTrail. Para obter informações sobre como usar o console do CloudTrail para configurar uma trilha para registrar em log eventos de dados do S3, consulte [Eventos de dados](#) no Guia do usuário do AWS CloudTrail.

Important

Há cobranças adicionais para eventos de dados. Para obter mais informações, consulte [Definição de preço do AWS CloudTrail](#).

O procedimento a seguir mostra como usar o console do Amazon S3 para configurar uma trilha do CloudTrail para registrar em log eventos de dados para um bucket do S3.

Para habilitar registro de eventos de dados do CloudTrail para um bucket do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket.
3. Escolha Properties (Propriedades).
4. Em AWS CloudTrail data events (Eventos de dados do AWS CloudTrail), selecione Configure in CloudTrail (Configurar no CloudTrail).

Você pode criar uma nova trilha do CloudTrail ou reutilizar uma trilha existente e configurar eventos de dados do Amazon S3 para serem registrados em sua trilha. Para obter informações sobre como criar trilhas no console do CloudTrail, consulte [Criação e atualização de uma trilha com o console](#) no Guia do usuário do AWS CloudTrail. Para obter informações sobre como configurar o registro em log de eventos de dados do Amazon S3 no console do CloudTrail, consulte [Registro em log de eventos de dados para objetos do Amazon S3](#) no Guia do usuário do AWS CloudTrail.

Note

Se você usar o console do CloudTrail ou o console do Amazon S3 para configurar uma trilha para registro em log de eventos de dados para um bucket do S3, o console do Amazon S3 mostra que o registro está habilitado no nível do objeto para o bucket.

Para desabilitar o registro de eventos de dados do CloudTrail para objetos em um bucket do S3

- Para desabilitar o registro em log no nível do objeto para o bucket, abra o console do CloudTrail e remova o nome do bucket de Data events (Eventos de dados) da trilha.

Para obter informações sobre como habilitar o registro no nível do objeto ao criar um bucket do S3, consulte [Criação de um bucket \(p. 129\)](#).

Para obter mais informações sobre o registro em log do CloudTrail com buckets do S3, consulte os seguintes tópicos:

- Visualização das propriedades de um bucket do S3 (p. 134)
- Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail (p. 1023)
- Como trabalhar com arquivos de log do CloudTrail no Guia do usuário do AWS CloudTrail

Identificar solicitações do Amazon S3 usando o CloudTrail

No Amazon S3, você pode identificar solicitações usando um log de eventos do AWS CloudTrail. O AWS CloudTrail é a maneira preferida de identificar solicitações do Amazon S3, mas se você estiver usando logs de acesso ao servidor do Amazon S3, consulte [the section called “Identificar solicitações do S3” \(p. 1063\)](#).

Tópicos

- [Identificar solicitações feitas ao Amazon S3 em um log do CloudTrail \(p. 1035\)](#)
- [Identificar solicitações do Amazon S3 Signature versão 2 usando o CloudTrail \(p. 1037\)](#)
- [Identificar o acesso a objetos do S3 usando o CloudTrail \(p. 1039\)](#)

Identificar solicitações feitas ao Amazon S3 em um log do CloudTrail

Eventos registrados pelo CloudTrail são armazenados como objetos JSON compactados no formato gzip no bucket do S3. Para encontrar solicitações com eficiência, você deve usar um serviço como o Amazon Athena para indexar e consultar os logs do CloudTrail. Para obter mais informações sobre o CloudTrail e o Athena, consulte [Consulta de logs do AWS CloudTrail](#) no Guia do usuário do Amazon Athena.

Usar o Athena com logs do CloudTrail

Após configurar o CloudTrail para entregar eventos a um bucket, você deve começar a ver objetos irem para seu bucket de destino no console do Amazon S3. Eles são formatados da seguinte maneira:

`s3://myawsexamplebucket1/AWSLogs/111122223333/CloudTrail/Region/yyyy/mm/dd`

Example : Use o Athena para consultar logs de eventos do CloudTrail para solicitações específicas

Localize seus logs de eventos do CloudTrail:

`s3://myawsexamplebucket1/AWSLogs/111122223333/CloudTrail/us-east-2/2019/04/14`

Com os logs de eventos do CloudTrail, agora você pode criar um banco de dados e uma tabela do Athena para consultá-los da seguinte forma:

1. Abra o console do Athena em <https://console.aws.amazon.com/athena/>.
2. Altere a Região da AWS para que seja a mesma do bucket do S3 de destino do CloudTrail.
3. Na janela de consulta, crie um banco de dados do Athena para seus eventos do CloudTrail.

```
CREATE DATABASE s3_cLOUDTRAIL_EVENTS_DB
```

4. Use a consulta a seguir para criar uma tabela para todos os eventos do CloudTrail no bucket. Altere o nome do bucket de `CloudTrail_myawsexamplebucket1` para o nome de seu bucket. Além disso, forneça o `AWS_account_ID` CloudTrail que é usado no bucket.

```
CREATE EXTERNAL TABLE s3_cLOUDTRAIL_EVENTS_DB.cloudtrail_myawsexamplebucket1_table(
    eventversion STRING,
    useridentity STRUCT<
        type:STRING,
        principalid:STRING,
        arn:STRING,
        accountid:STRING,
        invokedby:STRING,
        accesskeyid:STRING,
        userName:STRING,
        sessioncontext:STRUCT<
            attributes:STRUCT<
                mfaauthenticated:STRING,
                creationdate:STRING>,
            sessionissuer:STRUCT<
                type:STRING,
                principalId:STRING,
                arn:STRING,
                accountId:STRING,
                userName:STRING>
        >,
        eventtime STRING,
        eventsource STRING,
        eventname STRING,
        awsregion STRING,
        sourceipaddress STRING,
        useragent STRING,
        errorcode STRING,
        errormessage STRING,
        requestparameters STRING,
        responseelements STRING,
        additionaleventdata STRING,
        requestid STRING,
        eventid STRING,
        resources ARRAY<STRUCT<
            ARN:STRING,
            accountId:STRING,
            type:STRING>>,
        eventtype STRING,
        apiversion STRING,
        readonly STRING,
        recipientaccountid STRING,
        serviceeventdetails STRING,
        sharedeventid STRING,
        vpcendpointid STRING
    >
)
ROW FORMAT SERDE 'com.amazon.emr.hive.serde.CloudTrailSerde'
STORED AS INPUTFORMAT 'com.amazon.emr.cloudtrail.CloudTrailInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://myawsexamplebucket1/AWSLogs/111122223333/';
```

5. Teste o Athena para garantir que a consulta funcione.

```
SELECT * FROM s3_cLOUDTRAIL_EVENTS_DB.cloudtrail_myawsexamplebucket1_table
WHERE eventsource='s3.amazonaws.com'
LIMIT 2;
```

Identificar solicitações do Amazon S3 Signature versão 2 usando o CloudTrail

Você pode usar um log de eventos do CloudTrail para identificar qual versão de assinatura de API foi usada para assinar uma solicitação no Amazon S3. Esse recurso é importante, pois o suporte ao Signature versão 2 será desativado (suspenso). Depois disso, o Amazon S3 não aceitará mais solicitações que usam o Signature versão 2 e todas as solicitações deverão usar assinaturas do Signature versão 4.

É altamente recomendável que você use o CloudTrail para ajudar a determinar se algum dos seus fluxos de trabalho está usando o Signature versão 2. Corrija-os atualizando suas bibliotecas e o código para usar o Signature versão 4 para evitar qualquer impacto em seus negócios.

Para obter mais informações, consulte [Anúncio: o AWS CloudTrail para Amazon S3 adiciona novos campos para auditoria segurança aprimorada](#) nos fóruns de discussão da AWS.

Note

Os eventos do CloudTrail para o Amazon S3 incluem a versão do Signature nos detalhes da solicitação sob o nome de chave de “`additionalEventData`”. Para encontrar a versão da assinatura em solicitações feitas para objetos no Amazon S3, como GETs, PUTs e DELETEs, você deve habilitar a opção de eventos de dados do CloudTrail (que fica desativada por padrão).

O AWS CloudTrail é o método preferencial para identificar solicitações do Signature versão 2. Se você estiver usando logs de acesso ao servidor do Amazon S3, consulte [Identificar solicitações do Signature versão 2 usando logs de acesso do Amazon S3 \(p. 1066\)](#).

Tópicos

- [Exemplos de consulta do Athena para identificar solicitações do Amazon S3 Signature versão 2 \(p. 1037\)](#)
- [Particionar dados do Signature versão 2 \(p. 1038\)](#)

Exemplos de consulta do Athena para identificar solicitações do Amazon S3 Signature versão 2

Example : Selecione todos os eventos do Signature versão 2 e imprima apenas EventTime, S3 action, Request_Parameters, Region, SourceIP e UserAgent

Na consulta do Athena a seguir, substitua `<s3_cloudtrail_events_db.cloudtrail_myawsexamplebucket1_table>` pelos detalhes do Athena e aumente ou remova o limite, conforme necessário.

```
SELECT EventTime, EventName as S3_Action, requestParameters as Request_Parameters,
awsregion as AWS_Region, sourceipaddress as Source_IP, useragent as User_Agent
FROM s3_cloudtrail_events_db.cloudtrail_myawsexamplebucket1_table
WHERE eventsource='s3.amazonaws.com'
AND json_extract_scalar(additionalEventData, '$.SignatureVersion')='SigV2'
LIMIT 10;
```

Example : Selecione todos os solicitantes que estão enviando tráfego do Signature versão 2

```
SELECT useridentity.arn, Count(requestid) as RequestCount
```

```
FROM s3_cloudtrail_events_db.cloudtrail_myawsexamplebucket1_table
WHERE eventsource='s3.amazonaws.com'
    and json_extract_scalar(additionalEventData, '$.SignatureVersion')='SigV2'
Group by useridentity.arn
```

Particionar dados do Signature versão 2

Se você tiver uma grande quantidade de dados para consultar, poderá reduzir os custos e o tempo de execução do Athena criando uma tabela particionada.

Para fazer isso, crie uma tabela com partições conforme indicado a seguir.

```
CREATE EXTERNAL TABLE
s3_cloudtrail_events_db.cloudtrail_myawsexamplebucket1_table_partitioned(
    eventversion STRING,
    userIdentity STRUCT<
        type:STRING,
        principalId:STRING,
        arn:STRING,
        accountId:STRING,
        invokedBy:STRING,
        accessKeyId:STRING,
        userName:STRING,
        sessionContext:STRUCT<
            attributes:STRUCT<
                mfaAuthenticated:STRING,
                creationDate:STRING>,
            sessionIssuer:STRUCT<
                type:STRING,
                principalId:STRING,
                arn:STRING,
                accountId:STRING,
                userName:STRING>
        >
    >,
    eventTime STRING,
    eventSource STRING,
    eventName STRING,
    awsRegion STRING,
    sourceIpAddress STRING,
    userAgent STRING,
    errorCode STRING,
    errorMessage STRING,
    requestParameters STRING,
    responseElements STRING,
    additionalEventData STRING,
    requestId STRING,
    eventId STRING,
    resources ARRAY<STRUCT<ARN:STRING,accountId: STRING,type:STRING>>,
    eventType STRING,
    apiVersion STRING,
    readOnly STRING,
    recipientAccountId STRING,
    serviceEventDetails STRING,
    sharedEventId STRING,
    vpcEndpointId STRING
)
PARTITIONED BY (region string, year string, month string, day string)
ROW FORMAT SERDE 'com.amazon.emr.hive.serde.CloudTrailSerde'
STORED AS INPUTFORMAT 'com.amazon.emr.cloudtrail.CloudTrailInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://myawsexamplebucket1/AWSLogs/111122223333/';
```

Em seguida, crie as partições individualmente. Não é possível obter resultados de datas que não foram criadas.

```
ALTER TABLE s3_cloudtrail_events_db.cloudtrail_myawsexamplebucket1_table_partitioned ADD
    PARTITION (region= 'us-east-1', year= '2019', month= '02', day= '19') LOCATION 's3://
myawsexamplebucket1/AWSLogs/111122223333/CloudTrail/us-east-1/2019/02/19/'
    PARTITION (region= 'us-west-1', year= '2019', month= '02', day= '19') LOCATION 's3://
myawsexamplebucket1/AWSLogs/111122223333/CloudTrail/us-west-1/2019/02/19/'
    PARTITION (region= 'us-west-2', year= '2019', month= '02', day= '19') LOCATION 's3://
myawsexamplebucket1/AWSLogs/111122223333/CloudTrail/us-west-2/2019/02/19/'
    PARTITION (region= 'ap-southeast-1', year= '2019', month= '02', day= '19') LOCATION
's3://myawsexamplebucket1/AWSLogs/111122223333/CloudTrail/ap-southeast-1/2019/02/19/'
    PARTITION (region= 'ap-southeast-2', year= '2019', month= '02', day= '19') LOCATION
's3://myawsexamplebucket1/AWSLogs/111122223333/CloudTrail/ap-southeast-2/2019/02/19/'
    PARTITION (region= 'ap-northeast-1', year= '2019', month= '02', day= '19') LOCATION
's3://myawsexamplebucket1/AWSLogs/111122223333/CloudTrail/ap-northeast-1/2019/02/19/'
    PARTITION (region= 'eu-west-1', year= '2019', month= '02', day= '19') LOCATION 's3://
myawsexamplebucket1/AWSLogs/111122223333/CloudTrail/eu-west-1/2019/02/19/'
    PARTITION (region= 'sa-east-1', year= '2019', month= '02', day= '19') LOCATION 's3://
myawsexamplebucket1/AWSLogs/111122223333/CloudTrail/sa-east-1/2019/02/19/';
```

Você pode, então, fazer a solicitação com base nessas partições e não é preciso carregar todo o bucket.

```
SELECT useridentity.arn,
Count(requestid) AS RequestCount
FROM s3_cloudtrail_events_db.cloudtrail_myawsexamplebucket1_table_partitioned
WHERE eventsource='s3.amazonaws.com'
AND json_extract_scalar(additionalEventData, '$.SignatureVersion')='SigV2'
AND region='us-east-1'
AND year='2019'
AND month='02'
AND day='19'
Group by useridentity.arn
```

Identificar o acesso a objetos do S3 usando o CloudTrail

Você pode usar seu log de eventos do AWS CloudTrail para identificar solicitações de acesso a objetos do Amazon S3 para eventos de dados como `GetObject`, `DeleteObject` e `PutObject` e descobrir informações adicionais sobre essas solicitações.

O exemplo a seguir mostra como obter todas as solicitações do objeto PUT para o Amazon S3 com o log de eventos do AWS CloudTrail.

Tópicos

- [Exemplos de consulta do Athena para identificar solicitações de acesso a objetos do Amazon S3 \(p. 1039\)](#)

Exemplos de consulta do Athena para identificar solicitações de acesso a objetos do Amazon S3

Nos exemplos de consultas do Athena a seguir, substitua `<s3_cloudtrail_events_db.cloudtrail_myawsexamplebucket1_table>` pelos detalhes do Athena e modifique o intervalo de datas conforme necessário.

Example : Selecione todos os eventos que têm solicitações de acesso ao objeto PUT e imprima somente EventTime, EventSource, SourceIP, UserAgent, BucketName, object, e UserARN

```
SELECT
    eventTime,
    eventName,
    eventSource,
    sourceIpAddress,
    userAgent,
    json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
    json_extract_scalar(requestParameters, '$.key') as object,
    userIdentity.arn as userArn
FROM
    s3_cLOUDTRAIL_EVENTS_DB.cloudtrail_myawsexamplebucket_table
WHERE
    eventName = 'PutObject'
    AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example : Selecione todos os eventos que têm solicitações de acesso ao objeto GET e imprima somente EventTime, EventSource, SourceIP, UserAgent, BucketName, object e UserARN

```
SELECT
    eventTime,
    eventName,
    eventSource,
    sourceIpAddress,
    userAgent,
    json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
    json_extract_scalar(requestParameters, '$.key') as object,
    userIdentity.arn as userArn
FROM
    s3_cLOUDTRAIL_EVENTS_DB.cloudtrail_myawsexamplebucket_table
WHERE
    eventName = 'GetObject'
    AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example : selecionar todos os eventos de solicitante anônimo para um bucket em um período específico e imprimir somente EventTime, EventName, EventSource, SourceIP, UserAgent, BucketName, UserARN e AccountID

```
SELECT
    eventTime,
    eventName,
    eventSource,
    sourceIpAddress,
    userAgent,
    json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
    userIdentity.arn as userArn,
    userIdentity.accountId
FROM
    s3_cLOUDTRAIL_EVENTS_DB.cloudtrail_myawsexamplebucket_table
WHERE
    userIdentity.accountId = 'ANONYMOUS_PRINCIPAL'
    AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Note

- Esses exemplos de consulta também podem ser úteis para o monitoramento de segurança. Você pode ver os resultados de chamadas PutObject ou GetObject de solicitantes/endereços IP inesperados ou não autorizados e identificar solicitações anônimas ao seu bucket.

- Essa consulta recupera somente informações do momento no qual o registro estava habilitado.

Se você estiver usando logs de acesso ao servidor do Amazon S3, consulte [Identificar solicitações de acesso a objetos usando logs de acesso do Amazon S3 \(p. 1066\)](#).

Registrar em log as solicitações com registro em log de acesso ao servidor

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 ajudá-lo a conhecer sua base de clientes e entender a conta do Amazon S3.

Note

Os logs de acesso ao servidor não registram informações sobre erros de redirecionamento a regiões erradas para regiões lançadas após 20 de março de 2019. Erros de redirecionamento para regiões erradas ocorrem quando uma solicitação para um objeto ou bucket é feita fora da região em que o bucket existe.

Como faço para habilitar a entrega de logs?

Para habilitar a entrega de logs, execute as seguintes etapas básicas. Para obter mais detalhes, consulte [Habilitar o log de acesso ao servidor do Amazon S3 \(p. 1043\)](#).

1. Forneça o nome do bucket de destino. Esse bucket é onde você deseja que o Amazon S3 salve os logs de acesso como objetos. Os buckets de origem e de destino devem estar na mesma Região da AWS e pertencer à mesma conta.

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. No entanto, para um gerenciamento de logs mais simples, recomendamos que você salve logs de acesso em um bucket diferente.

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. Não recomendamos fazer isso, pois pode gerar um pequeno aumento na cobrança do armazenamento. Além disso, com os logs adicionais sobre logs, pode ser mais difícil encontrar o log que você está procurando. 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 começem com uma string em comum e seja fácil identificar esses objetos.

[Prefixos de chaves](#) também são úteis para distinguir entre buckets de origem quando vários buckets são registrados em log no mesmo bucket de origem.

2. (Opcional) Atribua um prefixo a todas as chaves de objeto de log do Amazon S3. 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.

`logs/2013-11-01-21-32-16-E568B2907131C0C0`

O prefixo de chaves também ajuda ao excluir os logs. Por exemplo, defina uma regra de configuração do ciclo de vida para que o Amazon S3 exclua objetos com um prefixo de chave específico. Para obter mais informações, consulte [Exclusão de arquivos de log do Amazon S3 \(p. 1063\)](#).

3. (Opcional) Defina permissões em concessões de destino para que outros possam acessar os logs gerados. Por padrão, somente o proprietário do bucket sempre tem acesso completo aos objetos de log.

Se o bucket de destino (onde os logs de acesso do servidor estão armazenados) usar a configuração imposta pelo proprietário do bucket para S3 Object Ownership para desabilitar as listas de controle de acesso (ACLs), você não poderá conceder permissões nas concessões de destino que usam ACLs. Porém, você pode atualizar sua política de bucket para o bucket de destino para conceder acesso a outros. Para obter mais informações, consulte [Identity and Access Management no Amazon S3 \(p. 419\)](#) e [Permissões para entrega de logs \(p. 1044\)](#).

Formato da chave de objeto de log

O Amazon S3 usa o seguinte formato de chave de objeto 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. Essas datas e horas estão em Tempo Universal Coordenado (UTC).

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.

A barra à direita / é necessária para denotar o fim do prefixo.

Como os logs são entregues?

O Amazon S3 coleta periodicamente os registros de log de acessos, consolida-os em arquivos de log e, em seguida, faz o 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 para gravar logs de acesso ao servidor. Essas gravações estão sujeitas a restrições usuais de controle de acesso. Recomendamos atualizar a política de bucket no bucket de destino para conceder acesso à entidade principal do serviço de log (`logging.s3.amazonaws.com`) para entrega de logs de acesso. Porém, você também pode conceder acesso para entrega de logs de acesso ao grupo de entrega de logs do S3 por meio da lista de controle de acesso (ACL) de bucket. Não é recomendável conceder acesso ao grupo de entrega de logs do S3 usando a ACL de bucket.

Ao habilitar o registro de acesso ao servidor e conceder acesso para entrega de logs de acesso por meio da política de bucket, atualize a política de bucket no bucket de destino para permitir acesso `s3:PutObject` para a entidade principal do serviço de log. Se você usar o console do Amazon S3 para habilitar logs de acesso ao servidor em um bucket, o console atualizará automaticamente a política de bucket no bucket de destino para conceder essas permissões à entidade principal do serviço de log. Para obter mais informações sobre como conceder permissões para logs de acesso ao servidor, consulte [Permissões para entrega de logs \(p. 1044\)](#).

Configuração aplicada pelo proprietário do bucket em S3 Object Ownership

Se o bucket de destino usar a configuração imposta pelo proprietário do bucket para Object Ownership, as ACLs serão desabilitadas e não afetarão mais as permissões. É necessário atualizar a política de bucket no bucket de destino para conceder acesso à entidade principal do serviço de log. Para obter informações

sobre o Object Ownership, consulte [Conceder acesso ao grupo de entrega de logs do S3 para log de acesso ao servidor \(p. 649\)](#).

Entrega de logs do 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 [AWS Management Console](#)) 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.

Para obter mais informações sobre registro em log e arquivos de log, consulte as seguintes seções:

Tópicos

- [Habilitar o log de acesso ao servidor do Amazon S3 \(p. 1043\)](#)
- [Formato dos logs de acesso ao servidor do Amazon S3 \(p. 1052\)](#)
- [Exclusão de arquivos de log do Amazon S3 \(p. 1063\)](#)
- [Uso de logs de acesso do Amazon S3 para identificar solicitações \(p. 1063\)](#)

Habilitar o log de acesso ao servidor do Amazon S3

O registro em log de acesso ao servidor fornece detalhes sobre as solicitações que são feitas a um bucket do Amazon S3. 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 ajudá-lo a conhecer sua base de clientes e entender a conta do Amazon S3.

Por padrão, o Amazon S3 não coleta logs de acesso ao servidor. Quando você habilita o registro em log, o Amazon S3 entrega logs de acesso a um bucket de origem ou de destino de sua escolha. O bucket de destino deve estar na mesma Região da AWS e Conta da AWS que o bucket de origem e não deve ter uma configuração de período de retenção.

Um registro do log de acesso contém detalhes sobre as solicitações feitas a um bucket. Essa informação pode incluir o tipo de solicitação, os recursos que foram especificados na solicitação e a hora e data em

que a solicitação foi processada. Para obter mais informações sobre noções básicas de registro em log, consulte [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#).

Important

- Não há custo adicional para a ativação de registro de acesso ao servidor em um bucket do Amazon S3. No entanto, todos os arquivos de log que o sistema fornece acumularão as cobranças normais de armazenamento. Você pode excluir os arquivos de registro em log a qualquer momento. Não computamos custo pela transferência de dados na entrega de arquivos de log, mas cobramos a taxa normal de transferência de dados pelo acesso aos arquivos de log.
- O bucket de destino não deve ter registro em log de acesso ao servidor habilitado. 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. No entanto, isso causaria um loop infinito de logs e não é recomendado. Para um gerenciamento de logs mais simples, recomendamos que você salve logs de acesso em um bucket diferente. Para obter mais informações, consulte [Como faço para habilitar a entrega de logs? \(p. 1041\)](#)

Você pode habilitar ou desabilitar o registro em log de acesso ao servidor usando o console do Amazon S3, a API do Amazon S3, a AWS Command Line Interface (AWS CLI) ou os AWS SDKs.

Antes de habilitar o registro em log de acesso ao servidor, considere o seguinte:

- Você pode usar uma política de bucket ou listas de controle de acesso de bucket (ACL) para conceder permissões de entrega de log. No entanto, recomendamos usar uma política de bucket. Se o bucket de destino usar a configuração imposta pelo proprietário do bucket para Object Ownership, as ACLs serão desabilitadas e não afetarão mais as permissões. É necessário usar uma política de bucket para conceder permissões de acesso à entidade principal do serviço de log. Para obter mais informações, consulte [Permissões para entrega de logs \(p. 1044\)](#).
- Adicionar condições de negação a uma política de bucket pode impedir que o Amazon S3 entregue logs de acesso.
- Você pode usar a [criptografia de bucket padrão](#) no bucket de destino somente se usar AES256 (SSE-S3). Não há suporte para criptografia padrão com chaves do AWS KMS (SSE-KMS).
- Não é possível habilitar o Bloqueio de objetos do S3 no bucket de destino.

Permissões para entrega de logs

O Amazon S3 usa uma conta especial de entrega de logs para gravar logs de acesso ao servidor. Essas gravações estão sujeitas a restrições usuais de controle de acesso. Recomendamos atualizar a política de bucket no bucket de destino para conceder acesso à entidade principal do serviço de log (`logging.s3.amazonaws.com`) para entrega de logs de acesso.

Para conceder acesso usando a política de bucket no bucket de destino, atualize a política de bucket para permitir acesso `s3:PutObject` para a entidade principal do serviço de log. Se você usar o console do Amazon S3 para habilitar logs de acesso ao servidor, o console atualizará automaticamente a política de bucket no bucket de destino para conceder essas permissões à entidade principal do serviço de log. Se você habilitar o log de acesso ao servidor de maneira programática, poderá atualizar manualmente a política de bucket para o bucket de destino para conceder acesso à entidade principal do serviço de log.

Você pode alternar com ACLs de bucket para conceder acesso à entrega de logs de acesso. Adicione uma entrada de concessão à ACL de bucket que concede as permissões `WRITE` e `READ_ACP` para o grupo de entrega de logs do S3. Não é recomendável conceder acesso ao grupo de entrega de logs do S3 usando a ACL de bucket.

Configuração aplicada pelo proprietário do bucket em S3 Object Ownership

Se o bucket de destino usar a configuração imposta pelo proprietário do bucket para Object Ownership, as ACLs serão desabilitadas e não afetarão mais as permissões. É necessário atualizar a política de bucket do bucket de destino para conceder acesso à entidade principal do serviço de log. Não é possível atualizar sua ACL de bucket para conceder acesso ao grupo de entrega de logs do S3. Também não é possível incluir concessões de destino na configuração [PutBucketLogging](#). Para obter informações sobre como migrar ACLs de bucket existentes para entrega de logs de acesso para uma política de bucket, consulte [Conceder acesso ao grupo de entrega de logs do S3 para log de acesso ao servidor \(p. 649\)](#). Para obter informações sobre o Object Ownership, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Conceder permissões de acesso à entidade principal do serviço de log usando uma política de bucket

Este exemplo de política de bucket concede permissões s3:PutObject à entidade principal do serviço de log (logging.s3.amazonaws.com). Para usar essa política de buckets, substitua os valores do exemplo.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "S3ServerAccessLogsPolicy",  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "logging.s3.amazonaws.com"  
            },  
            "Action": [  
                "s3:PutObject"  
            ],  
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/EXAMPLE-LOGGING-PREFIX\*",  
            "Condition": {  
                "ArnLike": {  
                    "aws:SourceArn": "arn:aws:s3:::SOURCE-BUCKET-NAME"  
                },  
                "StringEquals": {  
                    "aws:SourceAccount": "SOURCE-ACCOUNT-ID"  
                }  
            }  
        }  
    ]  
}
```

Conceder permissões para o grupo de entrega de logs usando a ACL de bucket

Embora esse método não seja recomendado, você pode conceder permissões ao grupo de entrega de logs usando a ACL de bucket. Porém, se o bucket de destino usar a configuração aplicada pelo proprietário do bucket para Object Ownership, você não poderá definir ACLs de bucket ou de objeto. Também não é possível incluir concessões de destino na configuração [PutBucketLogging](#). É necessário usar uma política de bucket para conceder acesso à entidade principal do serviço de log (logging.s3.amazonaws.com). Para obter mais informações, consulte [Permissões para entrega de logs \(p. 1044\)](#).

Na ACL de bucket, o grupo de entrega de logs é representado pelo URL a seguir.

```
http://acs.amazonaws.com/groups/s3/LogDelivery
```

Para conceder permissões WRITE e READ_ACP (leitura ACL), adicione as seguintes concessões à ACL do bucket de destino.

```
<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 obter exemplos de adição de concessões de ACL de forma programada, consulte [Configurar ACLs \(p. 596\)](#).

Important

Ao habilitar o log de acesso ao servidor do Amazon S3 usando o AWS CloudFormation em um bucket e usar ACLs para conceder acesso ao grupo de entrega de logs do S3, você também deverá adicionar "AccessControl": "LogDeliveryWrite" ao campo de propriedade do seu bucket. Isso é importante porque você só pode conceder essas permissões criando uma ACL para o bucket, mas não pode criar ACLs personalizadas para buckets no CloudFormation. Você só pode usar ACLs pré-configuradas.

Para habilitar o log de acesso ao servidor

Use os exemplos a seguir para habilitar o log de acesso ao servidor usando o AWS Management Console, a AWS CLI, a API REST e o AWS SDK for .NET.

Uso do console do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket para o qual você deseja habilitar o log de acesso ao servidor.
3. Escolha Properties (Propriedades).
4. Na seção Server access logging (Registro de acesso ao servidor) escolha Edit (Editar).
5. Em Server access logging (Registro de acesso ao servidor), selecione Enable (Ativar).
6. Em Target bucket (Bucket de destino), insira o nome do bucket que você deseja que receba os objetos de registro em log.

O bucket de destino deve estar na mesma região do bucket de origem e não deve ter uma configuração de período de retenção.

7. Selecione Save changes.

Quando você habilitar o log de acesso ao servidor em um bucket, o console habilitará o log no bucket de origem e atualizará a política de bucket do bucket de destino para conceder permissões s3:PutObject para a entidade principal do serviço de log (logging.s3.amazonaws.com). Para obter mais informações sobre esta política de bucket, consulte [Conceder permissões de acesso à entidade principal do serviço de log usando uma política de bucket \(p. 1045\)](#).

Você pode visualizar os logs no bucket de destino. Depois de habilitar o registro em log de acesso ao servidor, pode demorar algumas horas antes que os logs sejam entregues no bucket de destino. Para obter mais informações sobre como e quando os logs são entregues, consulte [Como os logs são entregues? \(p. 1042\)](#).

Para obter mais informações, consulte [Visualização das propriedades de um bucket do S3 \(p. 134\)](#).

Uso dos REST API

Para habilitar o registro em log, envie uma solicitação [PUT Bucket logging](#) para adicionar a configuração de registro em log 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 do S3, e o proprietário do bucket tem permissões totais sobre os objetos de log. Você tem a opção de usar concessões de destino para conceder permissões a outros usuários para que eles possam acessar os logs. Para obter mais informações, consulte [PUT Bucket Logging \(Registro em log de PUT Bucket\)](#).

Note

Se o bucket de destino usar a configuração aplicada pelo proprietário do bucket para Object Ownership, você não poderá usar concessões de destino para conceder permissões a outros usuários. Para conceder permissões a outros, você pode usar atualizar a política de bucket no bucket de destino. Para obter mais informações, consulte [Permissões para entrega de logs \(p. 1044\)](#).

O Amazon S3 também fornece a API [GET Bucket logging \(Registro em log de GET Bucket\)](#) para recuperar a configuração de log em um bucket. Para excluir a configuração de registro, envie a solicitação de registro em log do PUT Bucket 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 AWS SDK para habilitar o log em um bucket.

Uso de AWS SDKs

.NET

O exemplo C# a seguir habilita o log em um bucket. Você deve criar dois buckets, um de origem e um de destino. Primeiro, o exemplo atualiza a ACL de bucket no bucket de destino e concede ao grupo de entrega de logs as permissões necessárias para registrar logs no bucket de destino e habilita o log no bucket de origem.

Este exemplo não funcionará em buckets de destino que usam a configuração aplicada pelo proprietário do bucket para Object Ownership.

Se o bucket de destino usar a configuração aplicada pelo proprietário do bucket para Object Ownership, você não poderá definir ACLs de bucket ou de objeto. Também não é possível incluir concessões de destino na configuração `PutBucketLogging`. É necessário usar uma política de bucket para conceder acesso à entidade principal do serviço de log (`logging.s3.amazonaws.com`). Para obter mais informações, consulte [Permissões para entrega de logs \(p. 1044\)](#).

Example

```
using Amazon;
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);
    }
}
```

Uso do AWS CLI

Recomendamos que você crie um bucket de registro dedicado em cada Região da AWS em que tenha buckets do S3. Depois, faça com que o Amazon S3 acesse o log entregue a esse bucket do S3. Para obter mais informações e exemplos, consulte [put-bucket-logging](#) na Referência da AWS CLI.

Se o bucket de destino usar a configuração aplicada pelo proprietário do bucket para Object Ownership, você não poderá definir ACLs de bucket ou de objeto. Também não é possível incluir concessões de destino na configuração [PutBucketLogging](#). É necessário usar uma política de bucket para conceder acesso à entidade principal do serviço de log (`logging.s3.amazonaws.com`). Para obter mais informações, consulte [Permissões para entrega de logs \(p. 1044\)](#).

Example : Habilite logs de acesso com cinco buckets em duas regiões

Neste exemplo, você tem estes cinco buckets:

- 1-awsexamplebucket1-us-east-1
- 2-awsexamplebucket1-us-east-1
- 3-awsexamplebucket1-us-east-1
- 1-awsexamplebucket1-us-west-2
- 2-awsexamplebucket1-us-west-2

1. Crie dois buckets de registro nas seguintes regiões:

- awsexamplebucket1-logs-us-east-1
- awsexamplebucket1-logs-us-west-2

2. Depois, ative os logs de acesso do Amazon S3 da seguinte forma:

- 1-awsexamplebucket1-us-east-1 é registrado em log no bucket do S3 `awsexamplebucket1-logs-us-east-1` com prefixo `1-awsexamplebucket1-us-east-1`
- 2-awsexamplebucket1-us-east-1 é registrado em log no bucket do S3 `awsexamplebucket1-logs-us-east-1` com prefixo `2-awsexamplebucket1-us-east-1`
- 3-awsexamplebucket1-us-east-1 é registrado em log no bucket do S3 `awsexamplebucket1-logs-us-east-1` com prefixo `3-awsexamplebucket1-us-east-1`
- 1-awsexamplebucket1-us-west-2 é registrado em log no bucket do S3 `awsexamplebucket1-logs-us-west-2` com prefixo `1-awsexamplebucket1-us-west-2`
- 2-awsexamplebucket1-us-west-2 é registrado em log no bucket do S3 `awsexamplebucket1-logs-us-west-2` com prefixo `2-awsexamplebucket1-us-west-2`

3. Conceda permissões para entrega de logs de acesso ao servidor usando uma ACL de bucket ou uma política de bucket:

- Update the bucket policy (Atualizar a política de bucket) (recomendado): para conceder permissões à entidade principal do serviço de log, use put-bucket-policy:

```
aws s3api put-bucket-policy --bucket awsexamplebucket1-logs --policy file://
policy.json
```

Policy.json é um documento JSON na pasta atual que contém a política de buckets. Para usar essa política de buckets, substitua os valores do exemplo.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ServerAccessLogsPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::awsexamplebucket1-logs/*",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::SOURCE-BUCKET-NAME"
                },
                "StringEquals": {
                    "aws:SourceAccount": "SOURCE-ACCOUNT-ID"
                }
            }
        }
    ]
}
```

- Update the bucket ACL (Atualizar a ACL de bucket): para conceder permissões ao grupo de entrega de logs do S3, use put-bucket-acl.

```
aws s3api put-bucket-acl --bucket awsexamplebucket1-logs --grant-write
    URI=http://acs.amazonaws.com/groups/s3/LogDelivery --grant-read-acp URI=http://
    acs.amazonaws.com/groups/s3/LogDelivery
```

4. Em seguida, aplique a política de registro em log:

```
aws s3api put-bucket-logging --bucket awsexamplebucket1 --bucket-logging-status file://
logging.json
```

Logging.json é um documento JSON na pasta atual que contém a configuração de log. Se um bucket usar a configuração aplicada pelo proprietário do bucket para Object Ownership, sua configuração de log não poderá conter concessões de destino. Para obter mais informações, consulte [Permissões para entrega de logs \(p. 1044\)](#).

Example – **Logging.json** sem concessões de destino

O exemplo de arquivo `Logging.json` a seguir não contém concessões de destino e pode ser aplicado a um bucket que usa a configuração aplicada pelo proprietário do bucket para Object Ownership.

```
{  
    "LoggingEnabled": {  
        "TargetBucket": "awsexamplebucket1-logs",  
        "TargetPrefix": "awsexamplebucket1/"  
    }  
}
```

Example – **Logging.json** com concessões de destino

O exemplo a seguir de arquivo `Logging.json` contém concessões de destino.

Se o bucket de destino usar a configuração aplicada pelo proprietário do bucket para Object Ownership, você não poderá incluir concessões de destino na configuração [PutBucketLogging](#). Para obter mais informações, consulte [Permissões para entrega de logs \(p. 1044\)](#).

```
{  
    "LoggingEnabled": {  
        "TargetBucket": "awsexamplebucket1-logs",  
        "TargetPrefix": "awsexamplebucket1/",  
        "TargetGrants": [  
            {  
                "Grantee": {  
                    "Type": "AmazonCustomerByEmail",  
                    "EmailAddress": "user@example.com"  
                },  
                "Permission": "FULL_CONTROL"  
            }  
        ]  
    }  
}
```

5. Use um script bash para acessar o registro em log de todos os buckets em sua conta:

Note

Esse script somente funcionará se todos os buckets estiverem na mesma região. Se você tiver buckets em várias regiões, deverá ajustar o script.

Example – Conceda acesso com políticas de bucket e adicione log para os buckets de sua conta

```
loggingBucket='awsexamplebucket1-logs'  
region='us-west-2'  
  
# Create Logging bucket  
aws s3 mb s3://$loggingBucket --region $region  
  
aws s3api put-bucket-policy --bucket $loggingBucket --policy file://policy.json
```

```
# List buckets in this account
buckets=$(aws s3 ls | awk '{print $3}')"

# Put bucket logging on each bucket
for bucket in $buckets
    do printf '{
    "LoggingEnabled": {
        "TargetBucket": "%s",
        "TargetPrefix": "%s/"
    }
}' "$loggingBucket" "$bucket" > logging.json
    aws s3api put-bucket-logging --bucket $bucket --bucket-logging-status file://
logging.json
    echo "$bucket done"
done

rm logging.json

echo "Complete"
```

Example – Conceda acesso com ACLs de bucket e adicione log para os buckets de sua conta

```
loggingBucket='awsexamplebucket1-logs'
region='us-west-2'

# Create Logging bucket
aws s3 mb s3://$loggingBucket --region $region

aws s3api put-bucket-acl --bucket $loggingBucket --grant-write URI=http://
acs.amazonaws.com/groups/s3/LogDelivery --grant-read-acp URI=http://acs.amazonaws.com/
groups/s3/LogDelivery

# List buckets in this account
buckets=$(aws s3 ls | awk '{print $3}')"

# Put bucket logging on each bucket
for bucket in $buckets
    do printf '{
    "LoggingEnabled": {
        "TargetBucket": "%s",
        "TargetPrefix": "%s/"
    }
}' "$loggingBucket" "$bucket" > logging.json
    aws s3api put-bucket-logging --bucket $bucket --bucket-logging-status file://
logging.json
    echo "$bucket done"
done

rm logging.json

echo "Complete"
```

Formato dos logs de acesso ao servidor do Amazon S3

O registro em log de acesso ao servidor fornece detalhes sobre as solicitações que são feitas a um bucket do Amazon S3. Você pode usar logs de acesso ao servidor para auditorias de segurança e acesso, saber

mais sobre sua base de clientes ou entender sua fatura do Amazon S3. Esta seção descreve o formato e outros detalhes sobre os arquivos de log de acesso ao servidor do Amazon S3.

Os arquivos de log de acesso ao servidor consistem em uma sequência de registros de log delimitados por novas linhas. Cada registro do log representa uma solicitação e consiste em campos delimitados por espaço.

Veja a seguir o exemplo de um log que consiste em cinco registros de log.

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
awsexamplebucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 3E57427F3EXAMPLE
REST.GET.VERSIONING - "GET /awsexamplebucket1?versioning HTTP/1.1" 200 - 113 - 7 -
 "-" "S3Console/0.4" - s9lzHyrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/
XV/VLi31234= SigV2 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader awsexamplebucket1.s3.us-
west-1.amazonaws.com TLSV1.1
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
awsexamplebucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 891CE47D2EXAMPLE
REST.GET.LOGGING_STATUS - "GET /awsexamplebucket1?logging HTTP/1.1" 200 - 242
- 11 - "-" "S3Console/0.4" - 9vKBE6vMhrNiWHZmb2L0mXOcqPGzQOI5XLnCtZNPxev+Hf
+7tpT6sxDwDty4LHBu0ZJG96N1234= SigV2 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader
awsexamplebucket1.s3.us-west-1.amazonaws.com TLSV1.1
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
awsexamplebucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be A1206F460EXAMPLE
REST.GET.BUCKETPOLICY - "GET /awsexamplebucket1?policy HTTP/1.1" 404
NoSuchBucketPolicy 297 - 38 - "-" "S3Console/0.4" - BNabsXZQDbssi6xMBdBU2sLt
+Yf5kZDmeBUP35sFoKa3sLLeMC78iwEIWxs99CRUrbS4n11234= SigV2 ECDHE-RSA-AES128-GCM-SHA256
AuthHeader awsexamplebucket1.s3.us-west-1.amazonaws.com TLSV1.1
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
awsexamplebucket1 [06/Feb/2019:00:01:00 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 7B4A0FABBEXAMPLE
REST.GET.VERSIONING - "GET /awsexamplebucket1?versioning HTTP/1.1" 200 - 113 - 33 - "-"
"S3Console/0.4" - Ke1bUcazaN1jWuUlPJaxF64cQVpUEhoZKEG/hmy/gijN/I1DeWqDfFvnpybfEseEME/
u7ME1234= SigV2 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader awsexamplebucket1.s3.us-
west-1.amazonaws.com TLSV1.1
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
awsexamplebucket1 [06/Feb/2019:00:01:57 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
DD6CC733AEXAMPLE REST.PUT.OBJECT s3-dg.pdf "PUT /awsexamplebucket1/
s3-dg.pdf HTTP/1.1" 200 - 4406583 41754 28 "-" "S3Console/0.4" -
10S62Zv81kBW7BB6SX4XJ48o6kpc16LPwEoizzQQxJd5qDSCTLX0TgS37kYUBKQW3+bPdrgr1234= SigV4 ECDHE-
RSA-AES128-SHA AuthHeader awsexamplebucket1.s3.us-west-1.amazonaws.com TLSV1.1
```

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.

Tópicos

- [Campos de registro de log \(p. 1053\)](#)
- [Registro em log adicional para operações de cópia \(p. 1058\)](#)
- [Informações personalizadas do log de acesso \(p. 1062\)](#)
- [Considerações de programação para o formato do log de acesso ao servidor extensível \(p. 1062\)](#)

Campos de registro de log

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 outra forma de ID da Conta da AWS. Para obter mais informações sobre IDs de usuário canônico, consulte [Identificadores de conta da Conta da AWS](#) na Referência geral da AWS. Para obter informações sobre como encontrar o ID de usuário canônico da conta, consulte [Como localizar o ID de usuário canônico de sua Conta da AWS](#).

Exemplo de registro

```
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.

Exemplo de registro

```
awsexamplebucket1
```

Tempo

O horário em que a solicitação foi recebida. As datas e horas estão em Tempo Universal Coordenado (UTC). O formato que usa a terminologia `strftime()` é o seguinte: [%d/%b/%Y:%H:%M:%S %z]

Exemplo de registro

```
[ 06/Feb/2019:00:00:38 +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.

Exemplo de registro

```
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.

Exemplo de registro

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

ID da solicitação

Uma string gerada pelo Amazon S3 para identificar exclusivamente cada solicitação.

Exemplo de registro

```
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, ou S3.action.resource_type para [Ciclo de vida e registro em log \(p. 765\)](#).

Exemplo de registro

```
REST.PUT.OBJECT
```

Chave

A parte “chave” da solicitação, codificada pela URL ou “-”, se a operação não usar um parâmetro de chave.

Exemplo de registro

```
/photos/2019/08/puppy.jpg
```

Request-URI

A parte de Request-URI da mensagem de solicitação HTTP.

Entrada de exemplo

```
"GET /awsexamplebucket1/photos/2019/08/puppy.jpg?x-foo=bar HTTP/1.1"
```

Status HTTP

O código numérico do status do HTTP da resposta.

Exemplo de registro

```
200
```

Código de erro

O Amazon S3 [Código de erro \(p. 1277\)](#) ou “-”, se nenhum erro ocorreu.

Exemplo de registro

```
NoSuchBucket
```

Bytes enviados

O número de bytes de resposta enviados excluindo a sobrecarga do protocolo HTTP ou “-”, se zero.

Exemplo de registro

```
2662992
```

Tamanho do objeto

O tamanho total do objeto em questão.

Exemplo de registro

```
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.

Exemplo de registro

```
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.

Exemplo de registro

```
10
```

Referer

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.

Exemplo de registro

```
"http://www.amazon.com/webservices"
```

Agente de usuário

O valor do cabeçalho do agente de usuário do HTTP.

Exemplo de registro

```
"curl/7.15.1"
```

Id da versão

O ID da versão na solicitação ou “-”, se a operação não usar um parâmetro `versionId`.

Exemplo de registro

```
3HL4kqtJvjVBH40Nrjfkd
```

ID do host

O ID de solicitação estendida x-amz-id-2 ou Amazon S3.

Exemplo de registro

```
s9lzMHyFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Versão do Signature

A versão do Signature, SigV2 ou SigV4, que foi usada para autenticar a solicitação ou – para solicitações não autenticadas.

Exemplo de registro

```
SigV2
```

Pacote de criptografia

A codificação Secure Sockets Layer (SSL) que foi negociada para a solicitação HTTPS ou – para HTTP.

Exemplo de registro

```
ECDHE-RSA-AES128-GCM-SHA256
```

Tipo de autenticação

O tipo da autenticação de solicitação usado, AuthHeader para cabeçalhos de autenticação, QueryString para string de consulta (pre-signed URL) ou – para solicitações não autenticadas.

Exemplo de registro

```
AuthHeader
```

Cabeçalho de host

O endpoint usado para conectar-se ao Amazon S3.

Exemplo de registro

```
s3.us-west-2.amazonaws.com
```

Algumas regiões mais antigas oferecem suporte a endpoints legados. Você poderá ver esses endpoints nos logs de acesso ao servidor ou nos logs do AWS CloudTrail. Para obter mais informações, consulte [Endpoints herdados \(p. 1253\)](#). Para obter uma lista completa de regiões e endpoints do Amazon S3, consulte [Amazon S3 endpoints and quotas](#) (Endpoints e cotas do Amazon S3) na Amazon Web Services General Reference (Referência geral da Amazon Web Services).

Versão do TLS

A versão do Transport Layer Security (TLS) negociada pelo cliente. O valor é um dos seguintes: TLSv1, TLSv1.1, TLSv1.2 ou – se o TLS não foi usado.

Exemplo de registro

TLSv1.2

ARN (nome do recurso da Amazon) do ponto de acesso

O nome de recurso da Amazon (ARN) do ponto de acesso da solicitação. Se o ARN do ponto de acesso estiver malformado ou sem uso, o campo apresentará um '-'. Para obter mais informações sobre os pontos de acesso, consulte [Usar pontos de acesso \(p. 329\)](#). Para obter mais informações sobre os ARNs, consulte o tópico sobre o [Nome do recurso da Amazon \(ARN\)](#) no Guia de referência da AWS.

Exemplo de registro

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

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 seção 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 outra forma de ID da Conta da AWS. Para obter mais informações sobre IDs de usuário canônico, consulte [Identificadores de conta da Conta da AWS](#) na Referência geral da AWS. Para obter informações sobre como encontrar o ID de usuário canônico da conta, consulte [Como localizar o ID de usuário canônico de sua Conta da AWS](#).

Exemplo de registro

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Bucket

O nome do bucket que armazena o objeto que está sendo copiado.

Exemplo de registro

```
awsexamplebucket1
```

Tempo

O horário em que a solicitação foi recebida. As datas e horas estão em Tempo Universal Coordenado (UTC). O formato que usa a terminologia strftime() é o seguinte: [%d/%B/%Y:%H:%M:%S %z]

Exemplo de registro

```
[06/Feb/2019:00:00:38 +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.

Exemplo de registro

```
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.

Exemplo de registro

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

ID da solicitação

Uma string gerada pelo Amazon S3 para identificar exclusivamente cada solicitação.

Exemplo de registro

```
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`.

Exemplo de registro

```
REST.COPY.OBJECT_GET
```

Chave

A “chave” do objeto que está sendo copiado ou “-”, se a operação não usar um parâmetro de chave.

Exemplo de registro

```
/photos/2019/08/puppy.jpg
```

Request-URI

A parte de Request-URI da mensagem de solicitação HTTP.

Exemplo de registro

```
"GET /awsexamplebucket1/photos/2019/08/puppy.jpg?x-foo=bar"
```

Status HTTP

O código numérico do status do HTTP da parte GET da operação de cópia.

Exemplo de registro

200

Código de erro

O [Código de erro \(p. 1277\)](#) do Amazon S3, da parte GET da operação de cópia ou “-”, se nenhum erro tiver ocorrido.

Exemplo de registro

NoSuchBucket

Bytes enviados

O número de bytes de resposta enviados excluindo a sobrecarga do protocolo HTTP ou “-”, se zero.

Exemplo de registro

2662992

Tamanho do objeto

O tamanho total do objeto em questão.

Exemplo de registro

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.

Exemplo de registro

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.

Exemplo de registro

10

Referer

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.

Exemplo de registro

```
"http://www.amazon.com/webservices"
```

Agente de usuário

O valor do cabeçalho do agente de usuário do HTTP.

Exemplo de registro

```
"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
```

ID do host

O ID de solicitação estendida `x-amz-id-2` ou Amazon S3.

Exemplo de registro

```
s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsip/XV/VLi31234=
```

Versão do Signature

A versão do Signature, `SigV2` ou `SigV4`, que foi usada para autenticar a solicitação ou – para solicitações não autenticadas.

Exemplo de registro

```
SigV2
```

Pacote de criptografia

A codificação Secure Sockets Layer (SSL) que foi negociada para a solicitação HTTPS ou – para HTTP.

Exemplo de registro

```
ECDHE-RSA-AES128-GCM-SHA256
```

Tipo de autenticação

O tipo de autenticação de solicitação usada, `AuthHeader` para cabeçalhos de autenticação, `QueryString` para string de consulta (pre-signed URL) ou um – para solicitações não autenticadas.

Exemplo de registro

```
AuthHeader
```

Cabeçalho de host

O endpoint usado para conectar-se ao Amazon S3.

Exemplo de registro

```
s3.us-west-2.amazonaws.com
```

Algumas regiões mais antigas oferecem suporte a endpoints legados. Você poderá ver esses endpoints nos logs de acesso ao servidor ou nos logs do AWS CloudTrail. Para obter mais informações, consulte [Endpoints herdados \(p. 1253\)](#). Para obter uma lista completa de regiões e endpoints do Amazon S3, consulte [Amazon S3 endpoints and quotas \(Endpoints e cotas do Amazon S3\)](#) na Amazon Web Services General Reference (Referência geral da Amazon Web Services).

Versão do TLS

A versão do Transport Layer Security (TLS) negociada pelo cliente. O valor é um dos seguintes: `TLSv1`, `TLSv1.1`, `TLSv1.2` ou `-` se o TLS não foi usado.

Exemplo de registro

```
TLSv1.2
```

ARN (nome do recurso da Amazon) do ponto de acesso

O nome de recurso da Amazon (ARN) do ponto de acesso da solicitação. Se o ARN do ponto de acesso estiver malformado ou sem uso, o campo apresentará um `'.'`. Para obter mais informações sobre os pontos de acesso, consulte [Usar pontos de acesso \(p. 329\)](#). Para obter mais informações sobre os ARNs, consulte o tópico sobre o [Nome do recurso da Amazon \(ARN\)](#) no Guia de referência da AWS.

Exemplo de registro

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

Informações personalizadas do log de acesso

Você pode incluir informações personalizadas a serem armazenadas no registro de log de acesso de uma solicitação. Para fazer isso, adicione um parâmetro de string de consulta personalizado à URL da solicitação. O Amazon S3 ignora os parâmetros query-string que começam com `"x-"`, mas inclui esses parâmetros no registro do log de acesso da solicitação, como parte do campo `Request-URI` do registro do log.

Por exemplo, uma solicitação GET para `"s3.amazonaws.com/awsexamplebucket1/photos/2019/08/puppy.jpg?x-user=johndoe"` funciona da mesma forma que a solicitação para `"s3.amazonaws.com/awsexamplebucket1/photos/2019/08/puppy.jpg"`, exceto pelo fato de que a string `"x-user=johndoe"` está incluída no campo `Request-URI` do registro de log associado. Essa funcionalidade está disponível apenas na interface REST.

Considerações de programação para o formato do log de acesso ao servidor extensível

Ocasionalmente, podemos estender o formato de registro de log de acesso adicionando novos campos ao final de cada linha. Portanto, você deve escrever qualquer código que analise logs de acesso ao servidor para lidar com os campos finais que ele possa não entender.

Exclusão de arquivos de log do Amazon S3

Um bucket do Amazon S3 com registro em log de acesso ao servidor habilitado pode acumular muitos objetos de log do servidor ao longo do tempo. Seu aplicativo pode precisar desses logs de acesso por um período específico após sua criação e, depois disso, você poderá excluí-los. Você pode usar a configuração de ciclo de vida do Amazon S3 para definir regras para que o S3 coloque esses objetos automaticamente na fila para exclusão ao final de seus ciclos 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 [Gerenciando seu ciclo de vida de armazenamento \(p. 746\)](#).

Para obter mais informações sobre o registro em log de acesso ao servidor, consulte [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#).

Uso de logs de acesso do Amazon S3 para identificar solicitações

Você pode identificar solicitações do Amazon S3 usando logs de acesso do Amazon S3.

Note

- Recomendamos que você use eventos de dados do AWS CloudTrail em vez de logs de acesso do Amazon S3. Os eventos de dados do CloudTrail são mais fáceis de configurar e contêm mais informações. Para obter mais informações, consulte [Identificar solicitações do Amazon S3 usando o CloudTrail \(p. 1035\)](#).
- Dependendo de quantas solicitações de acesso você receber, poderá ser necessário mais recursos ou tempo para analisar seus logs.

Tópicos

- [Consultar logs de acesso para solicitações usando o Amazon Athena \(p. 1063\)](#)
- [Identificar solicitações do Signature versão 2 usando logs de acesso do Amazon S3 \(p. 1066\)](#)
- [Identificar solicitações de acesso a objetos usando logs de acesso do Amazon S3 \(p. 1066\)](#)

Consultar logs de acesso para solicitações usando o Amazon Athena

Você pode identificar solicitações do Amazon S3 com logs de acesso do Amazon S3 usando o Amazon Athena.

O Amazon S3 armazena logs de acesso ao servidor como objetos em um bucket do S3. Muitas vezes, é mais fácil usar uma ferramenta que possa analisar os logs no Amazon S3. O Athena oferece suporte à análise de objetos do S3 e pode ser usado para consultar logs de acesso do Amazon S3.

Example

O exemplo a seguir mostra como você pode consultar os logs de acesso ao servidor do Amazon S3 no Amazon Athena.

Note

Para especificar um local do Amazon S3 em uma consulta do Athena, é necessário formatar o nome do bucket de destino e o prefixo de destino onde os logs são entregues como um URI do S3, da seguinte forma: s3://*DOC-EXAMPLE-BUCKET1*-logs/*prefix*/

1. Abra o console do Athena em <https://console.aws.amazon.com/athena/>.
2. No Query Editor, execute um comando semelhante ao seguinte:

```
create database s3_access_logs_db
```

Note

É uma prática recomendada criar o banco de dados na mesma Região da AWS em que o bucket do S3 está.

3. No Query Editor, execute um comando semelhante ao seguinte para criar um esquema de tabela no banco de dados criado na etapa 2. Os valores dos tipo de dados STRING e BIGINT são propriedades do log de acesso. É possível consultar essas propriedades no Athena. Para LOCATION, insira o bucket do S3 e o caminho do prefixo conforme indicado anteriormente.

```
CREATE EXTERNAL TABLE `s3_access_logs_db.mybucket_logs`(
    `bucketowner` STRING,
    `bucket_name` STRING,
    `requestdatetime` STRING,
    `remoteip` STRING,
    `requester` STRING,
    `requestid` STRING,
    `operation` STRING,
    `key` STRING,
    `request_uri` STRING,
    `httpstatus` STRING,
    `errorcode` STRING,
    `bytesent` BIGINT,
    `objectsize` BIGINT,
    `totaltime` STRING,
    `turnaroundtime` STRING,
    `referrer` STRING,
    `useragent` STRING,
    `versionid` STRING,
    `hostid` STRING,
    `sigv` STRING,
    `ciphersuite` STRING,
    `authtype` STRING,
    `endpoint` STRING,
    `tlsversion` STRING)
ROW FORMAT SERDE
    'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
    'input.regex'='([^\n]*)([^ ]*)\\\[.*?\\]\\] ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)\\
    ([^\n]*[^\n]-) (-|[0-9]*)([^ ]*)([^ ]*)([^ ]*)([^ ]*)([^ ]*)([^ ]*)([^ ]*)\\
    \"|\"|(-)([^ ]*)(?: ([^ ]*)([^ ]*)([^ ]*)([^ ]*)([^ ]*)([^ ]*))?.*$')
STORED AS INPUTFORMAT
    'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
    'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
```

```
's3://awsexamplebucket1-logs/prefix/'
```

4. No painel de navegação, em Database (Banco de dados), escolha o banco de dados.
5. Em Tables (Tabelas), selecione Preview table (Visualizar tabela) ao lado do nome da tabela.

No painel Results (Resultados), você deve ver dados dos logs de acesso ao servidor, como `bucketowner`, `bucket`, `requestdatetime` e assim por diante. Isso significa que você criou a tabela do Athena com êxito. Agora você pode consultar os logs de acesso ao servidor do Amazon S3.

Example : Mostre quem excluiu um objeto e quando (timestamp, endereço IP e usuário IAM)

```
SELECT RequestDateTime, RemoteIP, Requester, Key
FROM s3_access_logs_db.mybucket_logs
WHERE key = 'images/picture.jpg' AND operation like '%DELETE%';
```

Example : Mostre todas as operações que foram realizadas por um usuário do IAM

```
SELECT *
FROM s3_access_logs_db.mybucket_logs
WHERE requester='arn:aws:iam::123456789123:user/user_name';
```

Example : mostrar todas as operações que foram realizadas em um objeto em um determinado período

```
SELECT *
FROM s3_access_logs_db.mybucket_logs
WHERE Key='prefix/images/picture.jpg'
AND parse_datetime(RequestDateTime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2017-02-18:07:00:00','yyyy-MM-dd:HH:mm:ss')
AND parse_datetime('2017-02-18:08:00:00','yyyy-MM-dd:HH:mm:ss');
```

Example : mostrar a quantidade de dados transferidos por um endereço IP específico em um determinado período

```
SELECT SUM(bytessent) AS uploadTotal,
       SUM(objectsize) AS downloadTotal,
       SUM(bytessent + objectsize) AS Total
  FROM s3_access_logs_db.mybucket_logs
 WHERE RemoteIP='1.2.3.4'
   AND parse_datetime(RequestDateTime,'dd/MMM/yyyy:HH:mm:ss Z')
   BETWEEN parse_datetime('2017-06-01','yyyy-MM-dd')
   AND parse_datetime('2017-07-01','yyyy-MM-dd');
```

Note

Para reduzir o tempo de retenção do log, você pode criar uma política de ciclo de vida do Amazon S3 para o bucket de logs de acesso ao servidor. Configure a política de ciclo de vida para remover os arquivos de log periodicamente. Fazer isso reduz a quantidade de dados que o Athena analisa

para cada consulta. Para obter mais informações, consulte [Definir a configuração do ciclo de vida em um bucket \(p. 753\)](#).

Identificar solicitações do Signature versão 2 usando logs de acesso do Amazon S3

O suporte do Amazon S3 para o Signature versão 2 será desativado (defasado). Depois disso, o Amazon S3 não aceitará mais solicitações que usam o Signature versão 2 e todas as solicitações deverão usar assinaturas do Signature versão 4. É possível identificar solicitações de acesso do Signature versão 2 usando logs de acesso do Amazon S3

Note

- Recomendamos que você use eventos de dados do AWS CloudTrail em vez de logs de acesso do Amazon S3. Os eventos de dados do CloudTrail são mais fáceis de configurar e contêm mais informações. Para obter mais informações, consulte [Identificar solicitações do Amazon S3 Signature versão 2 usando o CloudTrail \(p. 1037\)](#).

Example : mostrar todos os solicitantes que estão enviando tráfego do Signature versão 2

```
SELECT requester, Sigv, Count(Sigv) as SigCount
FROM s3_access_logs_db.mybucket_logs
GROUP BY requester, Sigv;
```

Identificar solicitações de acesso a objetos usando logs de acesso do Amazon S3

É possível usar consultas em logs de acesso ao servidor do Amazon S3 para identificar solicitações de acesso ao objeto do Amazon S3, para operações, como GET, PUT e DELETE, e descobrir mais informações sobre essas solicitações.

O exemplo de consulta do Amazon Athena a seguir mostra como obter todas as solicitações de objeto PUT para o Amazon S3 a partir do log de acesso ao servidor.

Example : mostrar todos os solicitantes que estão enviando solicitações de objeto PUT em um determinado período

```
SELECT Bucket, Requester, RemoteIP, Key, HTTPStatus, ErrorCode, RequestDateTime
FROM s3_access_logs_db
WHERE Operation='REST.PUT.OBJECT' AND
parse_datetime(RequestDateTime, 'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
```

O exemplo de consulta do Amazon Athena a seguir mostra como obter todas as solicitações de objeto GET para o Amazon S3 a partir do log de acesso ao servidor.

Example : mostrar todos os solicitantes que estão enviando solicitações de objeto GET em um determinado período

```
SELECT Bucket, Requester, RemoteIP, Key, HTTPStatus, ErrorCode, RequestDateTime
FROM s3_access_logs_db
```

```
WHERE Operation='REST.GET.OBJECT' AND
parse_datetime(RequestDateTime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

O exemplo de consulta do Amazon Athena a seguir mostra como obter todas as solicitações anônimas ao seu bucket do S3 do log de acesso ao servidor.

Example : mostrar todos os solicitantes anônimos que estão fazendo solicitações a um bucket em um determinado período

```
SELECT Bucket, Requester, RemoteIP, Key, HTTPStatus, ErrorCode, RequestDateTime
FROM s3_access_logs_db.mybucket_logs
WHERE Requester IS NULL AND
parse_datetime(RequestDateTime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

Note

- É possível modificar o intervalo de datas conforme necessário para atender às suas necessidades.
- Esses exemplos de consulta também podem ser úteis para o monitoramento de segurança. Você pode ver os resultados de chamadas PutObject ou GetObject de solicitantes/endereços IP inesperados ou não autorizados e identificar solicitações anônimas ao seu bucket.
- Essa consulta recupera somente informações do momento no qual o registro estava habilitado.
- Se você estiver usando logs do AWS CloudTrail do Amazon S3, consulte [Identificar o acesso a objetos do S3 usando o CloudTrail \(p. 1039\)](#).

Monitoramento de métricas com o Amazon CloudWatch

As métricas do Amazon CloudWatch para Amazon S3 podem ajudar você a entender e melhorar a performance das aplicações que usam o Amazon S3. Existem várias maneiras de usar o CloudWatch com o Amazon S3.

Métricas de armazenamento diárias para buckets

Monitore o armazenamento do bucket usando o CloudWatch, que coleta e processa dados de armazenamento do Amazon S3 para gerar métricas legíveis diárias. 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 solicitação

Monitore as solicitações do Amazon S3 para identificar e atuar rapidamente em problemas operacionais. As métricas estão disponíveis em intervalos de um minuto após alguma latência para processamento. Essas métricas do CloudWatch são cobradas na mesma taxa que as métricas personalizadas do Amazon CloudWatch. Para obter mais informações sobre a definição de preço do CloudWatch, consulte [Definição de preço do Amazon CloudWatch](#). Para saber como optar por obter essas métricas, consulte [Configurações de métricas do CloudWatch \(p. 1077\)](#).

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. Também é possível definir um filtro para as métricas usando um prefixo compartilhado, etiqueta de objeto ou ponto de acesso.

- Ponto de acesso: os pontos de acesso são endpoints de rede nomeados que estão conectados a buckets e simplificam o gerenciamento do acesso a dados em escala para conjuntos de dados compartilhados no S3. Com o filtro de ponto de acesso, você pode obter informações sobre o uso do ponto de acesso. Para obter mais informações sobre pontos de acesso, consulte [Monitorar e registrar de pontos de acesso \(p. 330\)](#).
- Prefixo: embora o modelo de dados do Amazon S3 seja uma estrutura plana, você pode inferir a hierarquia usando um prefixo. Um prefixo é semelhante a um nome de diretório que permite agrupar objetos semelhantes em um bucket. O console do 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. Para obter mais informações sobre prefixos, consulte [Organizar objetos usando prefixos \(p. 258\)](#).
- Etiquetas: etiquetas são pares de nomes de chave-valor que você pode adicionar aos objetos. As etiquetas ajudam a encontrar e organizar objetos com facilidade. Você também pode usar etiquetas como um filtro para configurações de métricas para que somente objetos com essas etiquetas sejam incluídos na configuração de métricas. Para obter mais informações sobre etiquetas de objeto, consulte [Categorizando seu armazenamento usando tags \(p. 878\)](#).

Para alinhar essas métricas a aplicações de negócios, fluxos de trabalho ou organizações internas específicas, você pode filtrar por um prefixo compartilhado, marca de objeto ou ponto de acesso.

Métricas de replicação

Métricas de replicação: monitore o número total de operações de API do S3 que estão pendentes de replicação, o tamanho total de objetos com replicação pendente e o tempo máximo de replicação para a região de destino. As regras de replicação com Controle de tempo de replicação do S3 (S3 RTC) ou métricas de replicação do S3 habilitadas publicarão as métricas de replicação.

Para obter mais informações, consulte [Monitoramento do progresso com métricas de replicação e notificações de eventos do Amazon S3 \(p. 857\)](#) ou [Atendimento aos requisitos de conformidade usando o Controle do tempo de replicação do S3 \(S3 RTC\) \(p. 860\)](#).

Métricas do Amazon S3 Storage Lens

É possível publicar métricas de uso e atividade do S3 Storage Lens no Amazon CloudWatch para criar uma visão unificada de sua integridade operacional nos [painéis](#) do CloudWatch. As métricas do S3 Storage Lens estão disponíveis no namespace `AWS/S3/Storage-Lens`. A opção de publicação do CloudWatch está disponível para painéis do S3 Storage Lens atualizados para métricas e recomendações avançadas. É possível habilitar a opção de publicação do CloudWatch para uma configuração de painel nova ou existente no S3 Storage Lens.

Para obter mais informações, consulte [Monitorar métricas do S3 Storage Lens no CloudWatch \(p. 1138\)](#).

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 aplicação Web estão se saindo. Para obter mais informações, consulte [O que é o Amazon CloudWatch?](#) no Guia do usuário do Amazon CloudWatch.

Para obter mais informações, consulte os tópicos a seguir.

Tópicos

- [Métricas e dimensões \(p. 1069\)](#)
- [Acessar métricas do CloudWatch \(p. 1076\)](#)
- [Configurações de métricas do CloudWatch \(p. 1077\)](#)

Métricas e dimensões

As métricas e dimensões de armazenamento que o Amazon S3 envia para o CloudWatch estão listadas abaixo.

Tópicos

- [Métricas de armazenamento diárias do Amazon S3 para buckets no CloudWatch \(p. 1069\)](#)
- [Métricas de solicitação do Amazon S3 no CloudWatch \(p. 1070\)](#)
- [Métricas de replicação do Amazon S3 no CloudWatch \(p. 1072\)](#)
- [Métricas do Amazon S3 Storage Lens no CloudWatch \(p. 1073\)](#)
- [Métricas do Amazon S3 on Outposts no CloudWatch \(p. 1073\)](#)
- [Dimensões do Amazon S3 no CloudWatch \(p. 1074\)](#)
- [Dimensões do Amazon S3 Storage Lens no CloudWatch \(p. 1076\)](#)

Métricas de armazenamento diárias do Amazon S3 para buckets no CloudWatch

O namespace AWS/S3 inclui as métricas de armazenamento diário a seguir para buckets.

Métrica	Descrição
<code>BucketSizeBytes</code>	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), S3 Glacier Instant Retrieval, Deep Archive Storage (S3 Glacier Deep Archive) ou S3 Glacier Flexible Retrieval (GLACIER). O valor é calculado somando o tamanho de todos os objetos e metadados no bucket (objetos atuais e não atuais), incluindo o tamanho de todas as partes de todos os uploads fracionados incompletos do bucket. Filtros de tipo de armazenamento válidos: <code>StandardStorage</code> , <code>IntelligentTieringFAStorage</code> , <code>IntelligentTieringIAStorage</code> , <code>IntelligentTieringAAStorage</code> , <code>IntelligentTieringAIAStorage</code> , <code>IntelligentTieringDAAStorage</code> , <code>StandardIAStorage</code> , <code>StandardIASizeOverhead</code> , <code>StandardIAObjectOverhead</code> , <code>OneZoneIAStorage</code> , <code>OneZoneIASizeOverhead</code> , <code>ReducedRedundancyStorage</code> , <code>GlacierInstantRetrievalSizeOverhead</code> , <code>GlacierInstantRetrievalStorage</code> , <code>GlacierStorage</code> , <code>GlacierStagingStorage</code> , <code>GlacierObjectOverhead</code> , <code>GlacierS3ObjectOverhead</code> , <code>DeepArchiveStorage</code> , <code>DeepArchiveObjectOverhead</code> , <code>DeepArchiveS3ObjectOverhead</code> e <code>DeepArchiveStagingStorage</code> (consulte a dimensão <code>StorageType</code>) Unidade: bytes Estatística válida: média
<code>NumberOfObjects</code>	O número total de objetos armazenados em um bucket para todas as classes de armazenamento. O valor é calculado contando todos os objetos do bucket (objetos atuais e não atuais) e o número total de partes de todos os multipart uploads incompletos do bucket.

Métrica	Descrição
	<p>Filtros de tipo de armazenamento válidos: <code>AllStorageTypes</code> (consulte a dimensão <code>StorageType</code>)</p> <p>Unidade: contagem</p> <p>Estatística válida: média</p>

Métricas de solicitação do Amazon S3 no CloudWatch

O namespace AWS/S3 inclui as métricas de solicitação a seguir.

Métrica	Descrição
<code>AllRequests</code>	<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 que atendam aos requisitos do filtro.</p> <p>Unidade: contagem</p> <p>Estatística válida: soma</p>
<code>GetRequests</code>	<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>Note</p> <p>Solicitações paginadas orientadas a listas, como Listar multipart uploads, Listar partes, Obter versões de objetos do bucket e outras, não estão incluídas nessa métrica.</p>
<code>PutRequests</code>	<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>
<code>DeleteRequests</code>	<p>O número de solicitações HTTP DELETE feitas para objetos em um bucket do Amazon S3. Isso também inclui solicitações Excluir vários objetos. Essa métrica exibe o número de solicitações, não o número de objetos excluídos.</p> <p>Unidade: contagem</p> <p>Estatística válida: soma</p>
<code>HeadRequests</code>	<p>O número de solicitações HTTP HEAD feitas para um bucket do Amazon S3.</p> <p>Unidade: contagem</p> <p>Estatística válida: soma</p>
<code>PostRequests</code>	<p>O número de solicitações HTTP POST feitas para um bucket do Amazon S3.</p>

Métrica	Descrição
	<p>Unidade: contagem</p> <p>Estatística válida: soma</p> <p>Note</p> <p>As solicitações Excluir vários objetos e Conteúdo de objetos do SELECT não estão incluídas nessa métrica.</p>
SelectRequests	<p>O número de solicitações Conteúdo de objetos do SELECT do Amazon S3 feitas para os objetos em um bucket do Amazon S3.</p> <p>Unidade: contagem</p> <p>Estatística válida: soma</p>
SelectBytesScanned	<p>Número de bytes de dados verificados com solicitações Conteúdo de objetos do SELECT do Amazon S3 feitas em um bucket do Amazon S3.</p> <p>Unidade: bytes</p> <p>Estatísticas válidas: média (bytes por solicitação), soma (bytes por período), contagem de amostras, mín., máx. (o mesmo que p100), qualquer percentil entre p0,0 e p99,9</p>
SelectBytesReturned	<p>Número de bytes de dados retornados com solicitações Conteúdo de objetos do SELECT do Amazon S3 feitas em um bucket do Amazon S3.</p> <p>Unidade: bytes</p> <p>Estatísticas válidas: média (bytes por solicitação), soma (bytes por período), contagem de amostras, mín., máx. (o mesmo que p100), qualquer percentil entre p0,0 e p99,9</p>
ListRequests	<p>O número de solicitações HTTP que listam o conteúdo de um bucket.</p> <p>Unidade: contagem</p> <p>Estatística válida: soma</p>
BytesDownloaded	<p>O número de bytes baixados para solicitações feitas para um bucket do Amazon S3, em que a resposta inclui um corpo.</p> <p>Unidade: bytes</p> <p>Estatísticas válidas: média (bytes por solicitação), soma (bytes por período), contagem de amostras, mín., máx. (o mesmo que p100), qualquer percentil entre p0,0 e p99,9</p>
BytesUploaded	<p>O número de bytes que contêm um corpo de solicitação, carregados para um bucket do Amazon S3.</p> <p>Unidade: bytes</p> <p>Estatísticas válidas: média (bytes por solicitação), soma (bytes por período), contagem de amostras, mín., máx. (o mesmo que p100), qualquer percentil entre p0,0 e p99,9</p>

Métrica	Descrição
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
5xxErrors	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. Unidade: contagens Estatísticas válidas: média (relatórios por solicitação), soma (relatórios por período), mín., máx., contagem de amostras
FirstByteLatency	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. Unidade: milissegundos Estatísticas válidas: média, soma, mín., máx. (o mesmo que p100), contagem de amostras, qualquer percentil entre p0,0 e p100
TotalRequestLatency	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> . Unidade: milissegundos Estatísticas válidas: média, soma, mín., máx. (o mesmo que p100), contagem de amostras, qualquer percentil entre p0,0 e p100

Métricas de replicação do Amazon S3 no CloudWatch

Você pode monitorar o progresso da replicação com métricas de replicação do S3 rastreando bytes pendentes, operações pendentes e latência de replicação. Para obter mais informações, consulte [Monitoramento do progresso com métricas de replicação](#).

Note

Você pode habilitar alarmes para suas métricas de replicação no Amazon CloudWatch. Quando você configurar alarmes para as métricas de replicação, defina o campo Missing data treatment (Tratamento de dados ausentes) como Treat missing data as ignore (maintain the alarm state) (Ignorar dados ausentes [manter o estado de alarme]).

Métrica	Descrição
ReplicationLatency	O número máximo de segundos em que a região de destino da replicação está atrás da região de origem de uma determinada regra de replicação.

Métrica	Descrição
	Unidade: segundos Estatísticas válidas: Max
BytesPendingReplicat	O número total de bytes de objetos com replicação pendente para uma determinada regra de replicação. Unidade: bytes Estatísticas válidas: Max
OperationsPendingReplicat	O número de operações com replicação pendente para uma determinada regra de replicação. Unidade: contagens Estatísticas válidas: Max

Métricas do Amazon S3 Storage Lens no CloudWatch

É possível publicar métricas de uso e atividade do S3 Storage Lens no Amazon CloudWatch para criar uma visão unificada de sua integridade operacional nos [painéis](#) do CloudWatch. As métricas da lente de armazenamento do S3 são publicadas no namespace AWS/S3/Storage-Lens no CloudWatch. A opção de publicação do CloudWatch está disponível para painéis do S3 Storage Lens atualizados para métricas e recomendações avançadas.

Para obter uma lista de métricas do S3 Storage Lens publicadas no CloudWatch, consulte [Glossário de métricas de lente de armazenamento do Amazon S3 \(p. 1152\)](#). Para obter uma lista completa de dimensões, consulte [Dimensões \(p. 1140\)](#).

Métricas do Amazon S3 on Outposts no CloudWatch

O namespace S3Outposts inclui as seguintes métricas para buckets do Amazon S3 em Outposts. É possível monitorar o número total de bytes provisionados do S3 em Outposts, o total de bytes livres disponíveis para objetos e o tamanho total de todos os objetos de determinado bucket.

Note

O S3 on Outposts apenas é compatível com as métricas do Amazon S3 a seguir.

Como o S3 on Outposts tem capacidade limitada, é possível criar alertas do CloudWatch que avisam quando a utilização do armazenamento excede um limite.

Métrica	Descrição
OutpostTotalBytes	A capacidade provisionada total em bytes para um Outpost. Unidade: bytes Período: 5 minutos
OutpostFreeBytes	A contagem de bytes livres disponíveis em um Outpost para armazenar dados de clientes. Unidade: bytes Período: 5 minutos

Métrica	Descrição
<code>BucketUsedBytes</code>	O tamanho total de todos os objetos de determinado bucket. Unidade: contagens Período: 5 minutos
<code>AccountUsedBytes</code>	O tamanho total de todos os objetos da conta do Outposts especificada. Unidade: bytes Período: 5 minutos

Dimensões do Amazon S3 no CloudWatch

As seguintes dimensões são usadas para filtrar as métricas do Amazon S3.

Dimensão	Descrição
<code>BucketName</code>	Essa dimensão filtra os dados que você solicita apenas para o bucket identificado.
<code>StorageType</code>	Essa dimensão filtra os dados que você armazenou em um bucket pelos seguintes tipos de armazenamento: <ul style="list-style-type: none"> <code>StandardStorage</code>: o número de bytes usados para objetos na classe de armazenamento STANDARD. <code>IntelligentTieringAAStorage</code>: o número de bytes usados para objetos no nível de acesso ao arquivo da classe de armazenamento INTELLIGENT_TIERING. <code>IntelligentTieringAIAStorage</code>: o número de bytes usados para objetos no nível de acesso ao arquivo instantâneo da classe de armazenamento INTELLIGENT_TIERING. <code>IntelligentTieringDAStorage</code>: o número de bytes usados para objetos no nível de acesso ao arquivo profundo da classe de armazenamento INTELLIGENT_TIERING. <code>IntelligentTieringFASTorage</code>: o número de bytes usados para objetos no nível de acesso frequente da classe de armazenamento INTELLIGENT_TIERING. <code>IntelligentTieringIAStorage</code>: o número de bytes usados para objetos no nível de acesso pouco frequente da classe de armazenamento INTELLIGENT_TIERING. <code>StandardIAStorage</code>: o número de bytes usados para objetos no nível de acesso pouco frequente da classe de armazenamento padrão (STANDARD_IA). <code>StandardIASizeOverhead</code>: o número de bytes usados para objetos com menos de 128 KB na classe de armazenamento STANDARD_IA. <code>IntAAObjectOverhead</code>: para cada objeto na classe de armazenamento INTELLIGENT_TIERING no nível de acesso ao arquivo, o S3 Glacier adiciona 32 KB de armazenamento para índice e metadados relacionados. Esses dados extras são

Dimensão	Descrição
	<p>necessários para identificar e recuperar seu objeto. Taxas do S3 Glacier são cobradas por esse armazenamento adicional.</p> <ul style="list-style-type: none"> • IntAAS3ObjectOverhead: para cada objeto na classe de armazenamento INTELLIGENT_TIERING no nível de acesso ao arquivo, o Simple Storage Service (Amazon S3) usa 8 KB de armazenamento para o nome do objeto e outros metadados. Você é cobrado pelas taxas de STANDARD nesse armazenamento adicional. • IntDAAObjectOverhead: para cada objeto na classe de armazenamento INTELLIGENT_TIERING no nível de acesso ao arquivo profundo, o S3 Glacier adiciona 32 KB de armazenamento para índice e metadados relacionados. Esses dados extras são necessários para identificar e recuperar seu objeto. Você é cobrado pelas taxas do S3 Glacier Deep Archive nesse armazenamento adicional. • IntDAAS3ObjectOverhead: para cada objeto na classe de armazenamento INTELLIGENT_TIERING no nível de acesso ao arquivo profundo, o Simple Storage Service (Amazon S3) adiciona 8 KB de armazenamento para índice e metadados relacionados. Esses dados extras são necessários para identificar e recuperar seu objeto. Você é cobrado pelas taxas de STANDARD nesse armazenamento adicional. • OneZoneIASStorage: o número de bytes usados para objetos na classe de armazenamento S3 One Zone de acesso pouco frequente (ONEZONE_IA). • OneZoneIASizeOverhead: o número de bytes usados para objetos com menos de 128 KB na classe de armazenamento ONEZONE_IA. • ReducedRedundancyStorage: o número de bytes usados para objetos na classe de armazenamento de redundância reduzida (RRS - Reduced Redundancy Storage). • GlacierInstantRetrievalSizeOverhead: o número de bytes usados para objetos com menos de 128 KB na classe de armazenamento Instant Retrieval do S3 Glacier. • GlacierInstantRetrievalStorage: o número de bytes usados para objetos na classe de armazenamento Instant Retrieval do S3 Glacier. • GlacierStorage: o número de bytes usados para objetos na classe de armazenamento S3 Glacier Flexible Retrieval. • GlacierStagingStorage: o número de bytes usados para as partes de objetos fracionados antes que a solicitação CompleteMultipartUpload seja concluída em objetos na classe de armazenamento S3 Glacier Flexible Retrieval. • GlacierObjectOverhead: para cada objeto arquivado, o S3 Glacier adiciona 32 KB de armazenamento para índice e metadados relacionados. Esses dados extras são necessários para identificar e recuperar seu objeto. Você é cobrado nas tarifas do S3 Glacier Flexible Retrieval por esse armazenamento adicional. • GlacierS3ObjectOverhead: para cada objeto arquivado no S3 Glacier Flexible Retrieval, o Simple Storage Service (Amazon S3) usa 8 KB de armazenamento para o nome do objeto e outros

Dimensão	Descrição
	<p>metadados. Você é cobrado pelas taxas de STANDARD nesse armazenamento adicional.</p> <ul style="list-style-type: none"> • DeepArchiveStorage: o número de bytes usados para objetos na classe de armazenamento S3 Glacier Deep Archive. • DeepArchiveObjectOverhead: para cada objeto arquivado, o S3 Glacier adiciona 32 KB de armazenamento para índice e metadados relacionados. Esses dados extras são necessários para identificar e recuperar seu objeto. Você é cobrado pelas taxas do S3 Glacier Deep Archive nesse armazenamento adicional. • DeepArchiveS3ObjectOverhead: para cada objeto arquivado no S3 Glacier Deep Archive, o Simple Storage Service (Amazon S3) usa 8 KB de armazenamento para o nome do objeto e outros metadados. Você é cobrado pelas taxas de STANDARD nesse armazenamento adicional. • DeepArchiveStagingStorage: o número de bytes usados para partes de objetos Multipart antes de a solicitação CompleteMultipartUpload seja realizada em objetos na classe de armazenamento S3 Glacier Deep Archive.
FilterId	Esta dimensão filtra as configurações de métricas que você especifica para métricas de solicitação em um bucket, por exemplo, um prefixo ou uma etiqueta. Você especifica um ID de filtro ao criar uma configuração de métricas. Para obter mais informações, consulte Criar uma configuração de métricas .

Dimensões do Amazon S3 Storage Lens no CloudWatch

Para obter uma lista das dimensões usadas para filtrar métricas do S3 Storage Lens no CloudWatch, consulte [Dimensões \(p. 1140\)](#).

Acessar métricas do CloudWatch

Você pode usar os procedimentos a seguir para visualizar as métricas de armazenamento para o Amazon S3. Para obter as métricas do Amazon S3 envolvidas, é necessário definir um timestamp 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.

Uso do AWS CLI

Por exemplo, se você usar a AWS CLI para obter a média do tamanho de um bucket específico em bytes, é possível usar o comando a seguir.

```
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 saída a seguir.

```
{
  "Datapoints": [
```

```
{  
    "Timestamp": "2016-10-19T00:00:00Z",  
    "Average": 1025328.0,  
    "Unit": "Bytes"  
}  
,  
"Label": "BucketSizeBytes"  
}
```

Uso do console do S3

Como exibir 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, insira o nome da métrica na caixa de pesquisa.
5. (Opcional) Para filtrar pela dimensão StorageType, insira o nome de classe de armazenamento na caixa de pesquisa.

Para ver uma lista de métricas válidas armazenadas para sua Conta da AWS usando a AWS CLI

- Em um prompt de comando, use o seguinte comando.

```
aws cloudwatch list-metrics --namespace "AWS/S3"
```

Configurações de métricas do CloudWatch

Com as métricas de solicitação do Amazon CloudWatch para o Amazon S3, você pode receber métricas do CloudWatch de 1 minuto, definir alarmes do CloudWatch e acessar painéis do CloudWatch para visualizar operações quase em tempo real e performance do armazenamento do Amazon S3. Para aplicativos que dependem do 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.

Se quiser obter as métricas de solicitações do CloudWatch para objetos em um bucket, será necessário criar uma configuração de métricas para o bucket. Para obter mais informações, consulte [Criar uma configuração de métricas do CloudWatch para todos os objetos em seu bucket \(p. 1078\)](#).

Você também pode usar um prefixo compartilhado, etiquetas do objeto ou um ponto de acesso para definir um filtro para as métricas coletadas. Esse método de definir um filtro permite que você alinhe filtros de métricas para aplicações de negócios, fluxos de trabalho ou organizações internas específicas. Para obter mais informações, consulte [Criação de uma configuração de métricas que filtre por prefixo, etiqueta de objeto ou ponto de acesso \(p. 1080\)](#). Para obter mais informações sobre as métricas do CloudWatch que estão disponíveis e as diferenças entre métricas de armazenamento e métricas de solicitação, consulte [Monitoramento de métricas com o Amazon CloudWatch \(p. 1067\)](#).

Tenha o seguinte em mente ao usar as configurações de métricas:

- Você pode ter um máximo de 1.000 configurações de métricas por bucket.
- Você pode escolher que objetos em um bucket serão incluídos nas configurações de métricas, utilizando filtros. Você pode filtrar em um prefixo compartilhado, etiqueta de objeto ou ponto de acesso para alinhar filtros de métricas para aplicações de negócios, fluxos de trabalho ou organizações internas específicas. 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 \(p. 1070\)](#). As métricas específicas de operações (como `PostRequests`) serão relatadas somente se houver solicitações daquele tipo para o bucket ou filtro.
- As métricas de solicitações são relatadas para todas as operações no nível de objeto. Elas também são informadas para operações que listam o conteúdo do bucket, como [GET Bucket \(List Objects\)](#) [Obter Bucket(listar objetos)], [GET Bucket Object Versions](#) (Obter versões de objeto do bucket) e [List Multipart Uploads](#) (Listar carregamentos fracionados), mas não são informadas para outras operações em buckets.
- As métricas de solicitações oferecem suporte à filtragem por prefixo, etiquetas do objeto ou ponto de acesso, mas as métricas de armazenamento não.

Entrega de métricas do CloudWatch com o melhor esforço

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 integridade e pontualidade das métricas não são garantidas. O ponto de dados para uma solicitação específica pode ser retornado com um timestamp posterior à solicitação processada. O ponto de dados de um minuto pode ser atrasado antes de ser disponibilizado pelo CloudWatch ou pode nem ser entregue. As métricas de solicitação do CloudWatch lhe dão uma ideia da natureza do tráfego em seu bucket em tempo quase 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.

Para obter mais informações sobre como trabalhar com métricas do CloudWatch no Amazon S3, consulte os tópicos a seguir.

Tópicos

- [Criar uma configuração de métricas do CloudWatch para todos os objetos em seu bucket \(p. 1078\)](#)
- [Criação de uma configuração de métricas que filtre por prefixo, etiqueta de objeto ou ponto de acesso \(p. 1080\)](#)
- [Excluir um filtro de métricas \(p. 1082\)](#)

Criar uma configuração de métricas do CloudWatch para todos os objetos em seu bucket

Ao configurar métricas de solicitação, você pode criar uma configuração de métricas do CloudWatch para todos os objetos no bucket ou filtrar por prefixo, etiqueta de objeto ou ponto de acesso. Os procedimentos neste tópico mostram como criar uma configuração para todos os objetos em seu bucket. Para criar uma configuração que filtre por etiqueta de objeto ou prefixo ou ponto de acesso, consulte [Criação de uma configuração de métricas que filtre por prefixo, etiqueta de objeto ou ponto de acesso \(p. 1080\)](#).

Existem três tipos de métricas do Amazon CloudWatch para o Amazon S3: métricas de armazenamento, métricas de solicitação e métricas de replicação. As métricas de armazenamento são relatadas uma vez por dia e fornecidas para todos os clientes sem qualquer custo adicional. As métricas de solicitação estão disponíveis em intervalos de um minuto após alguma latência para processamento. As métricas

de solicitação são cobradas na taxa padrão do CloudWatch. Você deve aceitar métricas de solicitação configurando-as no console ou usando a API do Amazon S3.

Para obter mais informações sobre métricas do CloudWatch para o Amazon S3, consulte [Monitoramento de métricas com o Amazon CloudWatch \(p. 1067\)](#).

Você pode adicionar configurações de métricas a um bucket usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI) ou a API REST do Amazon S3.

Uso do console do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, selecione o nome do bucket que contém os objetos para os quais você deseja solicitar métricas.
3. Escolha a guia Metrics.
4. Em Bucket metrics (Métricas de bucket), escolha View additional charts (Exibir gráficos adicionais).
5. Escolha a guia Request metrics (Solicitar métricas) .
6. Escolha Create Filter (Criar filtro).
7. Na caixa Filter name (Nome do filtro) insira o nome do filtro.

Nomes podem conter apenas letras, números, pontos, traços e sublinhados. Recomendamos usar o nome `EntireBucket` para um filtro que se aplica a todos os objetos.

8. Em Filter scope (Escopo do filtro), escolha This filter applies to all objects in the bucket (Este filtro se aplica a todos os objetos no bucket).

Você também pode definir um filtro para que as métricas sejam coletadas e relatadas apenas em um subconjunto de objetos no bucket. Para obter mais informações, consulte [Criação de uma configuração de métricas que filtre por prefixo, etiqueta de objeto ou ponto de acesso \(p. 1080\)](#).

9. Selecione Save changes.
10. Na guia Request metrics (métricas de solicitação), em Filters (Filtros), escolha o filtro que você acabou de criar.

Após cerca de 15 minutos, o CloudWatch começa a rastrear essas métricas de solicitação. Você pode vê-las na guia Request metrics (Métricas de solicitação) . Você pode ver gráficos para as métricas no console do Amazon S3 ou do CloudWatch. As métricas de solicitação são cobradas na taxa padrão do CloudWatch. Para obter mais informações, consulte [Preço do Amazon CloudWatch](#).

Uso dos REST API

Você também pode adicionar configurações de métricas de maneira programática com a API REST do Amazon S3. Para obter mais informações sobre como adicionar e trabalhar com configurações de métricas, consulte os tópicos a seguir na Referência da API do Amazon Simple Storage Service:

- [Configuração de métrica PUT Bucket](#)
- [Configuração de métrica GET Bucket](#)
- [Configuração de métrica List Bucket](#)
- [Configuração de métrica DELETE Bucket](#)

Uso do AWS CLI

1. Instalar e configurar a AWS CLI. Para obter instruções, consulte [Instalação, atualização e desinstalação da AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

2. Abra um terminal.
3. Execute o comando a seguir para adicionar uma configuração de métricas.

```
aws s3api put-bucket-metrics-configuration --endpoint https://s3.us-west-2.amazonaws.com --bucket bucket-name --id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id"}'
```

Criação de uma configuração de métricas que filtre por prefixo, etiqueta de objeto ou ponto de acesso

Existem três tipos de métricas do Amazon CloudWatch para o Amazon S3: métricas de armazenamento, métricas de solicitação e métricas de replicação. As métricas de armazenamento são relatadas uma vez por dia e fornecidas para todos os clientes sem qualquer custo adicional. As métricas de solicitação estão disponíveis em intervalos de um minuto após alguma latência para processamento. As métricas de solicitação são cobradas na taxa padrão do CloudWatch. Você deve aceitar métricas de solicitação configurando-as no console ou usando a API do Amazon S3.

Para obter mais informações sobre métricas do CloudWatch para o Amazon S3, consulte [Monitoramento de métricas com o Amazon CloudWatch \(p. 1067\)](#).

Ao configurar métricas do CloudWatch, você pode criar um filtro para todos os objetos em seu bucket ou 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 tipos de filtros:

- 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. Para obter mais informações sobre prefixos, consulte [Organizar objetos usando prefixos \(p. 258\)](#).
- Etiqueta: você pode adicionar etiquetas que são pares de nome chave-valor, aos objetos. As tags ajudam a encontrar e organizar objetos com facilidade. Você também pode usar etiquetas como filtros para configurações de métricas. Para obter mais informações sobre etiquetas de objeto, consulte [Categorizando seu armazenamento usando tags \(p. 878\)](#).
- Ponto de acesso: os pontos de acesso são nomeados endpoints de rede que estão conectados a buckets e simplificam o gerenciamento do acesso a dados em escala para conjuntos de dados compartilhados no S3. Quando você cria um filtro de ponto de acesso, o Amazon S3 inclui solicitações ao ponto de acesso que você especifica na configuração de métricas. Para obter mais informações, consulte [Monitorar e registrar de pontos de acesso \(p. 330\)](#).

Note

Ao criar uma configuração de métricas que filtra por ponto de acesso, você deve usar o ponto de acesso Nome de recurso da Amazon (ARN), não o alias do ponto de acesso. Certifique-se de usar o ARN para o próprio ponto de acesso, não o ARN para um objeto específico. Para obter mais informações sobre ARNs de ponto de acesso, consulte [Usar pontos de acesso \(p. 329\)](#).

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 [List](#) 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.

Uso do console do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista de buckets, selecione o nome do bucket que contém os objetos para os quais você deseja solicitar métricas.
3. Escolha a guia Metrics.
4. Em Bucket metrics (Métricas de bucket), escolha View additional charts (Exibir gráficos adicionais).
5. Escolha a guia Request metrics (Solicitar métricas) .
6. Escolha Create Filter (Criar filtro).
7. Na caixa Filter name (Nome do filtro), insira o nome do filtro.

Nomes podem conter apenas letras, números, pontos, traços e sublinhados.
8. Em Filter scope (Escopo do filtro), escolha Limit the scope of this filter using a prefix, object tags, and an S3 Access Point, or a combination of all three (Limitar o escopo desse filtro usando um prefixo, etiquetas de objeto e um ponto de acesso do S3 ou uma combinação de todos os três).
9. Em Filter type (Tipo de filtro), escolha pelo menos um tipo de filtro: Prefix (Prefixo), Object tags (Etiquetas do objeto) ou Access Point (Ponto de acesso).
10. Para definir um filtro de prefixo e limitar o escopo do filtro a um único caminho, na caixa Prefix (Prefixo), insira um prefixo.
11. Para definir um filtro de etiquetas do objeto, em Object tags (Etiquetas do objeto), selecione Add tag (Adicionar etiqueta) e, em seguida, insira uma Key (Chave) e Value (Valor) da etiqueta.
12. Para definir um filtro de ponto de acesso, no campo S3 Access Point (Ponto de acesso do S3), informe o ARN do ponto de acesso ou escolha Browse S3 (Navegar no S3) para navegar até o ponto de acesso.

Important

Não é possível inserir um alias de ponto de acesso. Você deve inserir o ARN para o próprio ponto de acesso, não o ARN de um objeto específico.

13. Selecione Save changes.

O Amazon S3 cria um filtro que usa o prefixo, as etiquetas ou o ponto de acesso especificado.

14. Na guia Request metrics (métricas de solicitação), em Filters (Filtros), escolha o filtro que você acabou de criar.

Agora você criou um filtro que limita o escopo de métricas de solicitação por prefixo, etiquetas de objeto ou ponto de acesso. Cerca de 15 minutos após o CloudWatch começar a rastrear essas métricas de solicitação, você pode ver gráficos para as métricas nos consoles do Amazon S3 e do CloudWatch. As métricas de solicitação são cobradas na taxa padrão do CloudWatch. Para obter mais informações, consulte [Definição de preços do Amazon CloudWatch](#).

Você também pode configurar métricas de solicitação no nível de bucket. Para obter mais informações, consulte [Criar uma configuração de métricas do CloudWatch para todos os objetos em seu bucket \(p. 1078\)](#).

Uso dos REST API

Você também pode adicionar configurações de métricas de maneira programática com a API REST do Amazon S3. Para obter mais informações sobre como adicionar e trabalhar com configurações de métricas, consulte os tópicos a seguir na Referência da API do Amazon Simple Storage Service:

- [Configuração de métrica PUT Bucket](#)
- [Configuração de métrica GET Bucket](#)

- Configuração de métrica List Bucket
- Configuração de métrica DELETE Bucket

Uso do AWS CLI

1. Instalar e configurar a AWS CLI. Para obter instruções, consulte [Instalação, atualização e desinstalação da AWS CLI](#) no Guia do usuário da AWS Command Line Interface.
2. Abra um terminal.
3. Para adicionar uma configuração de métricas, execute um dos seguintes comandos:

Example : para filtrar por prefixo

```
aws s3api put-bucket-metrics-configuration --bucket DOC-EXAMPLE-BUCKET1 --  
id metrics-config-id --metrics-configuration '{"Id":"'metrics-config-id', "Filter":  
{"Prefix":"prefix1"}}
```

Example : para filtrar por etiquetas

```
aws s3api put-bucket-metrics-configuration --bucket DOC-EXAMPLE-BUCKET1 --id metrics-  
config-id --metrics-configuration '{"Id":"'metrics-config-id', "Filter": {"Tag": {"Key":  
"string", "Value": "string"}}}'
```

Example : para filtrar por ponto de acesso

```
aws s3api put-bucket-metrics-configuration --bucket DOC-EXAMPLE-BUCKET1 --  
id metrics-config-id --metrics-configuration '{"Id":"'metrics-config-id', "Filter":  
{"AccessPointArn": "arn:aws:s3:Region:account-id:accesspoint/access-point-name"}'}
```

Example : para filtrar por prefixo, etiquetas e ponto de acesso

```
aws s3api put-bucket-metrics-configuration --endpoint https://s3.Region.amazonaws.com  
--bucket DOC-EXAMPLE-BUCKET1 --id metrics-config-id --metrics-configuration '  
{  
    "Id": "metrics-config-id",  
    "Filter": {  
        "And": {  
            "Prefix": "string",  
            "Tags": [  
                {  
                    "Key": "string",  
                    "Value": "string"  
                }  
            ],  
            "AccessPointArn": "arn:aws:s3:Region:account-id:accesspoint/access-point-  
name"  
        }  
    }  
}'
```

Excluir um filtro de métricas

Você pode excluir um filtro de métricas de solicitação do Amazon CloudWatch se não precisar mais dele. Ao excluir um filtro, você não será mais cobrado pelas métricas de solicitação que usam esse filtro específico. No entanto, você continuará a ser cobrado por qualquer outra configuração de filtro existente.

Ao excluir um filtro, você não poderá mais usar o filtro para métricas de solicitação. A exclusão de um filtro não pode ser desfeita.

Para obter informações sobre como criar um filtro de métricas de solicitação, consulte os seguintes tópicos:

- [Criar uma configuração de métricas do CloudWatch para todos os objetos em seu bucket \(p. 1078\)](#)
- [Criação de uma configuração de métricas que filtre por prefixo, etiqueta de objeto ou ponto de acesso \(p. 1080\)](#)

Uso do console do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket.
3. Escolha a guia Metrics.
4. Em Bucket metrics (Métricas de bucket), escolha View additional charts (Exibir gráficos adicionais).
5. Escolha a guia Request metrics (Solicitar métricas) .
6. Escolha Manage filters (Gerenciar filtros).
7. Escolha o seu filtro.

Important

A exclusão de um filtro não pode ser desfeita.

8. Escolha Delete (Excluir).

O Amazon S3 exclui seu filtro.

Uso dos REST API

Você também pode adicionar configurações de métricas de maneira programática com a API REST do Amazon S3. Para obter mais informações sobre como adicionar e trabalhar com configurações de métricas, consulte os tópicos a seguir na Referência da API do Amazon Simple Storage Service:

- [Configuração de métrica PUT Bucket](#)
- [Configuração de métrica GET Bucket](#)
- [Configuração de métrica List Bucket](#)
- [Configuração de métrica DELETE Bucket](#)

Notificações de eventos do Amazon S3

Você pode usar o recurso Notificações de eventos do Amazon S3 para receber notificações quando determinados eventos acontecerem no bucket do S3. Para habilitar notificações, adicione uma configuração de notificação que identifique os eventos que deseja que o Amazon S3 publique. Verifique se ele também identifica os destinos para os quais você deseja que o Amazon S3 envie as notificações. Armazene essa configuração no sub-recurso notification associado a um bucket. Para obter mais informações, consulte [Opções de configuração do bucket \(p. 125\)](#). O Amazon S3 fornece uma API para gerenciamento desse sub-recurso.

Important

As notificações de eventos do Amazon S3 são projetadas para serem entregues pelo menos uma vez. Em geral, as notificações de eventos são entregues em segundos, mas, às vezes, podem levar um minuto ou mais.

Visão geral das notificações de eventos do Amazon S3

Atualmente, o Amazon S3 pode publicar notificações para os seguintes eventos:

- Eventos de criação de novos objetos
- Eventos de remoção de objetos
- Eventos de restauração de objetos
- Eventos de perda de objeto com Reduced Redundancy Storage (RRS)
- Eventos de replicação
- Eventos de validade do S3 Lifecycle
- Eventos de transição do S3 Lifecycle
- Eventos de arquivamento automático do S3 Intelligent-Tiering
- Eventos de marcação de objetos
- Eventos PUT de ACL de objeto

Para obter descrições completas de todos os tipos de evento compatíveis, consulte [Tipos de evento compatíveis com SQS, SNS e Lambda \(p. 1087\)](#).

O Amazon S3 pode enviar mensagens de notificação de eventos aos seguintes destinos. Você especifica o valor do nome de recurso da Amazon (ARN) desses destinos na configuração de notificação.

- Amazon Simple Notification Service (Amazon SNS) topics
- Filas do Amazon Simple Queue Service (Amazon SQS)
- Função AWS Lambda

Para obter mais informações, consulte [Destinos de eventos suportados \(p. 1085\)](#).

Warning

Se a notificação for gravada no mesmo bucket que aciona a notificação, isso poderá causar um loop de execução. Por exemplo, se o bucket acionar uma função do Lambda toda vez que houver um upload de objeto, e a função fizer upload de um objeto no bucket, a função será acionada indiretamente. Para evitar isso, use dois buckets ou configure o trigger para só se aplicar a um prefixo usado em objetos recebidos.

Para obter mais informações e um exemplo de uso de notificações do Amazon S3 com AWS Lambda, consulte [Uso do AWS Lambda com o Amazon S3](#) no Guia do desenvolvedor do AWS Lambda.

Para obter mais informações sobre o número de configurações de notificação de eventos que você pode criar por bucket, consulte [Cotas de serviço do Amazon S3](#) na Referência geral da AWS.

Para obter mais informações sobre notificações de eventos do S3, consulte as seções a seguir.

Tópicos

- [Tipos e destinos de notificações de evento \(p. 1085\)](#)
- [Usar o Amazon SQS, o Amazon SNS e o Lambda \(p. 1089\)](#)
- [Uso do EventBridge \(p. 1108\)](#)

Tipos e destinos de notificações de evento

O Amazon S3 oferece suporte a vários tipos de notificação de eventos e destinos nos quais as notificações podem ser publicadas. Você pode especificar o tipo de evento e o destino ao configurar suas notificações de eventos.

Tópicos

- [Destinos de eventos suportados \(p. 1085\)](#)
- [Tipos de evento compatíveis com SQS, SNS e Lambda \(p. 1087\)](#)
- [Tipos de evento compatíveis com o Amazon EventBridge \(p. 1089\)](#)

Destinos de eventos suportados

O Amazon S3 pode enviar mensagens de notificação de eventos aos seguintes destinos.

- Amazon Simple Notification Service (Amazon SNS) topics
- Filas do Amazon Simple Queue Service (Amazon SQS)
- AWS Lambda
- Amazon EventBridge

Note

Você deve conceder ao Amazon S3 permissões 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 instruções sobre como conceder essas permissões, consulte [Conceder permissões para publicar mensagens de notificação de evento a um destino \(p. 1090\)](#).

Tópico do Amazon SNS

O Amazon SNS é um serviço de mensagens por push flexível e totalmente gerenciado. Você pode usar esse serviço para enviar mensagens por push para dispositivos móveis ou serviços distribuídos. Com o SNS, você pode publicar uma mensagem uma vez e entregá-la uma ou mais vezes. Atualmente, o SNS padrão só é permitido como um destino de notificação de eventos do S3, enquanto o SNS FIFO não é permitido.

O Amazon SNS coordena e gerencia o envio e a entrega de mensagens para endpoints ou clientes assinantes. Você pode usar o console do Amazon SNS para criar um tópico do Amazon SNS para o qual suas notificações podem ser enviadas.

O tópico deve estar na mesma Região da AWS que o bucket do Amazon S3. Para obter instruções sobre como criar um tópico do Amazon SNS, consulte [Conceitos básicos do Amazon SNS](#) no Guia do desenvolvedor do Amazon Simple Notification Service e as [Perguntas frequentes do Amazon SNS](#).

Antes de poder usar o tópico do Amazon SNS criado como um destino de notificação de evento, você precisará de:

- O Nome de recurso da Amazon (ARN) do tópico do Amazon SNS
- Uma assinatura válida do tópico do Amazon SNS. Com ela, os assinantes do tópico são notificados quando uma mensagem é publicada em seu tópico do Amazon SNS.

Fila do Amazon SQS

O Amazon SQS oferece filas hospedadas confiáveis e escaláveis para o armazenamento de mensagens à medida que transitam entre computadores. Você pode usar o Amazon SQS para transmitir qualquer volume de dados sem exigir que outros serviços estejam sempre disponíveis. Você pode usar o console do Amazon SQS para criar uma fila do Amazon SQS para a qual suas notificações podem ser enviadas.

A fila do Amazon SQS deve estar na mesma Região da AWS que o bucket do Amazon S3. Para obter instruções sobre como criar uma fila do Amazon SQS, consulte [O que é o Amazon Simple Queue Service e Conceitos básicos do Amazon SQS](#) no Guia do desenvolvedor do Amazon Simple Queue Service.

Antes de poder usar a fila do Amazon SQS como um destino de notificação de evento, você precisa do seguinte:

- O nome do recurso da Amazon (ARN) da fila do Amazon SQS

Função Lambda

É possível usar o AWS Lambda para estender outros produtos da AWS com lógica personalizada ou criar seu próprio backend que opere de acordo com a escala, a performance e a segurança da AWS. Com o Lambda, é possível criar aplicações discretas orientadas por eventos que são executadas somente quando necessário. Você também pode usá-lo para escalar essas aplicações automaticamente de poucas solicitações por dia a milhares por segundo.

O Lambda pode executar código personalizado em resposta a eventos do bucket do Amazon S3. Carregue o código personalizado no Lambda e crie o que é chamado de função Lambda. Quando o Amazon S3 detecta um evento de um tipo específico, ele pode publicar o evento no AWS Lambda e invocar sua função no Lambda. Em resposta, o Lambda executa a função. Um tipo de evento que ele pode detectar, por exemplo, é um evento criado por objeto.

Você pode usar o console do AWS Lambda para criar uma função do Lambda que use a infraestrutura da AWS para executar o código em seu nome. A função do Lambda deve estar na mesma região que seu bucket do S3. Você também deve ter o nome ou o ARN de uma função do Lambda para configurar a função do Lambda como um destino de notificação de evento.

Warning

Se a notificação for gravada no mesmo bucket que aciona a notificação, isso poderá causar um loop de execução. Por exemplo, se o bucket acionar uma função do Lambda toda vez que houver um upload de objeto, e a função fizer upload de um objeto no bucket, a função será acionada indiretamente. Para evitar isso, use dois buckets ou configure o trigger para só se aplicar a um prefixo usado em objetos recebidos.

Para obter mais informações e um exemplo de uso de notificações do Amazon S3 com AWS Lambda, consulte [Uso do AWS Lambda com o Amazon S3](#) no Guia do desenvolvedor do AWS Lambda.

Amazon EventBridge

O Amazon EventBridge é um barramento de eventos sem servidor, que recebe eventos de produtos da AWS. É possível configurar regras para corresponder a eventos e entregá-los a destinos, como um produto da AWS ou um endpoint de HTTP. Para obter mais informações, consulte [O que é o EventBridge](#) no Guia do usuário do Amazon EventBridge.

Ao contrário de outros destinos, você pode habilitar ou desabilitar eventos a serem entregues ao EventBridge para um bucket. Se você habilitar a entrega, todos os eventos serão enviados ao EventBridge. Além disso, você pode usar regras do EventBridge para encaminhar eventos para outros destinos.

Tipos de evento compatíveis com SQS, SNS e Lambda

O Amazon S3 pode publicar eventos dos seguintes tipos. Você especifica esses tipos de evento na configuração de notificação.

Tipos de evento	Descrição
s3:TestEvent	<p>Quando uma notificação é habilitada, o Amazon S3 publica uma notificação de teste. Isso serve para verificar se o tópico existe e se o proprietário do bucket tem permissão para publicar o tópico especificado.</p> <p>Se a ativação da notificação falhar, você não receberá a notificação de teste.</p>
s3:ObjectCreated:*	<p>As operações de API do Amazon S3, como PUT, POST e COPY, podem criar um objeto. Com esses tipos de evento, você pode habilitar notificações quando um objeto é criado usando uma operação de API específica. Se preferir, você pode usar o tipo de evento s3:ObjectCreated:* para solicitar notificação, qualquer que seja a API usada para criar um objeto.</p> <p>Você não recebe notificações de eventos de operações que falharam.</p> <p>s3:ObjectCreated:CompleteMultipartUpload inclui objetos que são criados usando UploadPartCopy para operações de cópia.</p>
s3:ObjectRemoved:*	<p>Usando os tipos de evento ObjectRemoved, você pode habilitar a notificação quando um objeto ou um lote de objetos é removido de um bucket.</p> <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 s3:ObjectRemoved:Delete. Se preferir, você pode solicitar uma notificação quando um marcador de exclusão for criado para um objeto com versionamento usando s3:ObjectRemoved:DeleteMarkerCreated. Para obter instruções sobre como excluir objetos com versionamento, consulte Excluir versões de objetos de um bucket com versionamento habilitado (p. 700). Você também pode usar um caractere curinga s3:ObjectRemoved:* para solicitar uma notificação sempre que um objeto for excluído.</p> <p>Essas notificações de eventos não alertam sobre exclusões automáticas de políticas de ciclo de vida ou operações que falharam.</p>
s3:ObjectRestore:*	<p>Com os tipos de evento ObjectRestore, você pode receber notificações de iniciação e conclusão de eventos ao restaurar objetos das classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive. Você também pode receber notificações quando a cópia restaurada de um objeto expirar.</p>

Tipos de evento	Descrição
	O tipo de evento <code>s3:ObjectRestore:Post</code> notifica você sobre o início da restauração de objetos. O tipo de evento <code>s3:ObjectRestore:Completed</code> notifica você sobre a conclusão da restauração. O tipo de evento <code>s3:ObjectRestore:Delete</code> notifica você quando a cópia temporária de um objeto restaurado expira.
<code>s3:ReducedRedundancyLostObject</code>	Você recebe esse evento de notificação quando o Amazon S3 detecta que um objeto da classe de armazenamento RRS foi perdido.
<code>s3:Replication:*</code> <code>s3:Replication:OperationFailedReplication</code> <code>s3:Replication:OperationMissedThreshold</code> <code>s3:Replication:OperationReplicatedAfterThreshold</code> <code>s3:Replication:OperationNotTracked</code>	<p>Com tipos de evento Replication, você pode receber notificações de eventos para configurações de replicação que tenham métricas de replicação do S3 ou o S3 Replication Time Control (S3 RTC) habilitado. Você pode monitorar o progresso minuto a minuto de eventos de replicação, rastreando bytes pendentes, operações pendentes e latência de replicação. Para obter informações sobre métricas de replicação, consulte Monitoramento do progresso com métricas de replicação e notificações de eventos do Amazon S3 (p. 857).</p> <p>O tipo de evento <code>s3:Replication:OperationFailedReplication</code> notifica você quando um objeto qualificado para replicação falhou ao replicar. O tipo de evento <code>s3:Replication:OperationMissedThreshold</code> notifica você quando um objeto qualificado para replicação excede o limite de 15 minutos para replicação.</p> <p>O tipo de evento <code>s3:Replication:OperationReplicatedAfterThreshold</code> notifica você quando um objeto qualificado para replicação que usa o S3 Replication Time Control replica após o limite de 15 minutos. O tipo de evento <code>s3:Replication:OperationNotTracked</code> notifica você quando um objeto estava qualificado para replicação usando o S3 Replication Time Control, mas não é mais rastreado por métricas de replicação.</p>
<code>s3:LifecycleExpiration:*</code> <code>s3:LifecycleExpiration:Delete</code> <code>s3:LifecycleExpiration:DeleteMarkerCreated</code>	<p>Ao usar os tipos de evento <code>LifecycleExpiration</code>, você pode receber uma notificação quando o Amazon S3 exclui um objeto com base na configuração S3 Lifecycle (Ciclo de vida do S3).</p> <p>O tipo de evento <code>s3:LifecycleExpiration:Delete</code> notifica você quando um objeto em um bucket sem versionamento é excluído. Ele também notifica quando uma versão de objeto é excluída permanentemente por uma configuração do S3 Lifecycle. O tipo de evento <code>s3:LifecycleExpiration:DeleteMarkerCreated</code> notifica você quando o S3 Lifecycle cria um marcador de exclusão quando uma versão atual de um objeto no bucket com versionamento é excluída.</p>
<code>s3:LifecycleTransition</code>	Você recebe esse evento de notificação quando um objeto é transferido para outra classe de armazenamento do Amazon S3 por uma configuração do S3 Lifecycle.

Tipos de evento	Descrição
s3:IntelligentTiering	Você recebe esse evento de notificação quando um objeto dentro da classe de armazenamento S3 Intelligent-Tiering for movido para o nível de acesso de arquivo ou para o nível de acesso de arquivo profundo.
s3:ObjectTagging: s3:ObjectTagging:Put s3:ObjectTagging:Delete	Usando os tipos de evento ObjectTagging, você pode habilitar a notificação quando uma etiqueta de objeto é adicionada ou excluída de um objeto. O tipo de evento s3 : ObjectTagging : Put notifica você quando ocorre PUT de uma etiqueta em um objeto ou quando uma etiqueta existente é atualizada. O tipo de evento s3 : ObjectTagging : Delete notifica você quando uma etiqueta é removida de um objeto.
s3:ObjectAcl:Put	Você recebe esse evento de notificação quando ocorre PUT de uma ACL em um objeto ou quando uma ACL existente é alterada. Não é gerado evento quando uma solicitação não resulta em alteração na ACL de um objeto.

Tipos de evento compatíveis com o Amazon EventBridge

Para obter uma lista de tipos de eventos que o Amazon S3 enviará ao Amazon EventBridge, consulte [Uso do EventBridge \(p. 1108\)](#)

Usar o Amazon SQS, o Amazon SNS e o Lambda

A habilitação de notificações é uma operação no nível do bucket. Você armazena informações de configuração de notificação no sub-recurso de notificação que está associado a um bucket. Depois que você cria ou altera a configuração de notificação de bucket, normalmente é necessário aguardar cinco minutos para que as alterações entrem em vigor. Ocorre um s3 : TestEvent quando a notificação é habilitada pela primeira vez. Você pode usar qualquer um dos métodos a seguir para gerenciar a configuração de notificação:

- Usar o console do Amazon S3: você pode usar a interface de usuário do console para definir uma configuração de notificação em um bucket sem ter que escrever nenhum código. Para obter mais informações, consulte [Habilitar e configurar notificações de eventos usando o console do Amazon S3 \(p. 1092\)](#).
- Uso dos AWS SDKs de forma programática: internamente, tanto o console como os SDKs chamam a API REST do Amazon S3 para gerenciar sub-recursos de notificação que estão associados ao bucket. Para obter exemplos de configurações de notificação usando exemplos do AWS SDK, consulte [Demonstração: configurar um bucket para notificações \(tópico do SNS ou fila do SQS\) \(p. 1095\)](#).

Note

Você também pode fazer chamadas de API REST do Amazon S3 diretamente do seu código. Porém, isso pode ser complicado, porque é necessário gravar o código para autenticar suas solicitações.

Independentemente do método usado, o Amazon S3 armazena a configuração de notificação como XML no sub-recurso notification (notificação) que está associado a um bucket. Para obter informações sobre sub-recursos de bucket, consulte [Opções de configuração do bucket \(p. 125\)](#).

Tópicos

- [Conceder permissões para publicar mensagens de notificação de vento a um destino \(p. 1090\)](#)
- [Habilitar e configurar notificações de eventos usando o console do Amazon S3 \(p. 1092\)](#)
- [Configuração de notificações de eventos de forma programática \(p. 1094\)](#)
- [Demonstração: configurar um bucket para notificações \(tópico do SNS ou fila do SQS\) \(p. 1095\)](#)
- [Configurar notificações de eventos usando filtragem de nomes de chave de objeto \(p. 1101\)](#)
- [Estrutura de mensagens de evento \(p. 1105\)](#)

Conceder permissões para publicar mensagens de notificação de vento a um destino

É necessário garantir à entidade principal do Amazon S3 as permissões necessárias para chamar a API relevante para publicar mensagens em um tópico do SNS, uma fila do SQS ou uma função Lambda. Isso serve para que o Amazon S3 possa publicar mensagens de notificação de eventos em um destino.

Tópicos

- [Conceder permissões para invocar uma função do AWS Lambda \(p. 1090\)](#)
- [Conceder permissões para publicar mensagens em um tópico do SNS ou em uma fila do SQS \(p. 1090\)](#)

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 eventos em um bucket do Amazon S3 para uma função do Lambda, o console configurará as permissões necessárias na função Lambda. Isso ocorre para que o Amazon S3 tenha permissões para invocar a função do bucket. Para obter mais informações, consulte [Habilitar e configurar notificações de eventos usando o console do Amazon S3 \(p. 1092\)](#).

Você também pode conceder permissões do Amazon S3 do AWS Lambda para invocar a função do Lambda. Para obter mais informações, consulte [Tutorial: uso do AWS Lambda com o Amazon S3](#) no Guia do desenvolvedor do AWS Lambda.

Conceder permissões para publicar mensagens em um tópico do SNS ou em uma fila do SQS

Para conceder ao Amazon S3 permissões para publicar mensagens em um tópico do SNS ou uma fila do SQS, anexe uma política do AWS Identity and Access Management (IAM) ao tópico do SNS ou à fila do SQS de destino.

Para obter um exemplo de como anexar uma política a um tópico do SNS ou a uma fila do SQS, consulte [Demonstração: configurar um bucket para notificações \(tópico do SNS ou fila do SQS\) \(p. 1095\)](#). Para obter mais informações sobre permissões, consulte os tópicos a seguir:

- [Exemplos de casos de controle de acesso do Amazon SNS](#) no Guia do desenvolvedor do Amazon Simple Notification Service
- [Gerenciamento de identidade e acesso no Amazon SQS](#) no Guia do desenvolvedor do Amazon Simple Queue Service

Política do IAM para um tópico do SNS de destino

Veja a seguir um exemplo de uma política do AWS Identity and Access Management (IAM) anexada ao tópico do SNS de destino. Para obter mais instruções sobre como usar essa política para configurar um

tópico do Amazon SNS de destino para notificações de eventos, consulte [Demonstração: configurar um bucket para notificações \(tópico do SNS ou fila do SQS\) \(p. 1095\)](#).

```
{  
    "Version": "2012-10-17",  
    "Id": "example-ID",  
    "Statement": [  
        {  
            "Sid": "Example SNS topic policy",  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "s3.amazonaws.com"  
            },  
            "Action": [  
                "SNS:Publish"  
            ],  
            "Resource": "SNS-topic-ARN",  
            "Condition": {  
                "ArnLike": {  
                    "aws:SourceArn": "arn:aws:s3:::bucket-name"  
                },  
                "StringEquals": {  
                    "aws:SourceAccount": "bucket-owner-account-id"  
                }  
            }  
        }  
    ]  
}
```

Política do IAM para uma fila do SQS de destino

Veja a seguir um exemplo de uma política do IAM anexada à fila do SQS de destino. Para obter mais instruções sobre como usar essa política para configurar uma fila do Amazon SQS de destino para notificações de eventos, consulte [Demonstração: configurar um bucket para notificações \(tópico do SNS ou fila do SQS\) \(p. 1095\)](#).

Para usar essa política, você deve atualizar o ARN da fila do Amazon SQS, o nome do bucket e o ID da Conta da AWS do proprietário do bucket.

```
{  
    "Version": "2012-10-17",  
    "Id": "example-ID",  
    "Statement": [  
        {  
            "Sid": "example-statement-ID",  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "s3.amazonaws.com"  
            },  
            "Action": [  
                "SQS:SendMessage"  
            ],  
            "Resource": "arn:aws:sqs:Region:account-id:queue-name",  
            "Condition": {  
                "ArnLike": {  
                    "aws:SourceArn": "arn:aws:s3:::awsexamplebucket1"  
                },  
                "StringEquals": {  
                    "aws:SourceAccount": "bucket-owner-account-id"  
                }  
            }  
        }  
    ]  
}
```

}

Tanto 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" }  
}
```

Política de chaves do AWS KMS

Se a fila do SQS ou os tópicos do SNS estiverem criptografados com uma chave gerenciada pelo cliente do AWS Key Management Service (AWS KMS), será necessário conceder permissão para que a entidade de serviço do Amazon S3 trabalhe com a fila ou os tópicos criptografados. Para conceder a permissão à entidade de serviço do Amazon S3, adicione a instrução a seguir à política de chave para a chave gerenciada pelo cliente.

```
{  
    "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": "*"  
        }  
    ]  
}
```

Para obter mais informações sobre políticas de chave do AWS KMS, consulte [Uso de políticas de chave no AWS KMS](#) no Guia do desenvolvedor do AWS Key Management Service.

Para obter mais informações sobre como usar a criptografia no lado do servidor com o AWS KMS para o Amazon SQS e o Amazon SNS, consulte:

- [Gerenciamento de chaves](#) no Guia do desenvolvedor do Amazon Simple Notification Service.
- [Gerenciamento de chaves](#) no Guia do desenvolvedor do Amazon Simple Queue Service.
- [Criptografia de mensagens publicadas no Amazon SNS com o AWS KMS](#) no Blog de computação da AWS.

Habilitar e configurar notificações de eventos usando o console do Amazon S3

Você pode habilitar certos eventos de bucket do Amazon S3 para enviar uma mensagem de notificação para um destino sempre que ocorrer um desses eventos. Esta seção explica como usar o console do Amazon S3 para habilitar notificações de evento. Para obter informações sobre como usar notificações de eventos com os AWS SDKs e as APIs REST do Amazon S3, consulte [Configuração de notificações de eventos de forma programática \(p. 1094\)](#).

Pré-requisitos: antes de habilitar as notificações de eventos para seu bucket, você deve configurar um dos tipos de destino e, em seguida, configurar permissões. Para obter mais informações, consulte [Destinos de](#)

eventos suportados (p. 1085) e [Conceder permissões para publicar mensagens de notificação de evento a um destino \(p. 1090\)](#).

Tópicos

- [Habilitar notificações do Amazon SNS, Amazon SQS ou Lambda usando o console do Amazon S3 \(p. 1093\)](#)

Habilitar notificações do Amazon SNS, Amazon SQS ou Lambda usando o console do Amazon S3

Para habilitar e configurar notificações de evento para um bucket do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket para o qual você deseja habilitar eventos.
3. Escolha Properties (Propriedades).
4. Navegue até a seção Event Notifications (Notificações de eventos) e escolha Create event notification (Criar notificação de evento).
5. Na seção General configuration (Configuração geral), especifique o nome do evento descritivo para sua notificação de evento. Opcionalmente, você também pode especificar um prefixo e um sufixo para limitar as notificações a objetos com chaves terminando nos caracteres especificados.
 - a. Insira uma descrição para o Event name (Nome do evento).

Se você não inserir um nome, um identificador exclusivo global (GUID) será gerado e usado para o nome.

- b. (Opcional) Para filtrar notificações de eventos por prefixo, insira um Prefix (Prefixo).

Por exemplo, é possível configurar um filtro de prefixo para receber notificações somente quando os arquivos são adicionados a uma pasta específica (por exemplo, `images/`).

- c. (Opcional) Para filtrar notificações de eventos por sufixo, insira um Suffix (Sufixo).

Para obter mais informações, consulte [Configurar notificações de eventos usando filtragem de nomes de chave de objeto \(p. 1101\)](#).

6. Na seção Event types (Tipos de evento), selecione um ou mais tipos de eventos para os quais você deseja receber notificações.

Para obter uma lista dos diferentes tipos de evento, consulte [Tipos de evento compatíveis com SQS, SNS e Lambda \(p. 1087\)](#).

7. Na seção Destination (Destino), escolha o destino da notificação de evento.

Note

Antes de publicar notificações de eventos, você deve conceder à entidade principal do Amazon S3 as permissões necessárias para chamar a API relevante. Desse modo, ela pode publicar notificações em uma função Lambda, tópico do SNS ou fila SQS.

- a. Selecione o tipo de destino: Lambda Function (Função do Lambda), SNS Topic (Tópico do SNS) ou SQS Queue (Fila do SQS).
- b. Depois de escolher o tipo de destino, escolha uma função, um tópico ou uma fila na lista.
- c. Como alternativa, se você preferir especificar um nome de recurso da Amazon (ARN), selecione Enter ARN (Inserir ARN) e insira o ARN.

Para obter mais informações, consulte [Destinos de eventos suportados \(p. 1085\)](#).

8. Escolha Save changes (Salvar alterações) e o Amazon S3 enviará uma mensagem de teste para o destino de notificação de evento.

Configuração de notificações de eventos de forma programática

Por padrão, as notificações não estão habilitadas para nenhum tipo de evento. Portanto, o sub-recurso notification (notificação) armazena inicialmente 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.

Para publicar mensagens de eventos em uma fila do SQS

Para definir uma fila do SQS como o destino de notificação para um ou mais tipos de evento, adicione o QueueConfiguration.

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>optional-id-string</Id>
    <Queue>sqs-queue-arn</Queue>
    <Event>event-type</Event>
    <Event>event-type</Event>
    ...
  </QueueConfiguration>
  ...
</NotificationConfiguration>
```

Para publicar mensagens de eventos em um tópico do SNS

Para definir um tópico do SNS como o destino de notificação para tipos de eventos específicos, adicione o TopicConfiguration.

```
<NotificationConfiguration>
  <TopicConfiguration>
    <Id>optional-id-string</Id>
    <Topic>sns-topic-arn</Topic>
    <Event>event-type</Event>
    <Event>event-type</Event>
    ...
  </TopicConfiguration>
  ...
</NotificationConfiguration>
```

Para invocar a função do AWS Lambda e fornecer uma mensagem de evento como um argumento

Para definir uma função do Lambda como o destino de notificação para tipos de evento específicos, adicione o CloudFunctionConfiguration.

```
<NotificationConfiguration>
  <CloudFunctionConfiguration>
    <Id>optional-id-string</Id>
    <CloudFunction>cloud-function-arn</CloudFunction>
    <Event>event-type</Event>
    <Event>event-type</Event>
    ...
  </CloudFunctionConfiguration>
</NotificationConfiguration>
```

```
</CloudFunctionConfiguration>
...
</NotificationConfiguration>
```

Para remover todas as notificações configuradas em um bucket

Para remover todas as notificações configuradas em um bucket, salve 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 \(p. 1105\)](#).

Para obter mais informações sobre como configurar notificações de eventos, consulte os seguintes tópicos:

- [Demonstração: configurar um bucket para notificações \(tópico do SNS ou fila do SQS\) \(p. 1095\)](#).
- [Configurar notificações de eventos usando filtragem de nomes de chave de objeto \(p. 1101\)](#)

Demonstração: configurar um bucket para notificações (tópico do SNS ou fila do SQS)

Você pode receber notificações do Amazon S3 usando o Amazon Simple Notification Service (Amazon SNS) ou o Amazon Simple Queue Service (Amazon SQS). Nesta demonstração, você adiciona uma configuração de notificação ao seu bucket usando um tópico do Amazon SNS e uma fila do Amazon SQS.

Tópicos

- [Resumo da demonstração \(p. 1095\)](#)
- [Etapa 1: Criar uma fila do Amazon SQS \(p. 1096\)](#)
- [Etapa 2: Criar um tópico do Amazon SNS \(p. 1097\)](#)
- [Etapa 3: Adicionar a configuração de notificação para o bucket \(p. 1098\)](#)
- [Etapa 4: Testar a configuração \(p. 1100\)](#)

Resumo da demonstração

Esta demonstração ajuda você a fazer o seguinte:

- 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 [Usar o Amazon SQS, o Amazon SNS e o Lambda \(p. 1089\)](#).

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 para ajudar a adicionar configurações de notificação de maneira programática.

Este procedimento inclui as seguintes etapas:

1. Crie uma fila do Amazon SQS.

Usando o console do Amazon SQS, crie uma fila do SQS. É possível acessar todas as mensagens que o Amazon S3 envia à fila de forma programática. Porém, para esta demonstração, verifique as mensagens de notificação no console.

Anexe uma política de acesso à fila para conceder ao Amazon S3 permissão para postar mensagens.

2. Crie um tópico do Amazon SNS.

Usando o console do Amazon SNS, crie um tópico do SNS e assine o tópico. Assim, todos os eventos postados nele são entregues a você. Especifique o e-mail como o protocolo de comunicações. Depois de criar um tópico, o Amazon SNS envia um e-mail. Use o link no e-mail para confirmar a assinatura do tópico.

Anexe 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 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 que está anexada à fila pela política a seguir.
 - a. No console do Amazon SQS, na lista Queues (Filas), escolha o nome da fila.
 - b. Na guia Access policy (Política de acesso), escolha Edit (Editar).
 - c. Substitua a política de acesso que está anexada à fila. Nele, forneça o ARN do Amazon SQS, o nome do bucket de origem e o ID da conta do proprietário do bucket.

```
{  
    "Version": "2012-10-17",  
    "Id": "example-ID",  
    "Statement": [  
        {  
            "Sid": "example-statement-ID",  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "s3.amazonaws.com"  
            },  
            "Action": [  
                "SQS:SendMessage"  
            ],  
            "Resource": "SQS-queue-ARN",  
            "Condition": {  
                "ArnLike": {  
                    "aws:SourceArn": "arn:aws:s3:::awsexamplebucket1"  
                },  
                "StringEquals": {  
                    "aws:SourceAccount": "bucket-owner-account-id"  
                }  
            }  
        }  
    ]  
}
```

- d. Escolha Save (Salvar).
3. (Opcional) Se a fila do Amazon SQS ou o tópico do Amazon SNS for habilitado para criptografia no lado do servidor com o AWS Key Management Service (AWS KMS), adicione a política a seguir à chave de criptografia simétrica correspondente gerenciada pelo cliente.

Adicione a política a uma chave gerenciada pelo cliente porque você não pode modificar a chave gerenciada pela AWS para o Amazon SQS ou o Amazon SNS.

```
{
```

```
"Version": "2012-10-17",
"Id": "example-ID",
"Statement": [
    {
        "Sid": "example-statement-ID",
        "Effect": "Allow",
        "Principal": {
            "Service": "s3.amazonaws.com"
        },
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": "*"
    }
]
```

Para obter mais informações sobre como usar o SSE para o Amazon SQS e o Amazon SNS com o AWS KMS, consulte:

- [Gerenciamento de chaves](#) no Guia do desenvolvedor do Amazon Simple Notification Service.
 - [Gerenciamento de chaves](#) no Guia do desenvolvedor do Amazon Simple Queue Service.
4. Anotar o ARN da fila.

A fila do SQS que você criou é outro recurso em sua Conta da AWS. Ele tem um nome do recurso da Amazon (ARN) exclusivo. Esse ARN será necessário na próxima etapa. O ARN tem este formato:

```
arn:aws:sqs:aws-region:account-id:queue-name
```

Etapa 2: Criar um tópico do Amazon SNS

Siga as etapas para criar e assinar um tópico do Amazon SNS.

1. Usando o console do Amazon SNS, crie um tópico. Para obter instruções, consulte [Criação de um tópico do Amazon SNS](#) 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 [Assinatura de um tópico do Amazon SNS](#) no Guia do desenvolvedor do Amazon Simple Notification Service.

Você receberá um e-mail solicitando a confirmação da assinatura do tópico. Confirme a assinatura.

3. Substitua a política de acesso anexada ao tópico pela seguinte política. Nele, forneça o ARN do tópico SNS, o nome do bucket de origem e o ID da conta do proprietário do bucket.

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "Example SNS topic policy",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SNS:Publish"
            ],
            "Resource": "SNS-topic-ARN",
            "Condition": {

```

```
        "ArnLike": {
            "aws:SourceArn": "arn:aws:s3:::*:bucket-name"
        },
        "StringEquals": {
            "aws:SourceAccount": "bucket-owner-account-id"
        }
    }
}
```

4. Anote o ARN do tópico.

O tópico do SNS que você criou é outro recurso em sua Conta da AWS e tem um 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 3: Adicionar a configuração de notificação para o bucket

Você pode habilitar notificações de bucket usando o console do Amazon S3 ou de forma 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.

Opção A: habilitar notificações em um bucket usando o console

Usando o console do Amazon S3, adicione uma configuração de notificação solicitando que o Amazon S3 faça o seguinte:

- Publique eventos do tipo Todos os eventos criados por objetos na fila do Amazon SQS.
- Publique eventos do tipo Objeto em RRS perdido no tópico do Amazon SNS.

Depois de salvar a configuração de notificação, o Amazon S3 posta uma mensagem de teste, que você recebe via e-mail.

Para obter instruções, consulte [Habilitar e configurar notificações de eventos usando o console do Amazon S3 \(p. 1092\)](#).

Opção B: habilitar notificações em um bucket usando os AWS SDKs

.NET

O exemplo de código C# a seguir fornece uma lista completa de códigos que adicionam uma configuração de notificação a um bucket. Você deve atualizar o código e fornecer o nome do bucket e o ARN do tópico do SNS. Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class EnableNotificationsTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string snsTopic = "*** SNS topic ARN ***";
```

```
private const string sqsQueue = "*** SQS topic ARN ***";
// Specify your bucket region (an example region is shown).
private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
private static IAmazonS3 client;

public static void Main()
{
    client = new AmazonS3Client(bucketRegion);
    EnableNotificationAsync().Wait();
}

static async Task EnableNotificationAsync()
{
    try
    {
        PutBucketNotificationRequest request = new PutBucketNotificationRequest
        {
            BucketName = bucketName
        };

        TopicConfiguration c = new TopicConfiguration
        {
            Events = new List<EventType> { EventType.ObjectCreatedCopy },
            Topic = snsTopic
        };
        request.TopicConfigurations = new List<TopicConfiguration>();
        request.TopicConfigurations.Add(c);
        request.QueueConfigurations = new List<QueueConfiguration>();
        request.QueueConfigurations.Add(new QueueConfiguration()
        {
            Events = new List<EventType> { EventType.ObjectCreatedPut },
            Queue = sqsQueue
        });

        PutBucketNotificationResponse response = await
client.PutBucketNotificationAsync(request);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:'{0}' ",
e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown error encountered on server. Message:'{0}' ",
", e.Message);
    }
}
}
```

Java

O seguinte exemplo mostra como adicionar uma configuração de notificação a um bucket. Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

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

```
import com.amazonaws.services.s3.model.*;
import java.io.IOException;
import java.util.EnumSet;

public class EnableNotificationOnABucket {

    public static void main(String[] args) throws IOException {
        String bucketName = "*** Bucket name ***";
        Regions clientRegion = Regions.DEFAULT_REGION;
        String snsTopicARN = "*** SNS Topic ARN ***";
        String sqsQueueARN = "*** SQS Queue ARN ***";

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

            // Add an SNS topic notification.
            notificationConfiguration.addConfiguration("snsTopicConfig",
                new TopicConfiguration(snsTopicARN,
EnumSet.of(S3Event.ObjectCreated)));

            // Add an SQS queue notification.
            notificationConfiguration.addConfiguration("sqsQueueConfig",
                new QueueConfiguration(sqsQueueARN,
EnumSet.of(S3Event.ObjectCreated)));

            // Create the notification configuration request and set the bucket
notification configuration.
            SetBucketNotificationConfigurationRequest request = new
SetBucketNotificationConfigurationRequest(
                bucketName, notificationConfiguration);
            s3Client.setBucketNotificationConfiguration(request);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

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](#) na seção “Conceitos básicos” do Guia do desenvolvedor do Amazon Simple Queue Service.

Configurar notificações de eventos usando filtragem de nomes de chave de objeto

Ao configurar uma notificação de evento do Amazon S3, você deve especificar quais tipos de evento suportados do Amazon S3 fazem com que o Amazon S3 envie a notificação. Se um tipo de evento que você não especificou ocorrer no bucket do S3, o Amazon S3 não enviará a notificação.

Você pode configurar as notificações para serem filtradas pelo prefixo e pelo sufixo do nome da chave de objetos. Por exemplo, é possível definir uma configuração para que uma notificação seja enviada para você apenas quando arquivos de imagem com uma extensão de nome de arquivo ".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 fazer com que notificações para objetos com um prefixo "logs/" no mesmo bucket sejam entregues a uma função do AWS Lambda.

É possível definir configurações de notificação que usem filtro de nomes de chaves de objetos no console do Amazon S3. Você pode fazer isso usando as APIs do Amazon S3 por meio dos AWS SDKs 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 [Habilitar e configurar notificações de eventos usando o console do Amazon S3 \(p. 1092\)](#).

O Amazon S3 armazena a configuração de notificação como XML no sub-recurso notificação associado a um bucket, conforme descrito em [Usar o Amazon SQS, o Amazon SNS e o Lambda \(p. 1089\)](#). Use a estrutura XML do `Filter` para definir regras para que as notificações sejam filtradas pelo prefixo ou pelo sufixo do nome da chave de um objeto. Para obter informações sobre a estrutura XML `Filter`, consulte [Notificação de PUT Bucket](#) na Referência da API do Amazon Simple Storage Service.

Se uma notificação de evento do Amazon S3 estiver configurada para usar a filtragem de nomes de chave de objeto, as notificações serão publicadas somente para objetos com um determinado prefixo ou sufixo de nome de chave. Um caractere selvagem ("*") não pode ser usado em filtros como um prefixo ou sufixo para representar qualquer caractere. Se você usar quaisquer caracteres especiais no valor do prefixo ou sufixo, deverá inseri-los no [formato codificado em URL \(codificado em porcentagem\)](#). Para obter mais informações, consulte [Diretrizes de nomeação de chave de objeto \(p. 163\)](#).

Configurações de notificação que usam o `Filter` não podem definir regras de filtragem com prefixes sobrepostos, sufixos sobrepostos ou prefixos e sufixos sobrepostos. As seções a seguir têm exemplos de configurações de notificação válidas com filtragem de nome de chave de objeto. Elas também contêm exemplos de configurações de notificação que não são válidas devido à sobreposição de prefixo e sufixo.

Tópicos

- [Exemplos de configurações válidas de notificação com filtragem de nome de chave de objeto \(p. 1101\)](#)
- [Exemplos de configurações de notificação com sobreposição inválida de prefixo e sufixo \(p. 1104\)](#)

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 são publicados sempre que um objeto que tenha um prefixo `images/` e um sufixo `.jpg` é `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/` vão para a fila A, enquanto as notificações para solicitações PUT na pasta `logs/` vão para a fila 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 por cloud-function-A do Lambda e que todas as imagens `.png` recém-adicionadas sejam processadas por cloud-function-B. Os sufixos `.png` e `.jpg` não se sobrepõem, apesar de terminarem com a mesma letra. Dois sufixos são considerados sobrepostos se determinada string puder terminar com ambos os sufixos. Uma string não pode terminar com `.png` e com `.jpg`, então 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>
```

```
</S3Key>
</Filter>
<CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-A</CloudFunction>
<Event>s3:ObjectCreated:Put</Event>
</CloudFunctionConfiguration>
<CloudFunctionConfiguration>
<Id>2</Id>
<Filter>
<S3Key>
<FilterRule>
<Name>suffix</Name>
<Value>.png</Value>
</FilterRule>
</S3Key>
</Filter>
<CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-B</CloudFunction>
<Event>s3:ObjectCreated:Put</Event>
</CloudFunctionConfiguration>
</NotificationConfiguration>
```

As configurações de notificação que usam o `Filter` não podem definir regras de filtro com prefixos sobrepostos para os mesmos tipos de evento. Eles só podem fazer isso se os prefixos sobrepostos forem usados com sufixos que não são sobrepostos. A configuração de exemplo a seguir mostra como objetos criados com um prefixo comum, mas com sufixos 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>
<CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-A</CloudFunction>
<Event>s3:ObjectCreated:Put</Event>
</CloudFunctionConfiguration>
<CloudFunctionConfiguration>
<Id>2</Id>
<Filter>
<S3Key>
<FilterRule>
<Name>prefix</Name>
<Value>images</Value>
</FilterRule>
<FilterRule>
<Name>suffix</Name>
<Value>.png</Value>
</FilterRule>
</S3Key>
</Filter>
<CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-B</CloudFunction>
<Event>s3:ObjectCreated:Put</Event>
</CloudFunctionConfiguration>
</NotificationConfiguration>
```

Exemplos de configurações de notificação com sobreposição inválida de prefixo e sufixo

Na maioria das vezes, as configurações de notificação que usam `Filter` não podem definir regras de filtro com prefixos sobrepostos, sufixos sobrepostos ou combinações sobrepostas de prefixos e sufixos para os mesmos tipos de evento. Você pode ter prefixos sobrepostos, desde que os sufixos não se sobreponham. Para ver um exemplo, consulte [Configurar notificações de eventos usando filtragem de nomes de chave de objeto \(p. 1101\)](#).

Você pode usar filtros de nomes de chave de objetos sobrepostos com diferentes tipos de evento. Por exemplo, você poderia criar uma configuração de notificação que usasse o prefixo `image/` para o tipo de evento `ObjectCreated:Put` e o prefixo `image/` para o tipo de evento `ObjectRemoved:*`.

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 ou a API do Amazon S3. Esta seção mostra exemplos de configurações de notificação que não são válidas 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 não é válida porque tem prefixos sobrepostos. Especificamente, o prefixo raiz sobrepõe qualquer outro prefixo. O mesmo se aplica se você usar 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 não é válida porque tem sufixos sobrepostos. Dois sufixos são considerados sobrepostos se determinada string puder terminar com ambos os sufixos. Uma string pode terminar com `.jpg` e `.pg`. Portanto, os sufixos se sobrepõem. O mesmo vale para prefixos. Dois prefixos são considerados sobrepostos se determinada string puder começar com ambos os 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>
```

A configuração de notificação a seguir não é válida 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>
```

Estrutura de mensagens de evento

A mensagem de notificação que o Amazon S3 envia para publicar um evento tem o formato JSON.

Para obter uma visão geral e instruções sobre como configurar notificações de eventos, consulte [Notificações de eventos do Amazon S3 \(p. 1083\)](#).

Este exemplo mostra a versão 2.2 da estrutura JSON da notificação do evento. O Amazon S3 usa as versões 2.1, 2.2 e 2.3 desta estrutura de evento. O Amazon S3 usa a versão 2.2 para notificações de eventos de replicação entre regiões. Utiliza a versão 2.3 para o S3 Lifecycle, S3 Intelligent-Tiering, ACL de objeto, marcação de objetos e eventos de exclusão de restauração de objetos. Essas versões contêm informações adicionais específicas para essas operações. As versões 2.2 e 2.3 são compatíveis com a versão 2.1, que o Amazon S3 usa atualmente para outros tipos de notificação de eventos.

```
{
```

```

"Records": [
    {
        "eventVersion": "2.2",
        "eventSource": "aws:s3",
        "awsRegion": "us-west-2",
        "eventTime": "The time, in ISO-8601 format, for example, 1970-01-01T00:00:00.000Z, when Amazon S3 finished processing the request",
        "eventName": "event-type",
        "userIdentity": {
            "principalId": "Amazon-customer-ID-of-the-user-who-caused-the-event"
        },
        "requestParameters": {
            "sourceIPAddress": "ip-address-where-request-came-from"
        },
        "responseElements": {
            "x-amz-request-id": "Amazon S3 generated request ID",
            "x-amz-id-2": "Amazon S3 host that processed the request"
        },
        "s3": {
            "s3SchemaVersion": "1.0",
            "configurationId": "ID found in the bucket notification configuration",
            "bucket": {
                "name": "bucket-name",
                "ownerIdentity": {
                    "principalId": "Amazon-customer-ID-of-the-bucket-owner"
                },
                "arn": "bucket-ARN"
            },
            "object": {
                "key": "object-key",
                "size": "object-size in bytes",
                "eTag": "object eTag",
                "versionId": "object version if bucket is versioning-enabled, otherwise null",
                "sequencer": "a string representation of a hexadecimal value used to determine event sequence, only used with PUTS and DELETES"
            }
        },
        "glacierEventData": {
            "restoreEventData": {
                "lifecycleRestorationExpiryTime": "The time, in ISO-8601 format, for example, 1970-01-01T00:00:00.000Z, of Restore Expiry",
                "lifecycleRestoreStorageClass": "Source storage class for restore"
            }
        }
    }
]
}

```

Sobre a estrutura de mensagens de evento, observe:

- O valor da chave `eventVersion` contém uma versão principal e secundária no formulário `<major>.<minor>`.

A versão principal será incrementada se o Amazon S3 fizer uma alteração na estrutura do evento que não seja compatível com versões anteriores. Isso inclui remover um campo JSON que já está presente ou alterar como os conteúdos de um campo são representados (por exemplo, um formato de data).

A versão secundária será incrementada se o Amazon S3 adicionar novos campos à estrutura do evento. Isso pode ocorrer se forem fornecidas novas informações para alguns ou todos os eventos existentes. Isso também pode ocorrer se novas informações forem fornecidas apenas nos tipos de evento recém-apresentados. Os aplicativos devem ignorar novos campos para permanecerem compatíveis com novas versões secundárias da estrutura do evento.

Se novos tipos de evento forem apresentados, mas a estrutura do evento não for modificada, a versão do evento não será alterada.

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 por sua aplicação estejam presentes, também recomendamos fazer uma comparação "maior que ou igual a" na versão secundária.

- O `eventName` faz referência à lista de [tipos de notificação de evento](#), mas não contém o prefixo `s3:`.
- O valor da chave `responseElements` será útil se você quiser rastrear uma solicitação acompanhando o AWS Support. Tanto o `x-amz-request-id` quanto 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. Isso ocorre para que eles 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 "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 mesma ordem em que os eventos ocorreram. No entanto, as notificações de eventos que criam objetos (PUTs) e excluem objetos contêm um `sequencer`. Ele pode ser usado para determinar a ordem dos eventos de 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 `sequencer` maior será o evento que ocorreu depois. Se você estiver usando notificações de evento para manter um banco de dados ou um índice separado dos objetos do Amazon S3, recomendamos comparar e armazenar os valores de `sequencer` ao processar cada notificação de evento.

Observe o seguinte:

- Não é possível usar `sequencer` para determinar a ordem de eventos em chaves de objetos diferentes.
- Os sequenciadores podem ter comprimentos diferentes. Portanto, para comparar esses valores, primeiro preencha zeros à direita do menor valor e faça uma comparação lexicográfica.
- A chave `glacierEventData` só está visível para eventos `s3:ObjectRestore:Completed`.
- A chave `restoreEventData` contém atributos que estão relacionados à solicitação de restauração.
- A chave `replicationEventData` só é visível para eventos de replicação.
- A chave `intelligentTieringEventData` só fica visível para eventos do S3 Intelligent-Tiering.
- A chave `lifecycleEventData` só fica visível para eventos de transição do S3 Lifecycle.

Exemplos de mensagens

Veja a seguir exemplos de mensagens de notificação de eventos do Amazon S3.

Mensagem de teste do Amazon S3

Depois que você configura uma notificação de evento em um bucket, o Amazon S3 envia a mensagem de teste a seguir.

```
{  
    "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 uma solicitação PUT

A mensagem a seguir é um exemplo de mensagem que o Amazon S3 envia para publicar um evento s3:ObjectCreated:Put.

```
{  
    "Records": [  
        {  
            "eventVersion": "2.1",  
            "eventSource": "aws:s3:",  
            "awsRegion": "us-west-2",  
            "eventTime": "1970-01-01T00:00:00.000Z",  
            "eventName": "ObjectCreated:Put",  
            "userIdentity": {  
                "principalId": "AIDAJDPLRKLG7UEXAMPLE"  
            },  
            "requestParameters": {  
                "sourceIPAddress": "127.0.0.1"  
            },  
            "responseElements": {  
                "x-amz-request-id": "C3D13FE58DE4C810",  
                "x-amz-id-2": "FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/JRWeUWerMUE5JgHvANOjpD"  
            },  
            "s3": {  
                "s3SchemaVersion": "1.0",  
                "configurationId": "testConfigRule",  
                "bucket": {  
                    "name": "mybucket",  
                    "ownerIdentity": {  
                        "principalId": "A3NL1KOZZKExample"  
                    },  
                    "arn": "arn:aws:s3:::mybucket"  
                },  
                "object": {  
                    "key": "HappyFace.jpg",  
                    "size": 1024,  
                    "eTag": "d41d8cd98f00b204e9800998ecf8427e",  
                    "versionId": "096fKKXTRTtl3on89fVO.nfljtsv6qko",  
                    "sequencer": "0055AED6DCD90281E5"  
                }  
            }  
        }  
    ]  
}
```

Para obter uma definição de cada prefixo de identificação do IAM (por exemplo, AIDA, AROA, AGPA), consulte [Identificadores do IAM](#) no Guia do usuário do IAM.

Uso do EventBridge

O Amazon S3 pode enviar eventos ao Amazon EventBridge sempre que certos eventos ocorrerem no bucket. Ao contrário de outros destinos, não é necessário selecionar quais tipos de evento você deseja entregar. Depois que o EventBridge estiver habilitado, todos os eventos abaixo serão enviados ao EventBridge. Você pode usar regras do EventBridge para encaminhar eventos para outros destinos. A seguir, estão listados os eventos que o Amazon S3 envia ao EventBridge.

Tipo de evento	Descrição
Objeto criado	Um objeto foi criado.

Tipo de evento	Descrição
	O campo reason (motivo) na estrutura de mensagens de evento indica qual API do S3 foi usada para criar o objeto: PutObject , POST Object , CopyObject ou CompleteMultipartUpload .
Objeto excluído (DeleteObject) Objeto excluído (validade do ciclo de vida)	<p>Um objeto foi excluído.</p> <p>Quando um objeto é excluído usando uma chamada de API do S3, o campo reason (motivo) é definido como DeleteObject. Quando um objeto é excluído por uma regra de validade do S3 Lifecycle, o campo reason (motivo) é definido como Lifecycle Expiration (Validade do ciclo de vida). Para obter mais informações, consulte Expirando objetos (p. 753).</p> <p>Quando um objeto sem versionamento é excluído ou um objeto com versionamento é excluído permanentemente, o campo deletion-type (tipo de exclusão) é definido como Permanently Deleted (Excluído permanentemente). Quando um marcador de exclusão é criado para um objeto versionado, o campo do deletion-type (tipo de exclusão) é definido como Delete Marker Created (Marcador de exclusão criado). Para obter mais informações, consulte Excluir versões de objetos de um bucket com versionamento habilitado (p. 700).</p>
Restauração de objeto iniciada	Uma restauração de objetos foi iniciada da classe de armazenamento S3 Glacier ou S3 Glacier Deep Archive ou do nível de acesso de arquivo ou de acesso de arquivo profundo do S3 Intelligent-Tiering. Para obter mais informações, consulte Trabalhando com objetos arquivados (p. 712) .
Restauração de objeto concluída	Uma restauração de objeto foi concluída.
Restauração de objeto expirada	A cópia temporária de um objeto restaurado do S3 Glacier ou S3 Glacier Deep Archive expirou e foi excluída.
Classe de armazenamento de objetos alterada	Um objeto foi transferido para uma classe de armazenamento diferente. Para obter mais informações, consulte Transição de objetos usando o Amazon S3 Lifecycle (p. 748) .
Nível de acesso de objeto alterado	Um objeto foi transferido para o nível de acesso de arquivo ou de acesso de arquivo profundo do S3 Intelligent-Tiering. Para obter mais informações, consulte Amazon S3 Intelligent-Tiering (p. 737) .
ACL de objeto atualizada	Uma lista de controle de acesso (ACL) de objetos foi definida usando PutObjectACL. Não é gerado evento quando uma solicitação não resulta em alteração na ACL de um objeto. Para obter mais informações, consulte Visão geral da lista de controle de acesso (ACL) (p. 587) .
Etiquetas de objeto adicionadas	Um conjunto de etiquetas foi adicionado a um objeto usando PutObjectTagging. Para obter mais informações, consulte Categorizando seu armazenamento usando tags (p. 878) .
Etiquetas de objeto excluídas	Todas as etiquetas foram removidas de um objeto usando DeleteObjectTagging. Para obter mais informações, consulte Categorizando seu armazenamento usando tags (p. 878) .

Note

Para obter mais informações sobre como os tipos de eventos do Amazon S3 são mapeados para os tipos de evento do EventBridge, consulte [Mapeamento e solução de problemas do Amazon EventBridge \(p. 1114\)](#).

Você pode usar o Amazon S3 Event Notifications com o EventBridge para escrever regras que executam ações quando um evento ocorre em seu bucket. Por exemplo, ele pode enviar uma notificação para você. Para obter mais informações, consulte [O que é o EventBridge](#) no Guia do usuário do Amazon EventBridge.

Para obter informações sobre preços, consulte [Preços do Amazon EventBridge](#).

Tópicos

- [Permissões do Amazon EventBridge \(p. 1110\)](#)
- [Habilitar o Amazon EventBridge \(p. 1110\)](#)
- [Estrutura de mensagens de eventos do EventBridge \(p. 1111\)](#)
- [Mapeamento e solução de problemas do Amazon EventBridge \(p. 1114\)](#)

Permissões do Amazon EventBridge

O Amazon S3 não necessita de outras permissões para entregar eventos ao Amazon EventBridge.

Habilitar o Amazon EventBridge

Você pode habilitar o Amazon EventBridge usando o console do S3, a AWS Command Line Interface (AWS CLI) ou a API REST do Amazon S3.

Uso do console do S3

Para habilitar a entrega de eventos do EventBridge no console do S3.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket para o qual você deseja habilitar eventos.
3. Escolha Properties (Propriedades).
4. Navegue até a seção Event Notifications (Notificações de eventos) e encontre a subseção Amazon EventBridge. Escolha Edit (Editar).
5. Em Send notifications to Amazon EventBridge for all events in this bucket (Enviar notificações para o Amazon EventBridge para todos os eventos deste bucket), escolha No (Não).

Note

Depois que você habilitar o EventBridge, leva cerca de cinco minutos para que as alterações sejam implementadas.

Uso do AWS CLI

O exemplo a seguir cria uma configuração de notificação de bucket para bucket **DOC-EXAMPLE-BUCKET1** com o Amazon EventBridge habilitado.

```
aws s3api put-bucket-notification-configuration --bucket DOC-EXAMPLE-BUCKET1 --  
notification-configuration '{ "EventBridgeConfiguration": {} }'
```

Uso dos REST API

Você pode habilitar o Amazon EventBridge de maneira programática em um bucket chamando a API REST do Amazon S3. Para obter mais informações, consulte [PutBucketNotificationConfiguration](#) na Referência da API do Amazon Simple Storage Service.

O exemplo a seguir exibe o XML usado para criar uma configuração de notificação de bucket com o Amazon EventBridge habilitado.

```
<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <EventBridgeConfiguration>
    </EventBridgeConfiguration>
</NotificationConfiguration>
```

Criar regras do EventBridge

Depois de habilitado, você pode criar regras do Amazon EventBridge para determinadas tarefas. Por exemplo, é possível enviar notificações por e-mail quando um objeto é criado. Para obter um tutorial completo, consulte [Tutorial: enviar uma notificação quando um objeto do Amazon S3 for criado](#) no Guia do usuário do Amazon EventBridge.

Estrutura de mensagens de eventos do EventBridge

A mensagem de notificação que o Amazon S3 envia para publicar um evento tem o formato JSON. Quando o Amazon S3 envia um evento ao Amazon EventBridge, os campos a seguir estão presentes.

- **version** (versão): atualmente 0 (zero) para todos os eventos.
- **id**: um UUID versão 4 gerado para cada evento.
- **detail-type** (detalhe-tipo): o tipo de evento que está sendo enviado. Consulte [Uso do EventBridge \(p. 1108\)](#) para obter uma lista dos tipos de evento.
- **source** (origem): identifica o serviço que originou o evento.
- **account** (conta): o ID de 12 dígitos da Conta da AWS do proprietário do bucket.
- **time** (hora): a hora em que o evento ocorreu.
- **region** (região): identifica a Região da AWS do bucket.
- **resource** (recurso): uma matriz JSON que contém o nome do recurso da Amazon (ARN) do bucket.
- **detail** (detalhe): contém um objeto JSON com as informações sobre o evento. Para obter mais informações sobre o que pode ser incluído no campo, consulte [Campo de detalhes da mensagem de evento \(p. 1113\)](#).

Exemplos de estrutura de mensagens de evento

Veja a seguir exemplos de algumas das mensagens de notificação de eventos do Amazon S3 que podem ser enviadas ao Amazon EventBridge.

Objeto criado

```
{
  "version": "0",
  "id": "17793124-05d4-b198-2fde-7edede63b103",
  "detail-type": "Object Created",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
```

```
    "arn:aws:s3::::DOC-EXAMPLE-BUCKET1"
],
"detail": {
    "version": "0",
    "bucket": {
        "name": "DOC-EXAMPLE-BUCKET1"
    },
    "object": {
        "key": "example-key",
        "size": 5,
        "etag": "b1946ac92492d2347c6235b4d2611184",
        "version-id": "IYV3p45BT0ac8hjHg1houSdS1a.Mro8e",
        "sequencer": "617f08299329d189"
    },
    "request-id": "N4N7GDK58NMKJ12R",
    "requester": "123456789012",
    "source-ip-address": "1.2.3.4",
    "reason": "PutObject"
}
}
```

Objeto excluído (usando DeleteObject)

```
{
    "version": "0",
    "id": "2ee9cc15-d022-99ea-1fb8-1b1bac4850f9",
    "detail-type": "Object Deleted",
    "source": "aws.s3",
    "account": "111122223333",
    "time": "2021-11-12T00:00:00Z",
    "region": "ca-central-1",
    "resources": [
        "arn:aws:s3::::DOC-EXAMPLE-BUCKET1"
    ],
    "detail": {
        "version": "0",
        "bucket": {
            "name": "DOC-EXAMPLE-BUCKET1"
        },
        "object": {
            "key": "example-key",
            "etag": "d41d8cd98f00b204e9800998ecf8427e",
            "version-id": "1QW9g1Z99LUNbvaaYVpW9xDlOLU.qxgF",
            "sequencer": "617f0837b476e463"
        },
        "request-id": "0BH729840619AG5K",
        "requester": "123456789012",
        "source-ip-address": "1.2.3.4",
        "reason": "DeleteObject",
        "deletion-type": "Delete Marker Created"
    }
}
```

Objeto excluído (usando validade do ciclo de vida)

```
{
    "version": "0",
    "id": "ad1de317-e409-eba2-9552-30113f8d88e3",
    "detail-type": "Object Deleted",
    "source": "aws.s3",
    "account": "111122223333",
```

```
"time": "2021-11-12T00:00:00Z",
"region": "ca-central-1",
"resources": [
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
],
"detail": {
    "version": "0",
    "bucket": {
        "name": "DOC-EXAMPLE-BUCKET1"
    },
    "object": {
        "key": "example-key",
        "etag": "d41d8cd98f00b204e9800998ecf8427e",
        "version-id": "mtB0cV.jejK63XkRNceanNMC.qXPWLeK",
        "sequencer": "617b3980000000000"
    },
    "request-id": "20EB74C14654DC47",
    "requester": "s3.amazonaws.com",
    "reason": "Lifecycle Expiration",
    "deletion-type": "Delete Marker Created"
}
}
```

Restauração de objeto concluída

```
{
    "version": "0",
    "id": "6924de0d-13e2-6bbf-c0c1-b903b753565e",
    "detail-type": "Object Restore Completed",
    "source": "aws.s3",
    "account": "111122223333",
    "time": "2021-11-12T00:00:00Z",
    "region": "ca-central-1",
    "resources": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
    ],
    "detail": {
        "version": "0",
        "bucket": {
            "name": "DOC-EXAMPLE-BUCKET1"
        },
        "object": {
            "key": "example-key",
            "size": 5,
            "etag": "b1946ac92492d2347c6235b4d2611184",
            "version-id": "KKsjUC1.6gIjqtvhfg5AdMI0eCePIiT3"
        },
        "request-id": "189F19CB7FB1B6A4",
        "requester": "s3.amazonaws.com",
        "restore-expiry-time": "2021-11-13T00:00:00Z",
        "source-storage-class": "GLACIER"
    }
}
```

Campo de detalhes da mensagem de evento

O campo `detail` (detalhe) contém um objeto JSON com as informações sobre o evento. Os campos a seguir podem estar presentes no campo `detail` (detalhe).

- `version` (versão): atualmente 0 (zero) para todos os eventos.
- `bucket`: informações sobre o bucket do Amazon S3 envolvido no evento.

- object (objeto): informações sobre o objeto do Amazon S3 envolvido no evento.
- request-id (solicitação-id): ID da solicitação na resposta do S3.
- requester (solicitante): ID da Conta da AWS ou entidade principal do serviço da AWS do solicitante.
- source-ip-address (origem-ip-endereço): endereço IP de origem da solicitação do S3. Presente somente para eventos acionados por uma solicitação do S3.
- reason (motivo): para eventos Object Created (Objeto criado), a API do S3 usada para criar o objeto: [PutObject](#), [POST Object](#), [CopyObject](#) ou [CompleteMultipartUpload](#). Em eventos Object Deleted (Objeto excluído), é definido como DeleteObject quando um objeto é excluído por uma chamada de API do S3 ou Lifecycle Expiration (Validade do ciclo de vida) quando um objeto é excluído por uma regra de validade do S3 Lifecycle. Para obter mais informações, consulte [Expirando objetos \(p. 753\)](#).
- deletion-type (exclusão-tipo): em eventos Object Deleted (Objeto excluído) quando um objeto sem versionamento é excluído ou um objeto com versionamento é excluído permanentemente, o campo deletion-type é definido como Permanently Deleted (Excluído permanentemente). Quando um marcador de exclusão é criado para um objeto versionado, é definido como Delete Marker Created (Marcador de exclusão criado). Para obter mais informações, consulte [Excluir versões de objetos de um bucket com versionamento habilitado \(p. 700\)](#).
- restore-expiry-time (restaurar-validade-hora): para eventos Object Restore Completed (Restauração de objetos concluída), a hora em que a cópia temporária do objeto será excluída do S3. Para obter mais informações, consulte [Trabalhando com objetos arquivados \(p. 712\)](#).
- source-storage-class (origem-armazenamento-classe): para eventos Object Restore Initiated (Iniciar a restauração de objetos) e Object Restore Completed (Restauração de objetos concluída), a classe de armazenamento do objeto que está sendo restaurado. Para obter mais informações, consulte [Trabalhando com objetos arquivados \(p. 712\)](#).
- destination-storage-class (destino-armazenamento-classe): para eventos Object Storage Class Changed (Classe de armazenamento de objetos alterada), a nova classe de armazenamento do objeto. Para obter mais informações, consulte [Transição de objetos usando o Amazon S3 Lifecycle \(p. 748\)](#).
- destination-access-tier (destino-acesso-nível): para eventos Object Access Tier Changed (Nível de acesso a objetos alterado), o novo nível de acesso do objeto. Para obter mais informações, consulte [Amazon S3 Intelligent-Tiering \(p. 737\)](#).

Mapeamento e solução de problemas do Amazon EventBridge

A tabela a seguir descreve como os tipos de eventos do Amazon S3 são mapeados para os tipos de evento do Amazon EventBridge.

Tipo de evento do S3	Tipo de detalhes do Amazon EventBridge
ObjectCreated:Put	Objeto criado
ObjectCreated:Post	
ObjectCreated:Copy	
ObjectCreated:CompleteMultipartUpload	
ObjectRemoved:Delete	Objeto excluído
ObjectRemoved:DeleteMarkerCreated	
LifecycleExpiration:Delete	
LifecycleExpiration:DeleteMarkerCreated	
ObjectRestore:Post	Restauração de objeto iniciada

Tipo de evento do S3	Tipo de detalhes do Amazon EventBridge
ObjectRestore:Completed	Restauração de objeto concluída
ObjectRestore:Delete	Restauração de objeto expirada
LifecycleTransition	Classe de armazenamento de objetos alterada
IntelligentTiering	Nível de acesso de objeto alterado
ObjectTagging:Put	Etiquetas de objeto adicionadas
ObjectTagging:Delete	Etiquetas de objeto excluídas
ObjectAcl:Put	ACL de objeto atualizada

Solução de problemas do Amazon EventBridge

Para obter informações sobre como solucionar problemas do EventBridge, consulte [Solucionar problemas no Amazon EventBridge](#) no Guia do usuário do Amazon EventBridge.

Usar análises e insights

Você pode usar análises e insights no Amazon S3 para entender, analisar e otimizar o uso do armazenamento. Para obter mais informações, consulte os tópicos abaixo.

Tópicos

- [Análise do Amazon S3 – Análise de classe de armazenamento \(p. 1116\)](#)
- [Avaliação de sua atividade de armazenamento e uso com o Amazon S3 Storage Lens \(p. 1121\)](#)
- [Rastreamento de solicitações do Amazon S3 usando o AWS X-Ray \(p. 1186\)](#)

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 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 [Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#).

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 de classe de armazenamento fornece apenas recomendações para as classes IA padrão a padrão.

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 fornece visualizações de uso de armazenamento no console do Amazon S3 que são atualizadas diariamente. Também é possível exportar esses dados de uso diário para um bucket do S3 e visualizá-los em uma aplicação de planilha ou com ferramentas de business intelligence, como o Amazon QuickSight.

Há custos associados à análise da classe de armazenamento. Para obter informações sobre preços, consulte Gerenciamento e replicação [Definição de preço do Amazon S3](#).

Tópicos

- [Como configurar a análise de classe de armazenamento? \(p. 1117\)](#)
- [Como usar a análise de classe de armazenamento? \(p. 1117\)](#)

- [Como exportar dados de análise de classe de armazenamento? \(p. 1119\)](#)
- [Configurar análise de classe de armazenamento \(p. 1120\)](#)

Como 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 AWS SDKs 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 [Configurar análise de classe de armazenamento \(p. 1120\)](#).
- Para usar a API do Amazon S3, use a API REST [PutBucketAnalyticsConfiguration](#), ou equivalente, na AWS CLI ou nos AWS SDKs.

Como 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 entre 24 e 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.

A primeira vez que você configura um filtro, o console do Simple Storage Service (Amazon S3) pode levar algum tempo para analisar os dados.

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 informando que a análise está incompleta.

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 128 KB.
- 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 durante o período de observação.

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 durante o período de observação.

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 de 15 a 29 dias
- Objetos do Amazon S3 de 30 a 44 dias
- Objetos do Amazon S3 de 45 a 59 dias
- Objetos do Amazon S3 de 60 a 74 dias
- Objetos do Amazon S3 de 75 a 89 dias
- Objetos do Amazon S3 de 90 a 119 dias
- Objetos do Amazon S3 de 120 a 149 dias
- Objetos do Amazon S3 de 150 a 179 dias
- Objetos do Amazon S3 de 180 a 364 dias
- Objetos do Amazon S3 de 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 entre 24 e 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 pelos grupos etários de objetos predefinidos. O texto Frequently accessed (Acessado com frequência) ou Infrequently accessed (Acessado com pouca frequência) mostrado é uma ajuda visual para ajudá-lo no processo de criação do ciclo de vida.

Como 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 Amazon S3 Inventory e a análise do Simple Storage Service \(Amazon S3\) \(p. 526\)](#).

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 uma aplicação de planilha ou importar o arquivo para outras aplicações, 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	Count
8/17/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			
9/2/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						
8/22/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			
8/25/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			
9/6/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						
8/30/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						
8/28/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						
8/21/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			
9/5/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						

No final do relatório, a faixa etária de objeto é fornecida como ALL. As linhas ALL contêm totais cumulativos, incluindo objetos menores que 128 KB, para todas as faixas etárias desse dia.

8/24/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			
9/3/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			
8/28/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			
8/17/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			
8/25/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			
9/6/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			
9/4/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			
8/22/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			
8/21/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			
8/30/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			
8/20/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			

A próxima seção descreve as colunas usadas no relatório.

Layout de arquivos exportados

A tabela a seguir descreve o layout do arquivo exportado.

Configurar análise de classe de armazenamento

Usando a ferramenta de análise de classe de armazenamento do Amazon S3, é possível analisar padrões de acesso de armazenamento para ajudar a decidir quando fazer a transição dos dados certos para a classe de armazenamento certa. A 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). Para obter mais informações sobre STANDARD_IA, consulte as [perguntas frequentes do Amazon S3 e Uso de classes de armazenamento do Amazon S3 \(p. 731\)](#).

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 AWS SDKs para configurar 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 S3 Glacier Flexible Retrieval.

Se você quiser configurar a análise de classe de armazenamento para exportar suas descobertas como um arquivo.csv e o bucket de destino usar criptografia de bucket padrão com uma AWS KMS key, atualize a política de chaves do AWS KMS para conceder permissão ao Amazon S3 para criptografar o arquivo.csv. Para obter instruções, consulte [Conceder permissão para que o Amazon S3 use a AWS KMS key para criptografia \(p. 791\)](#).

Para obter mais informações sobre análises, consulte [Análise do Amazon S3 – Análise de classe de armazenamento \(p. 1116\)](#).

Uso do console do S3

Para configurar a análise de classe de armazenamento

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket para o qual você deseja configurar a análise de classe de armazenamento.
3. Escolha a guia Metrics.
4. Em Storage Class Analysis (Análise de classe de armazenamento), escolha Create analytics configuration (Criar configuração de análise).

5. Digite um nome para o filtro. Se você quiser analisar o bucket todo, deixe o campo Prefix (Prefixo) vazio.
6. No campo Prefix (Prefixo) digite o texto para o prefixo para os objetos que você deseja analisar.
7. Para adicionar uma tag, escolha Add tag. Insira uma chave e um valor para a tag. Você pode inserir um prefixo e várias tags.
8. Opcionalmente, você pode escolher Enable (Habilitar) em Export CSV (Exportar CSV) para exportar relatórios de análise para um arquivo simples com valores separados por vírgula (.csv). Escolha um bucket de destino onde o arquivo pode ser armazenado. Você pode digitar um prefixo para o bucket de destino. O bucket de destino deve estar na mesma Região da AWS que o bucket para o qual você está configurando a análise. O bucket de destino pode estar em uma Conta da AWS diferente.

Se o bucket de destino do arquivo .csv usar criptografia de bucket padrão com uma chave de KMS, você deverá atualizar a política de chaves do AWS KMS para conceder permissão do Amazon S3 para criptografar o arquivo .csv. Para obter instruções, consulte [Conceder permissão para que o Amazon S3 use a AWS KMS key para criptografia \(p. 791\)](#).

9. Selecione Create Configuration (Criar configuração).

O Amazon S3 cria uma política de bucket no bucket de destino que concede ao Amazon S3 permissão para gravação. Ela permite gravar os dados de exportação no bucket.

Se ocorrer um erro quando você tentar criar a política de bucket, serão fornecidas instruções sobre como resolvê-lo. Por exemplo, se você escolher um bucket de destino em outra Conta da AWS e não tiver permissões para ler e gravar na política de bucket, verá a mensagem a seguir. Você deve pedir ao proprietário do bucket de destino para adicionar a política de bucket exibida ao bucket de destino. Se a política não for adicionada ao bucket de destino, você não terá os dados de exportação porque o Amazon S3 não terá permissão para gravar no bucket de destino. Se o bucket de origem for de uma conta diferente da conta do usuário atual, o ID de conta correto do bucket de origem deverá ser substituído na política.

Para obter informações sobre os dados exportados e como o filtro funciona, consulte [Análise do Amazon S3 – Análise de classe de armazenamento \(p. 1116\)](#).

Uso dos REST API

Para configurar a Análise de classe de armazenamento usando a API REST, use [PutBucketAnalyticsConfiguration](#). Você também pode usar a operação equivalente com a AWS CLI ou os AWS SDKs.

Você pode usar as seguintes APIs REST para trabalhar com a Análise de classe de armazenamento:

- [Configuração de DELETE Bucket Analytics](#)
- [Configuração de GET Bucket Analytics](#)
- [Listar configuração de análise de bucket](#)

Avaliação de sua atividade de armazenamento e uso com o Amazon S3 Storage Lens

O Amazon S3 Storage Lens agrupa suas métricas de uso e atividade e exibe as informações no snapshot da conta na página inicial do console (Buckets) do Amazon S3, em painéis interativos ou por meio de uma exportação de métricas que você pode obter por download no formato CSV ou Parquet. Você pode usar o painel para visualizar insights e tendências, sinalizar discrepâncias e receber recomendações para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados. Você pode usar o S3 Storage Lens por meio do AWS Management Console, da AWS CLI, de AWS SDKs ou da API REST.

Como funciona o S3 Storage Lens

O Amazon S3 Storage Lens oferece uma visão única do uso e da atividade em seu armazenamento do Amazon S3. Com opções de detalhamento para gerar insights no nível da organização, conta, bucket, objeto ou até mesmo prefixo. Ele analisa as métricas de armazenamento para fornecer recomendações contextuais para ajudar a otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados.

Use o S3 Storage Lens para gerar insights resumidos, como descobrir quanto armazenamento você tem em toda a organização ou quais são os buckets e prefixos que mais crescem. Identifique discrepâncias em suas métricas de armazenamento e, em seguida, faça drill-down para investigar ainda mais a origem do pico de uso ou atividade.

Você pode avaliar seu armazenamento com base nas práticas recomendadas do S3, como analisar a porcentagem de buckets com criptografia ou bloqueio de objeto do S3 habilitado. Além disso, você pode identificar possíveis oportunidades de economia de custos, como, por exemplo, analisando sua atividade de solicitação por bucket para encontrar buckets em que os objetos possam ser transferidos para uma classe de armazenamento de menor custo. Para obter mais informações sobre os conceitos e a terminologia do S3 Storage Lens, consulte [Noções básicas sobre o Amazon S3 Storage Lens \(p. 1123\)](#).

Painel padrão

No [console do S3](#), o S3 Storage Lens fornece um painel padrão interativo que é atualizado diariamente. As métricas desse painel também são resumidas no snapshot da conta na página inicial do console do S3 (Buckets). Você pode criar outros painéis e definir o escopo deles por conta (para usuários do AWS Organizations), Regiões da AWS e buckets do S3 para fornecer [métricas de uso](#) gratuitamente. Por um custo adicional, você pode atualizar para receber métricas e recomendações avançadas. Isso inclui métricas de uso com agregação em nível de prefixo, métricas de atividade agregadas por bucket e recomendações contextuais (disponíveis apenas no console do Amazon S3) e publicação do Amazon CloudWatch. Para obter informações sobre como trabalhar com o painel do S3 Storage Lens, consulte [Usar o Amazon S3 Storage Lens no console \(p. 1158\)](#).

Exportação de métricas

Além do painel no console do S3, é possível exportar métricas em formato CSV ou Parquet para um bucket do S3 de sua escolha para análise posterior. Para mais informações, consulte [Exibição de métricas de lente de armazenamento do Amazon S3 usando uma exportação de dados \(p. 1134\)](#).

Opção de publicação no Amazon CloudWatch

É possível publicar métricas de uso e atividade do S3 Storage Lens no Amazon CloudWatch para criar uma visão unificada de sua integridade operacional nos [painéis](#) do CloudWatch. Também é possível usar os recursos do CloudWatch, como alarmes e ações acionadas, matemática métrica e detecção de anomalias para monitorar e realizar ações sobre as métricas do S3 Storage Lens. Além disso, as APIs do CloudWatch permitem que aplicações, inclusive provedores de terceiros, acessem as métricas do S3 Storage Lens. A opção de publicação no CloudWatch está disponível para painéis atualizados para métricas e recomendações avançadas do S3 Storage Lens. Para obter mais informações sobre suporte às métricas do S3 Storage Lens no CloudWatch, consulte [Monitorar métricas do S3 Storage Lens no CloudWatch \(p. 1138\)](#).

Para obter mais informações sobre como usar o S3 Storage Lens, consulte [the section called “Trabalhar com o S3 Storage Lens” \(p. 1158\)](#). Para obter mais informações sobre a definição de preço do S3 Storage Lens, consulte [Definição de preço do Amazon S3](#).

Tópicos

- [Noções básicas sobre o Amazon S3 Storage Lens \(p. 1123\)](#)
- [Usar o Amazon S3 Storage Lens com AWS Organizations \(p. 1128\)](#)
- [Permissões do Amazon S3 Storage Lens \(p. 1130\)](#)

- [Exibição de métricas de uso de armazenamento e atividade com o Amazon S3 Storage Lens \(p. 1133\)](#)
- [Como usar o Amazon S3 Storage Lens para otimizar seus custos de armazenamento \(p. 1149\)](#)
- [Glossário de métricas de lente de armazenamento do Amazon S3 \(p. 1152\)](#)
- [Trabalhar com o Amazon S3 Storage Lens usando o console e a API \(p. 1158\)](#)

Noções básicas sobre o Amazon S3 Storage Lens

O Amazon S3 Storage Lens oferece uma visão única do uso e da atividade do armazenamento de objetos em todo o armazenamento do Amazon S3. Ele inclui opções de detalhamento para gerar insights no nível da organização, conta, região, bucket ou até mesmo prefixo.

O Amazon S3 Storage Lens agrupa suas métricas de uso e atividade e exibe as informações no snapshot da conta na página inicial do console (Buckets) do Amazon S3, em painéis interativos ou por meio de uma exportação de métricas que você pode obter por download no formato CSV ou Parquet. Você pode usar o painel para visualizar insights e tendências, sinalizar discrepâncias e receber recomendações para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados. Você pode usar o S3 Storage Lens por meio do AWS Management Console, da AWS CLI, de AWS SDKs ou da API REST.

Conceitos e terminologia do Amazon S3 Storage Lens

Esta seção contém a terminologia e os conceitos essenciais para entender e usar o Amazon S3 Storage Lens com sucesso.

Tópicos

- [Configuração \(p. 1123\)](#)
- [Painel padrão \(p. 1123\)](#)
- [Painéis \(p. 1124\)](#)
- [Snapshot da conta \(p. 1124\)](#)
- [Exportação de métricas \(p. 1125\)](#)
- [Região inicial \(p. 1125\)](#)
- [Período de retenção \(p. 1125\)](#)
- [Tipos de métricas \(p. 1126\)](#)
- [Recomendações \(p. 1126\)](#)
- [Seleção de métricas \(p. 1127\)](#)
- [S3 Storage Lens e AWS Organizations \(p. 1128\)](#)

Configuração

O Amazon S3 Storage Lens requer uma configuração que contém as propriedades usadas para agregar métricas em seu nome para um único painel ou exportação. Isso inclui todas as seções ou parciais do armazenamento da conta da organização, incluindo filtragem por região, bucket e escopo de nível de prefixo (disponível somente com métricas avançadas). Ele inclui informações sobre se você escolheu métricas gratuitas ou métricas e recomendações avançadas. Ele também inclui se uma exportação de métricas é necessária e informações sobre onde colocar a exportação de métricas, se aplicável.

Painel padrão

O painel padrão do S3 Storage Lens no console é chamado default-account-dashboard. O S3 preconfigura o painel para visualizar os insights resumidos e as tendências de uso agregado de armazenamento e

as métricas de atividade de armazenamento agregado de toda a sua conta e atualiza-as diariamente no console do Amazon S3. Você não pode modificar o escopo de configuração do painel padrão, mas pode atualizar a seleção de métricas de métricas gratuitas para as métricas e recomendações avançadas pagas. Você também pode configurar a exportação de métricas opcionais ou até mesmo desabilitar o painel. No entanto, você não pode excluir o painel padrão.

Note

Se você desabilitar seu painel padrão, ele não será mais atualizado e você não receberá mais nenhuma nova métrica diária no S3 Storage Lens ou no snapshot da conta na página inicial do S3 (Buckets). Você ainda pode consultar dados históricos no painel até o período de 14 dias em que os dados estarão disponíveis para consulta expirar, ou 15 meses, se tiver assinatura das métricas e recomendações avançadas do painel. Você pode reativar o painel dentro do período de validade para acessar esses dados.

Painéis

Você também pode usar o Amazon S3 Storage Lens para configurar painéis que visualizem insights resumidos e tendências de uso de armazenamento agregado e métricas de atividade, atualizadas diariamente no console do Amazon S3. Você pode criar e modificar painéis do S3 Storage Lens para expressar todas ou parte das seções do armazenamento da organização ou da conta. É possível filtrar por Região da AWS, bucket e prefixo (disponível somente com métricas e recomendações avançadas). Você também pode desativar ou excluir painéis.

Note

- Você pode usar o S3 Storage Lens para criar até 50 painéis por região inicial.
- Se você desabilitar um painel, ele não será mais atualizado e você não receberá mais nenhuma nova métrica diária. Você ainda pode ver dados históricos até o período de expiração de 14 dias (ou 15 meses, se você se inscreveu em métricas avançadas e recomendações para esse painel). Você pode reativar o painel dentro do período de expiração para acessar esses dados.
- Se você excluir seu painel, você perderá todas as configurações do painel. Você não receberá mais nenhuma nova métrica diária e também perderá o acesso aos dados históricos associados a esse painel. Se você quiser acessar os dados históricos de um painel excluído, você deve criar outro painel com o mesmo nome na mesma região inicial.
- Os painéis de nível da organização só podem ser limitados a um escopo regional.

Snapshot da conta

O snapshot da conta do S3 Storage Lens exibe o armazenamento total, a contagem de objetos e o tamanho médio do objeto na página inicial (Buckets) do console do S3, resumindo as métricas do painel padrão. Isso permite acesso rápido a insights sobre o armazenamento sem ter que sair da página Buckets. O snapshot da conta também fornece acesso com um clique à página do S3 Storage Lens, na qual é possível fazer uma análise mais profunda das tendências de uso e atividade por Região da AWS, classe de armazenamento, bucket ou prefixo.

Você pode atualizar a seleção de métricas em seu painel de conta-padrão de métricas gratuitas para métricas e recomendações avançadas pagas. Em seguida, você pode exibir todas as solicitações, bytes carregados e bytes obtidos por download no snapshot da conta do S3 Storage Lens.

Note

- Não é possível modificar o escopo do painel padrão porque ele está vinculado ao snapshot da conta. No entanto, você pode atualizar a seleção de métricas de métricas gratuitas para métricas e recomendações avançadas pagas.

- Se você desabilitar o painel padrão, o snapshot da conta não será mais atualizado. Você pode reativar o painel de conta-padrão para retomar a exibição de métricas no snapshot da conta.

Exportação de métricas

Uma exportação de métricas do S3 Storage Lens é um arquivo que contém todas as métricas identificadas na configuração do S3 Storage Lens. Essas informações são geradas diariamente no formato CSV ou Parquet em um bucket do S3 de sua escolha para análise posterior. O bucket do S3 para a exportação de métricas deve estar na mesma região da configuração do S3 Storage Lens. Você pode gerar uma exportação de métricas do S3 Storage Lens a partir do console do S3 editando a configuração do painel ou usando a AWS CLI e SDKs.

Região inicial

A região inicial é a Região da AWS em que todas as métricas do Amazon S3 Storage Lens para um determinado painel ou configuração são armazenadas. Você deve escolher uma região inicial ao criar o painel ou configuração do Amazon S3 Storage Lens. Depois que uma região inicial for atribuída, ela não poderá ser alterada.

Note

Há suporte para a criação de uma região inicial nas seguintes regiões:

- Leste dos EUA (Norte da Virgínia), us-east-1
- Leste dos EUA (Ohio), us-east-2
- Oeste dos EUA (Norte da Califórnia), us-west-1
- Oeste dos EUA (Oregon), us-west-2
- Ásia-Pacífico (Mumbai), ap-south-1
- Ásia-Pacífico (Osaka) - ap-northeast-3
- Ásia-Pacífico (Seul), ap-northeast-2
- Ásia-Pacífico (Singapura), ap-southeast-1
- Ásia-Pacífico (Sydney), ap-southeast-2
- Ásia-Pacífico (Tóquio), ap-northeast-1
- Canadá (Central), ca-central-1
- China (Pequim), cn-north-1
- China (Ningxia), cn-northwest-1
- Europa (Frankfurt), eu-central-1
- Europa (Irlanda), eu-west-1
- Europa (Londres), eu-west-2
- Europa (Paris), eu-west-3
- Europa (Estocolmo), eu-north-1
- América do Sul (São Paulo), sa-east-1

Período de retenção

As métricas do Amazon S3 Storage Lens são mantidas para que você possa ver tendências históricas e comparar diferenças no uso e na atividade do armazenamento ao longo do tempo. Todas as métricas do S3 Storage Lens são mantidas durante um período de 15 meses. Um painel com apenas métricas gratuitas pode exibir métricas com até 14 dias de idade. Um painel com métricas e recomendações avançadas habilitadas pode exibir métricas com até 15 meses de idade.

Você pode usar as métricas do Amazon S3 Storage Lens para consultar o histórico de tendências e comparar diferenças no uso e na atividade do armazenamento ao longo do tempo. As métricas estão disponíveis por um período específico. A duração da retenção depende da [seleção de métricas](#) e não pode ser modificada. As métricas gratuitas estão disponíveis para consultas por 14 dias, e as métricas avançadas estão disponíveis para consultas por 15 meses.

Tipos de métricas

O S3 Storage Lens oferece dois tipos de métricas de armazenamento: uso e atividade.

- Métricas de uso

O S3 Storage Lens coleta métricas de uso para todos os painéis e configurações. As métricas de uso descrevem o tamanho, a quantidade e as características do armazenamento. Isso inclui o total de bytes armazenados, a contagem de objetos e a média de tamanho do objeto. Inclui também métricas que descrevem a utilização de recursos, como bytes criptografados, ou excluem contagens de objetos de mercado. Para obter mais informações sobre as métricas de uso agregadas pelo S3 Storage Lens, consulte [Glossário de métricas](#).

- Métricas de atividade

O S3 Storage Lens agrupa métricas de atividade para todos os painéis e configurações que têm o tipo de métricas avançadas e recomendações habilitado. As métricas de atividade descrevem os detalhes da frequência com que seu armazenamento é solicitado. Isso inclui o número de solicitações por tipo, bytes de upload e download e erros. Para obter mais informações sobre as métricas de atividade agregadas pelo S3 Storage Lens, consulte [Glossário de métricas](#).

Recomendações

O S3 Storage Lens fornece recomendações automatizadas para ajudá-lo a otimizar seu armazenamento. As recomendações são colocadas contextualmente ao lado de métricas relevantes no painel do S3 Storage Lens. Os dados históricos não são elegíveis para recomendações porque as recomendações são relevantes para o que está acontecendo no período mais recente. As recomendações só aparecem quando são relevantes.

As recomendações do S3 Storage Lens vêm nos seguintes formulários:

- Sugestões

As sugestões alertam você sobre tendências de uso e atividade de armazenamento que podem indicar uma oportunidade de otimização de custos de armazenamento ou uma prática recomendada de proteção de dados. Você pode usar os tópicos sugeridos no Guia do usuário do Amazon S3 e no painel do S3 Storage Lens para especificar mais detalhes sobre regiões, buckets ou prefixos específicos e obter mais ajuda.

- Chamadas

Chamadas são recomendações que alertam você para anomalias interessantes dentro do uso e atividade do armazenamento durante um período que pode precisar de mais atenção ou monitoramento.

- Chamadas discrepantes

O S3 Storage Lens fornece chamadas para métricas que são discrepantes, com base em sua tendência recente de 30 dias. A discrepância é calculada usando uma pontuação padrão, também conhecida como pontuação z. Nesta pontuação, a métrica do dia atual é subtraída da média dos últimos 30 dias para essa métrica e, em seguida, dividida pelo desvio padrão para essa métrica nos últimos 30 dias. A pontuação resultante geralmente é entre -3 e +3. Este número representa o número de desvios padrão que a métrica do dia atual é da média.

O S3 Storage Lens considera métricas com uma pontuação >2 ou <-2 como discrepâncias porque são maiores ou inferiores a 95% dos dados normalmente distribuídos.

- Chamadas de alterações significativas

A chamada de alteração significativa se aplica a métricas que devem ser alteradas com menos frequência. Portanto, ela é definida para uma sensibilidade maior do que o cálculo de discrepância, que geralmente é na faixa de +/- 20 por cento versus o dia anterior, semana ou mês.

Abordar chamadas em seu uso e atividade de armazenamento — se você receber uma chamada de alteração significativa, não é necessariamente um problema e pode ser o resultado de uma alteração antecipada no armazenamento. Por exemplo, você pode ter adicionado recentemente um grande número de novos objetos, excluído um grande número de objetos ou fez alterações planejadas semelhantes.

Se você vir uma chamada de alteração significativa no seu painel, tome nota e determine se ela pode ser explicada por circunstâncias recentes. Caso contrário, use o painel do S3 Storage Lens para fazer drill-down a fim de obter mais detalhes para entender as regiões, buckets ou prefixos específicos que estão gerando a flutuação.

- Lembretes

Os lembretes fornecem insights sobre como o Amazon S3 funciona. Eles podem ajudá-lo a aprender mais sobre maneiras de usar os recursos do S3 para reduzir os custos de armazenamento ou aplicar as práticas recomendadas de proteção de dados.

Seleção de métricas

O S3 Storage Lens oferece duas seleções de métricas que você pode escolher para seu painel e exportar: métricas gratuitas e métricas e recomendações avançadas.

- Métricas gratuitas

O S3 Storage Lens oferece métricas gratuitas para todos os painéis e configurações. As métricas gratuitas contêm métricas relevantes para o uso do armazenamento. Isso inclui o número de buckets, os objetos em sua conta e em que estado eles estão. Todas as métricas gratuitas são coletadas diariamente. Os dados ficam disponíveis para consultas por 14 dias. Para obter mais informações sobre quais métricas de uso são agregadas pelo S3 Storage Lens, consulte [Glossário de métricas](#).

- Métricas e recomendações avançadas

O S3 Storage Lens oferece métricas gratuitas para todos os painéis e configurações, bem como a opção de atualizar para métricas e recomendações avançadas. Aplicam-se cobranças adicionais. Para obter mais informações, consulte [Definição de preço do Amazon S3](#). As métricas avançadas contêm todas as métricas de uso incluídas nas métricas gratuitas. Isso inclui o número de buckets, os objetos em sua conta e em que estado eles estão.

Essa seleção de métricas também fornece recomendações para ajudá-lo a otimizar seu armazenamento. As recomendações são colocadas contextualmente ao lado de métricas relevantes no painel.

As métricas e recomendações avançadas também contam com os seguintes recursos:

- Métricas de atividade: gere métricas adicionais agregadas por bucket, como solicitações, bytes carregados ou baixados e erros. Os dados de métricas de atividade são relevantes para sua atividade de armazenamento. Isso inclui o número de solicitações, verificações e erros em relação ao escopo de configuração e em que estado eles estão.
- Publicação no Amazon CloudWatch: publique métricas de uso e atividade do S3 Storage Lens no CloudWatch para criar uma visão unificada de sua integridade operacional nos [painéis](#) do CloudWatch. Também é possível usar as APIs e os recursos do CloudWatch, como alarmes e ações acionadas, matemática métrica e detecção de anomalias, para monitorar e realizar ações sobre as métricas do S3 Storage Lens. Para obter mais informações, consulte [Monitorar métricas do S3 Storage Lens no CloudWatch \(p. 1138\)](#).

- Agregação de prefixos: cole métricas de uso no nível de prefixo. Métricas de nível de prefixo não são publicadas no CloudWatch.

Todas as métricas avançadas são coletadas diariamente. Os dados ficam disponíveis para consultas por 15 meses. Para obter mais informações sobre as métricas de armazenamento agregadas pelo S3 Storage Lens, consulte [Glossário de métricas de lente de armazenamento do Amazon S3 \(p. 1152\)](#).

Note

As recomendações só estão disponíveis quando você usa o painel do S3 Storage Lens no console do Amazon S3 e não por meio da AWS CLI e SDKs.

S3 Storage Lens e AWS Organizations

O AWS Organizations é um serviço da AWS que ajuda você a agregar todas as suas Contas da AWS em uma hierarquia da organização. O Amazon S3 Storage Lens funciona com o AWS Organizations para fornecer uma visão única do uso e da atividade do armazenamento de objetos em seu armazenamento do Amazon S3.

Para obter mais informações, consulte [Usar o Amazon S3 Storage Lens com AWS Organizations \(p. 1128\)](#).

- Acesso confiável

Usando a conta de gerenciamento da sua organização, você deve habilitar o acesso confiável para o S3 Storage Lens para agregar métricas de armazenamento e dados de uso para todas as contas membros da sua organização. Em seguida, você pode criar painéis ou exportações para sua organização usando sua conta de gerenciamento ou dando acesso ao administrador delegado a outras contas da organização.

Você pode desativar o acesso confiável para o S3 Storage Lens a qualquer momento, o que impede o S3 Storage Lens de agregar métricas para sua organização.

- Administrador delegado

Você pode criar painéis e métricas para o S3 Storage Lens para sua organização usando sua conta de gerenciamento do AWS Organizations ou concedendo acesso de administrador delegado a outras contas em sua organização. Você pode cancelar o registro de administradores delegados a qualquer momento, o que impede o S3 Storage Lens de coletar dados em nível de organização.

Para obter mais informações, consulte [Amazon S3 Storage Lens e AWS Organizations](#) no Guia do usuário do AWS Organizations.

Funções vinculadas ao serviço do Amazon S3 Storage Lens

Juntamente com o acesso confiável do AWS Organizations, o Amazon S3 Storage Lens usa funções vinculadas ao serviço do AWS Identity and Access Management (IAM). Uma função vinculada a serviço é um tipo exclusivo de função do IAM vinculada diretamente ao S3 Storage Lens. As funções vinculadas a serviços são predefinidas pelo S3 Storage Lens e incluem todas as permissões necessárias para coletar métricas diárias de uso de armazenamento e atividade de contas de membros em sua organização.

Para obter mais informações, consulte [Usar funções vinculadas de serviço ao Amazon S3 Storage Lens](#).

Usar o Amazon S3 Storage Lens com AWS Organizations

Você pode usar o Amazon S3 Storage Lens para coletar métricas de armazenamento e dados de uso de todas as Contas da AWS que fazem parte da hierarquia do AWS Organizations. Para fazer isso,

use o AWS Organizations e habilite o acesso confiável do S3 Storage Lens utilizando sua conta de gerenciamento do AWS Organizations.

Depois de habilitar o acesso confiável, você pode adicionar acesso de administrador delegado às contas em sua organização. Essas contas podem então criar configurações e painéis do S3 Storage Lens que coletam métricas de armazenamento e dados do usuário em toda a organização.

Para obter mais informações sobre como habilitar o acesso confiável, consulte [Amazon S3 Storage Lens e AWS Organizations](#) no Guia do usuário do AWS Organizations.

Tópicos

- [Ativando acesso confiável para o S3 Storage Lens \(p. 1129\)](#)
- [Desativação do acesso confiável para o S3 Storage Lens \(p. 1129\)](#)
- [Registro de um administrador delegado para o S3 Storage Lens \(p. 1130\)](#)
- [Cancelamento do registro de um administrador delegado para o S3 Storage Lens \(p. 1130\)](#)

Ativando acesso confiável para o S3 Storage Lens

Ao habilitar o acesso confiável, você permite que o Amazon S3 Storage Lens tenha acesso à sua hierarquia do AWS Organizations, associação e estrutura por meio das APIs do AWS Organizations. O S3 Storage Lens se torna um serviço confiável para toda a estrutura de sua organização.

Sempre que uma configuração de painel é criada, o S3 Storage Lens cria funções vinculadas a serviços nas contas de gerenciamento ou administrador delegado da sua organização. A função vinculada ao serviço concede permissões do S3 Storage Lens para descrever organizações, listar contas, verificar uma lista de acesso de serviço da AWS para as organizações e obter administradores delegados para a organização. O S3 Storage Lens pode garantir que tenha acesso para coletar as métricas de uso de armazenamento entre contas e atividades de contas em suas organizações. Para obter mais informações, consulte [Usar funções vinculadas a serviço do Amazon S3 Storage Lens](#).

Depois de habilitar o acesso confiável, você pode atribuir acesso de administrador delegado a contas em sua organização. Quando uma conta é marcada como administrador delegado para um serviço, a conta recebe autorização para acessar todas as APIs da organização somente leitura. Isso fornece visibilidade aos membros e estruturas da sua organização para que eles também possam criar painéis do S3 Storage Lens.

Note

Somente a conta de gerenciamento pode habilitar o acesso confiável para o Amazon S3 Storage Lens.

Desativação do acesso confiável para o S3 Storage Lens

Ao desativar o acesso confiável, você limita o S3 Storage Lens a trabalhar apenas em nível de conta. Além disso, cada titular da conta só pode ver os benefícios do S3 Storage Lens limitados ao escopo de sua conta e não à organização inteira. Os painéis que requerem acesso confiável não são mais atualizados, mas reterão seus dados históricos de acordo com o período em que [os dados ficarão disponíveis para consulta](#).

A remoção de uma conta como administrador delegado limitará o acesso às métricas do painel do S3 Storage Lens para funcionar apenas em nível de conta. Todos os painéis organizacionais que foram criados não são mais atualizados, mas reterão seus dados históricos de acordo com o período em que os dados ficarão disponíveis para consulta.

Note

- Essa ação também impede automaticamente todos os painéis no nível da organização de coletar e agregar métricas de armazenamento.

- Suas contas de gerenciamento e administrador delegado ainda poderão ver os dados históricos de seus painéis de nível de organização que saem durante o período em que os dados ficarem disponíveis para consulta.

Registro de um administrador delegado para o S3 Storage Lens

Você pode criar painéis no nível da organização usando a conta de gerenciamento da sua organização ou uma conta de administrador delegada. As contas de administrador delegado permitem que outras contas além da sua conta de gerenciamento criem painéis no nível da organização. A conta mestre de uma organização pode registrar e cancelar o registro de outras contas como administradores delegados da organização.

Para registrar um administrador delegado usando o console do Amazon S3, consulte [Registro de administradores delegados para o S3 Storage Lens \(p. 1169\)](#).

Você também pode registrar um administrador delegado usando a API REST do AWS Organizations, AWS CLI ou SDKs da conta de gerenciamento. Para obter mais informações, consulte [RegisterDelegatedAdministrator](#) na Referência da API do AWS Organizations.

Note

Antes de designar um administrador delegado usando a API REST do AWS Organizations, AWS CLI ou SDKs, você deve chamar a operação [EnableAwsOrganizationsAccess](#).

Cancelamento do registro de um administrador delegado para o S3 Storage Lens

Você também pode cancelar o registro de uma conta de administrador delegado. As contas de administrador delegado permitem que outras contas além da sua conta de gerenciamento criem painéis no nível da organização. Somente a conta de gerenciamento de uma organização pode cancelar o registro de contas como administradores delegados para a organização.

Para cancelar o registro de um administrador delegado usando o console do S3, consulte [Cancelamento do registro de administradores delegados para o S3 Storage Lens \(p. 1169\)](#).

Você também pode cancelar o registro de um administrador delegado usando a API REST do AWS Organizations, AWS CLI ou SDKs da conta de gerenciamento. Para obter mais informações, consulte [DeregisterDelegatedAdministrator](#) na Referência da API do AWS Organizations.

Note

- Essa ação também impede automaticamente todos os painéis no nível da organização criados por esse administrador delegado de agregar novas métricas de armazenamento.
- As contas de administrador delegado ainda poderão visualizar os dados históricos desses painéis enquanto os dados estiverem disponíveis para consultas.

Permissões do Amazon S3 Storage Lens

O Amazon S3 Storage Lens requer novas permissões no AWS Identity and Access Management (IAM) para autorizar o acesso às ações do S3 Storage Lens. Você pode anexar a política a usuários, grupos ou funções do IAM para conceder a eles permissões para ativar ou desativar o S3 Storage Lens ou acessar qualquer painel ou configuração do S3 Storage Lens.

O usuário ou função do IAM deve pertencer à conta que criou ou possui o painel ou a configuração, a menos que sua conta seja membro do AWS Organizations e você tenha acesso para criar painéis no nível da organização por sua conta de gerenciamento como administrador delegado.

Note

- Você não pode usar as credenciais de usuário root da sua conta para visualizar os painéis do Amazon S3 Storage Lens. Para acessar os painéis do S3 Storage Lens, você deve conceder as permissões do IAM necessárias a um usuário do IAM novo ou existente. Em seguida, faça login com essas credenciais de usuário para acessar os painéis do S3 Storage Lens. Para obter mais informações, consulte [Práticas recomendadas do AWS Identity and Access Management](#).
- O uso do S3 Storage Lens no console do Amazon S3 pode exigir várias permissões. Por exemplo; para editar um painel no console, você precisa das seguintes permissões:
 - `s3>ListStorageLensConfigurations`
 - `s3:GetStorageLensConfiguration`
 - `s3:PutStorageLensConfiguration`

Tópicos

- [Definição de permissões de conta para usar o S3 Storage Lens \(p. 1131\)](#)
- [Configuração de permissões para usar o Amazon S3 Storage Lens com AWS Organizations \(p. 1132\)](#)

Definição de permissões de conta para usar o S3 Storage Lens

Permissões do IAM relacionadas ao Amazon S3 Storage Lens

Ação	Permissões do IAM
Criar ou atualizar um painel do S3 Storage Lens no console do Amazon S3.	<code>s3>ListStorageLensConfigurations</code> <code>s3:GetStorageLensConfiguration</code> <code>s3:GetStorageLensConfigurationTagging</code> <code>s3:PutStorageLensConfiguration</code> <code>s3:PutStorageLensConfigurationTagging</code>
Obtenha tags de um painel do S3 Storage Lens no console do Amazon S3.	<code>s3>ListStorageLensConfigurations</code> <code>s3:GetStorageLensConfigurationTagging</code>
Veja um painel do S3 Storage Lens no console do Amazon S3.	<code>s3>ListStorageLensConfigurations</code> <code>s3:GetStorageLensConfiguration</code> <code>s3:GetStorageLensDashboard</code>
Exclua um painel do S3 Storage Lens no console do Amazon S3.	<code>s3>ListStorageLensConfigurations</code> <code>s3:GetStorageLensConfiguration</code> <code>s3>DeleteStorageLensConfiguration</code>
Crie ou atualize uma configuração do S3 Storage Lens na AWS CLI ou no SDK.	<code>s3:PutStorageLensConfiguration</code> <code>s3:PutStorageLensConfigurationTagging</code>
Obtenha tags de uma configuração do S3 Storage Lens na AWS CLI ou no SDK.	<code>s3:GetStorageLensConfigurationTagging</code>

Ação	Permissões do IAM
Veja uma configuração da lente de armazenamento do S3 na AWS CLI ou no SDK.	s3:GetStorageLensConfiguration
Exclua uma configuração do S3 Storage Lens na AWS CLI ou no SDK.	s3>DeleteStorageLensConfiguration

Note

- Você pode usar tags de recurso em uma política do IAM para gerenciar permissões.
- Um usuário/função do IAM com essas permissões pode ver métricas de buckets e prefixos dos quais eles podem não ter permissão direta para ler ou listar objetos.
- Para configurações do S3 Storage Lens com métricas avançadas e recomendações agregadas no nível do prefixo, se o prefixo selecionado corresponder às chaves de objeto, ele poderá mostrar a chave do objeto como seu prefixo até o delimitador e a profundidade máxima selecionada.
- Para as exportações de métricas, armazenadas em um bucket em sua conta, as permissões são concedidas usando a permissão s3:GetObject existente na política do IAM. Da mesma forma, para uma entidade do AWS Organizations, a conta de gerenciamento da organização ou administrador delegado pode usar políticas do IAM para gerenciar permissões de acesso para configurações e painéis no nível da organização.

Configuração de permissões para usar o Amazon S3 Storage Lens com AWS Organizations

Você pode usar o Amazon S3 Storage Lens para coletar métricas de armazenamento e dados de uso de todas as contas que fazem parte da hierarquia do AWS Organizations. A seguir estão as ações e permissões relacionadas ao uso do S3 Storage Lens com organizações.

AWS OrganizationsPermissões do IAM relacionadas ao para usar o S3 Storage Lens

Ação	Permissões do IAM
Habilite o acesso confiável para o S3 Storage Lens para sua organização.	organizations:EnableAWSServiceAccess
Desative o acesso confiável S3 Storage Lens para sua organização.	organizations:DisableAWSServiceAccess
Registre um administrador delegado para criar painéis ou configurações do S3 Storage Lens para sua organização.	organizations:RegisterDelegatedAdministrator
Cancelle o registro de um administrador delegado para criar painéis ou configurações do S3 Storage Lens para sua organização.	organizations:DeregisterDelegatedAdministrator
Permissões adicionais para criar configurações para toda a organização do S3 Storage Lens	organizations:DescribeOrganization organizations>ListAccounts organizations>ListAWSServiceAccessForOrganization organizations>ListDelegatedAdministrators

Ação	Permissões do IAM
	iam:CreateServiceLinkedRole

Exibição de métricas de uso de armazenamento e atividade com o Amazon S3 Storage Lens

Por padrão, todos os painéis são configurados com métricas gratuitas, que incluem [métricas de uso](#) agregadas no nível do bucket, e os dados ficam disponíveis para consultas por 14 dias. Isso significa que você pode ver todas as métricas de uso que o S3 Storage Lens agrupa e seus dados estarão disponíveis para consultas 14 dias a partir do dia em que foram agrupados.

As métricas e recomendações avançadas incluem métricas de uso com agregação em nível de prefixo, métricas de atividade agregadas por bucket e recomendações contextuais (disponíveis apenas no painel). As [métricas de atividade](#) ficam disponíveis para consultas por 15 meses. Há cobranças adicionais pelo uso do S3 Storage Lens com métricas avançadas. Para obter mais informações, consulte [Definição de preço do Amazon S3](#).

Tópicos

- [Exibição das métricas da lente de armazenamento do S3 nos painéis \(p. 1133\)](#)
- [Exibição de métricas de lente de armazenamento do Amazon S3 usando uma exportação de dados \(p. 1134\)](#)
- [Monitorar métricas do S3 Storage Lens no CloudWatch \(p. 1138\)](#)

Exibição das métricas da lente de armazenamento do S3 nos painéis

O S3 Storage Lens fornece um painel contendo métricas de uso sem custo adicional. Se você deseja receber métricas e recomendações avançadas, incluindo métricas de uso e atividade, agregações de prefixes e recomendações contextuais no painel, você deve selecioná-lo na página de configuração do painel no console do Amazon S3.

O painel fornece uma visualização interativa para o uso do armazenamento e as métricas de atividade. Você pode visualizar tendências em toda a organização ou ver tendências mais granulares por Conta da AWS, Região da AWS, classe de armazenamento, bucket do S3 ou prefixo.

Se sua conta for membro do AWS Organizations, você também poderá ver o uso e a atividade do armazenamento para toda a sua organização em contas membros. Essas informações estão disponíveis para você, desde que o S3 Storage Lens tenha recebido acesso confiável à sua organização e você seja uma conta de gerenciamento autorizado ou administrador delegado.

Use o painel interativo para explorar suas tendências e insights de uso de armazenamento e atividades, além de obter recomendações contextuais para práticas recomendadas para otimizar seu armazenamento. Para obter mais informações, consulte [Noções básicas sobre o Amazon S3 Storage Lens \(p. 1123\)](#).

O Amazon S3 preconfigura o painel padrão do S3 Storage Lens para ajudá-lo a visualizar insights resumidos e tendências de uso agregado de armazenamento e métricas de atividade de toda a sua conta (atualização opcional). Você não pode modificar o escopo de configuração padrão do painel, mas a seleção de métricas pode ser atualizada de métricas gratuitas para as métricas e recomendações avançadas pagas. Você pode configurar a exportação de métricas opcionais ou até mesmo desabilitar o painel. No entanto, o painel padrão não pode ser excluído.

Além do painel padrão criado pelo Amazon S3, você também pode criar painéis personalizados com escopo para as contas, regiões, buckets e prefixes da sua própria organização (somente no nível da

conta). Esses painéis personalizados podem ser editados, excluídos e desativados. As informações resumidas do painel padrão são exibidas na seção Snapshot da conta na página inicial do console do S3 (listas de buckets).

Note

O painel do Amazon S3 Storage Lens está disponível apenas no console do Amazon S3. Para obter mais informações, consulte [Exibição de um painel do Amazon S3 Storage Lens \(p. 1159\)](#).

Exibição de métricas de lente de armazenamento do Amazon S3 usando uma exportação de dados

As métricas do Amazon S3 Storage Lens são geradas diariamente em métricas formatadas CSV ou Apache Parquet exporta arquivos e colocadas em um bucket do S3 em sua conta. A partir daí, você pode ingerir as métricas exportadas para as ferramentas de análise de sua escolha, como o Amazon QuickSight e o Amazon Athena, onde você pode analisar o uso do armazenamento e as tendências de atividades.

Tópicos

- [Usar uma AWS KMS key para criptografar suas exportações de métricas \(p. 1134\)](#)
- [O que é um manifesto de exportação do S3 Storage Lens? \(p. 1135\)](#)
- [Compreendendo o esquema de exportação do Amazon S3 Storage Lens \(p. 1136\)](#)

Usar uma AWS KMS key para criptografar suas exportações de métricas

Para conceder permissão ao Amazon S3 Storage Lens para criptografar usando uma chave gerenciada pelo cliente, é necessário usar uma política de chaves. Para atualizar sua política de chaves para poder usar uma chave do KMS para criptografar as exportações de métricas do S3 Storage Lens, siga estas etapas.

Para conceder permissões de criptografia usando a chave do KMS

1. Faça login no AWS Management Console usando a Conta da AWS que é proprietária da chave gerenciada pelo cliente.
2. Abra o console do AWS KMS em <https://console.aws.amazon.com/kms>.
3. Para alterar a Região da AWS, use o Region selector (Seletor de regiões) no canto superior direito da página.
4. No painel de navegação, escolha Customer managed keys (Chaves gerenciadas pelo cliente).
5. Em Customer managed keys (Chaves gerenciadas pelo cliente), escolha a chave que você deseja usar para criptografar as exportações de métricas. As AWS KMS keys são específicas da região e devem estar na mesma região que o bucket S3 de destino de exportação de métricas.
6. Em Key policy (Política de chave), escolha Switch to policy view (Alternar para visualização de política).
7. Para atualizar a política de chave, escolha Edit (Editar).
8. Em Edit key policy (Editar política de chave), adicione a seguinte política de chave à existente.

```
{  
    "Sid": "Allow Amazon S3 Storage Lens use of the KMS key",  
    "Effect": "Allow",  
    "Principal": {  
        "Service": "storage-lens.s3.amazonaws.com"  
    },  
    "Action": [  
        "kms:GenerateDataKey"  
    ],  
    "Resource": "*",  
    "Condition": {
```

```
        "StringEquals": {
            "aws:SourceArn": "arn:aws:s3:us-east-1:source-account-id:storage-lens/your-
            dashboard-name",
            "aws:SourceAccount": "source-account-id"
        }
    }
}
```

9. Selecione Save changes.

Para obter mais informações sobre como criar chaves gerenciadas pelo cliente e usar políticas de chave, consulte os seguintes tópicos no Guia do desenvolvedor do AWS Key Management Service:

- [Conceitos básicos](#)
- [Usar políticas de chaves no AWS KMS](#)

Você também pode usar a AWS KMS política de chaves ([PutKeyPolicy](#)) para copiar a política de chaves para as chaves gerenciadas pelo cliente que deseja usar para criptografar as exportações de métricas usando a API REST, a AWS CLI e SDKs.

O que é um manifesto de exportação do S3 Storage Lens?

Dada a grande quantidade de dados agregados, uma exportação de métricas diárias do S3 Storage Lens pode ser dividida em vários arquivos. O arquivo de manifesto `manifest.json` descreve onde as métricas que exportam arquivos para esse dia estão localizadas. Sempre que uma nova exportação é entregue, ela é acompanhada por um novo manifesto. Cada manifesto contido no `manifest.json` fornece metadados e outras informações básicas sobre a exportação.

As informações do manifesto incluem as seguintes propriedades:

- `sourceAccountId` — O ID da conta do proprietário da configuração.
- `configId` — Um identificador exclusivo do painel.
- `destinationBucket` — O bucket de destino do Amazon Resource Name (ARN) no qual a exportação das métricas é colocada.
- `reportVersion` — A versão da exportação.
- `reportDate` — A data do relatório.
- `reportFormat` — O formato do relatório.
- `reportSchema` — O esquema do relatório.
- `reportFiles` — A lista real dos arquivos de relatório de exportação que estão no bucket de destino.

Veja a seguir um exemplo de manifesto em um arquivo `manifest.json` para um inventário em formato CSV.

```
{
    "sourceAccountId": "123456789012",
    "configId": "my-dashboard-configuration-id",
    "destinationBucket": "arn:aws:s3:::destination-bucket",
    "reportVersion": "V_1",
    "reportDate": "2020-11-03",
    "reportFormat": "CSV",

    "reportSchema": "version_number,configuration_id,report_date,aws_account_number,aws_region,storage_clas
    "reportFiles": [
        {
            "key": "DestinationPrefix/StorageLens/123456789012/my-dashboard-configuration-
            id/V_1/reports/dt=2020-11-03/a38f6bc4-2e3d-4355-ac8a-e2fdcf3de158.csv",
    ]}
```

```
        "size":1603959,  
        "md5Checksum":"2177e775870def72b8d84febe1ad3574"  
    ]  
}
```

Veja a seguir um exemplo de manifesto em um arquivo `manifest.json` para um inventário em formato Parquet.

```
{  
    "sourceAccountId":"123456789012",  
    "configId":"my-dashboard-configuration-id",  
    "destinationBucket":"arn:aws:s3:::destination-bucket",  
    "reportVersion":"V_1",  
    "reportDate":"2020-11-03",  
    "reportFormat":"Parquet",  
    "reportSchema":"message s3.storage.lens { required string version_number; required  
string configuration_id; required string report_date; required string aws_account_number;  
required string aws_region; required string storage_class; required string record_type;  
required string record_value; required string bucket_name; required string metric_name;  
required long metric_value; }",  
    "reportFiles": [  
        {  
            "key":"DestinationPrefix/StorageLens/123456789012/my-dashboard-configuration-id/  
V_1/reports/dt=2020-11-03/bd23de7c-b46a-4cf4-bcc5-b21aac5be0f5.par",  
            "size":14714,  
            "md5Checksum":"b5c741ee0251cd99b90b3e8eff50b944"  
        }  
    ]  
}
```

Você pode configurar a exportação de métricas para serem geradas como parte da configuração do painel no console do Amazon S3 ou usando a API REST do Amazon S3, a AWS CLI e os SDKs.

Compreendendo o esquema de exportação do Amazon S3 Storage Lens

A tabela a seguir contém o esquema da exportação de métricas do S3 Storage Lens.

Nome do atributo	Tipo de dados	Nome da coluna	Descrição
VersionNumber	String	version_number	A versão das métricas do S3 Storage Lens que está sendo usada.
ConfigurationId	String	configuration_id	O nome da configuração configuration_id do seu S3 Storage Lens.
ReportDate	String	report_date	A data em que as métricas foram rastreadas.
AwsAccountNumber	String	aws_account_number	Número da sua Conta da AWS.
AwsRegion	String	aws_region	A Região da AWS para a qual as métricas estão sendo rastreadas.

Nome do atributo	Tipo de dados	Nome da coluna	Descrição
StorageClass	String	storage_class	A classe de armazenamento do bucket em questão.
RecordType	ENUM	record_type	O tipo de artefato que está sendo relatado (CONTA, BUCKET ou PREFIXO).
RecordValue	String	record_value	O valor do registro. Este campo é preenchido quando o record_type é PREFIXO. Note O valor do registro é codificado em URL
BucketName	String	bucket_name	O nome do intervalo que está sendo relatado.
MetricName	String	metric_name	O nome da métrica que está sendo relatada.
MetricValue	Longo	metric_value	O valor da métrica que está sendo relatada.

Exemplo de exportação de métricas do S3 Storage Lens

Veja a seguir um exemplo de uma exportação de métricas do S3 Storage Lens com base nesse esquema.

version_r	configuration_id	report_date	aws_account_number	aws_region	storage_class	record_type	record_value	bucket_name	metric_name
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			StorageBytes
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ObjectCount
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ReplicatedObjects
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ReplicatedObjects
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			EncryptedObjects
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			EncryptedObjects
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			DeleteMarker
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ObjectLockEnabled
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ObjectLockEnabled
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			CurrentVersion
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			CurrentVersion
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			NonCurrent
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			NonCurrent
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			IncompleteMultipartUploads
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			IncompleteMultipartUploads
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc	StorageBytes	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc	ObjectCount	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc	ReplicatedObjects	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc	ReplicatedObjects	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc	EncryptedObjects	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc	EncryptedObjects	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc	DeleteMarker	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc	ObjectLockEnabled	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc	ObjectLockEnabled	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc	CurrentVersion	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc	CurrentVersion	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc	NonCurrent	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc	NonCurrent	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc	IncompleteMultipartUploads	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc	IncompleteMultipartUploads	AWSLogs%2F546264889236%2FCloudTrail%2Fus%3Acloudtrail-log-sfc

Monitorar métricas do S3 Storage Lens no CloudWatch

É possível publicar métricas de uso e atividade do S3 Storage Lens no Amazon CloudWatch para criar uma visão unificada de sua integridade operacional nos painéis do CloudWatch. Também é possível usar os recursos do CloudWatch, como alarmes e ações acionadas, matemática métrica e detecção de anomalias, para monitorar e realizar ações sobre as métricas do S3 Storage Lens. Além disso, as APIs do CloudWatch permitem que aplicações, inclusive provedores de terceiros, acessem as métricas do S3 Storage Lens. Para obter mais informações sobre os recursos do CloudWatch, consulte o [Guia do usuário do Amazon CloudWatch Events](#).

Você pode habilitar a opção de publicação no CloudWatch para configurações de painel novas ou existentes usando o console do Amazon S3, a API REST do Amazon S3, a AWS CLI e os AWS SDKs. Painéis atualizados para métricas e recomendações avançadas do S3 Storage Lens podem usar a opção de publicação no CloudWatch. Para saber mais sobre o preço das métricas e recomendações avançadas do S3 Storage Lens, consulte [Preços do Amazon S3](#). Não são aplicadas cobranças adicionais para publicação de métricas no CloudWatch, mas outras cobranças do CloudWatch, como painéis, alarmes e API, são aplicáveis. Para obter mais informações, consulte [Amazon CloudWatch Pricing](#) (Definição de preços do Amazon CloudWatch).

As métricas do S3 Storage Lens são publicadas no CloudWatch na conta proprietária da configuração do S3 Storage Lens. Depois de habilitar a opção de publicação no CloudWatch em métricas e recomendações avançadas, você pode acessar as métricas de atividade e de uso em nível de organização, conta e bucket no CloudWatch. As métricas de nível prefixo não estão disponíveis no CloudWatch.

Note

As métricas do S3 Storage Lens são métricas diárias e são publicadas no CloudWatch uma vez por dia. Quando você consulta métricas do S3 Storage Lens no CloudWatch, o período para a consulta deve ser de 1 dia (86.400 segundos). Depois que as métricas diárias do S3 Storage Lens são exibidas no painel do S3 Storage Lens no console do Amazon S3, essas mesmas métricas podem demorar algumas horas para serem exibidas no CloudWatch. Quando você habilita a opção de publicação do CloudWatch para métricas do S3 Storage Lens pela primeira vez, as métricas podem levar até 24 horas para serem publicadas no CloudWatch.

Depois de habilitar a opção de publicação do CloudWatch, você pode usar os seguintes recursos do CloudWatch para monitorar e analisar os dados do S3 Storage Lens:

- [Painéis \(p. 1148\)](#): use os painéis do CloudWatch para criar painéis personalizados do S3 Storage Lens. Compartilhe seu painel do CloudWatch com pessoas que não têm acesso direto à sua Conta da AWS, entre equipes, com partes interessadas e com pessoas externas às suas organizações.
- [Alarmes e ações acionadas \(p. 1148\)](#): configure alarmes que monitoram métricas e realizam ações quando um limite é violado. Por exemplo, é possível configurar um alarme que envia uma notificação do Amazon SNS quando os bytes de carregamentos fracionados incompletos excederem 1 GB por três dias consecutivos.
- [Detecção de anomalias \(p. 1148\)](#): habilite a detecção de anomalias para analisar continuamente métricas, determinar linhas de base normais e anomalias. Você pode criar um alarme de detecção de anomalias com base no valor esperado de uma métrica. Por exemplo, é possível monitorar anomalias para bytes habilitados para bloqueio de objeto para detectar a remoção não autorizada das configurações do bloqueio.
- [Matemática métrica: \(p. 1148\)](#) você também pode usar a matemática métrica para consultar várias métricas do S3 Storage Lens e usar expressões matemáticas para criar novas séries temporais de acordo com essas métricas. Por exemplo, você pode criar uma nova métrica para obter a média de tamanho do objeto dividindo StorageBytes por ObjectCount.

Para obter mais informações sobre a opção de publicação no CloudWatch para métricas do S3 Storage Lens, consulte os tópicos a seguir.

Tópicos

- [Métricas e dimensões do S3 Storage Lens \(p. 1139\)](#)
- [Habilitar publicação no CloudWatch para o S3 Storage Lens \(p. 1141\)](#)
- [Trabalhar com métricas do S3 Storage Lens no CloudWatch \(p. 1147\)](#)

Métricas e dimensões do S3 Storage Lens

Para enviar métricas do S3 Storage Lens ao CloudWatch, você deve habilitar a opção de publicação no CloudWatch nas métricas e recomendações avançadas do S3 Storage Lens. Depois de habilitada, é possível usar os [painéis](#) do CloudWatch para monitorar as métricas do S3 Storage Lens juntamente com métricas de outras aplicações e criar uma visão unificada de sua integridade operacional. Você pode usar dimensões para filtrar as métricas do S3 Storage Lens no CloudWatch por organização, conta, bucket, classe de armazenamento, região e ID de configuração de métricas.

As métricas do S3 Storage Lens são publicadas no CloudWatch na conta proprietária da configuração do S3 Storage Lens. Depois de habilitar a opção de publicação no CloudWatch em métricas e recomendações avançadas, você pode acessar as métricas de atividade e de uso em nível de organização, conta e bucket no CloudWatch. As métricas de nível prefixo não estão disponíveis no CloudWatch.

Note

As métricas do S3 Storage Lens são métricas diárias e são publicadas no CloudWatch uma vez por dia. Quando você consulta métricas do S3 Storage Lens no CloudWatch, o período para a consulta deve ser de 1 dia (86.400 segundos). Depois que as métricas diárias do S3 Storage Lens são exibidas no painel do S3 Storage Lens no console do Amazon S3, essas mesmas métricas podem demorar algumas horas para serem exibidas no CloudWatch. Quando você habilita a opção de publicação do CloudWatch para métricas do S3 Storage Lens pela primeira vez, as métricas podem levar até 24 horas para serem publicadas no CloudWatch.

Para obter mais informações sobre métricas e dimensões do S3 Storage Lens no CloudWatch, consulte os tópicos a seguir.

Tópicos

- [Métricas \(p. 1140\)](#)
- [Dimensões \(p. 1140\)](#)

Métricas

As métricas de uso e atividade do S3 Storage Lens estão disponíveis como métricas no CloudWatch. As métricas do S3 Storage Lens são publicadas no namespace AWS/S3/storage-Lens do CloudWatch. Este namespace serve apenas para métricas do S3 Storage Lens. As métricas de bucket, solicitação e replicação do Amazon S3 são publicadas no namespace AWS/S3.

As métricas do S3 Storage Lens são publicadas no CloudWatch na conta proprietária da configuração do S3 Storage Lens. Depois de habilitar a opção de publicação no CloudWatch em métricas e recomendações avançadas, você pode acessar as métricas de atividade e de uso em nível de organização, conta e bucket no CloudWatch. As métricas de nível prefixo não estão disponíveis no CloudWatch.

No S3 Storage Lens, as métricas são agregadas e armazenadas somente na região inicial designada. As métricas do S3 Storage Lens também são publicadas no CloudWatch na região inicial especificada na configuração do S3 Storage Lens.

Para obter uma lista completa das métricas do S3 Storage Lens, inclusive uma lista dessas métricas disponíveis no CloudWatch, consulte [Glossário de métricas de lente de armazenamento do Amazon S3 \(p. 1152\)](#).

Note

A estatística válida para métricas do S3 Storage Lens no CloudWatch é Average. Para obter mais informações sobre estatísticas no CloudWatch, consulte [Definições de estatísticas do CloudWatch](#) no Guia do usuário do Amazon CloudWatch.

Detalhamento das métricas do S3 Storage Lens no CloudWatch

O S3 Storage Lens oferece métricas com detalhamento de organização, conta, bucket e prefixo. O S3 Storage Lens publica métricas do S3 Storage Lens em nível de organização, conta e bucket no CloudWatch. As métricas de nível prefixo do S3 Storage não estão disponíveis no CloudWatch.

Para obter mais informações sobre o detalhamento das métricas do S3 Storage Lens disponíveis no CloudWatch, consulte a lista a seguir:

- Organização: métricas agregadas entre as contas dos membros de sua organização. O S3 Storage Lens publica métricas de contas de membros no CloudWatch na conta de gerenciamento.
- Organização e conta: métricas de contas de membros de sua organização.
- Organização e bucket: métricas de buckets do Amazon S3 de contas de membros de sua organização.
- Conta (nível não organizacional): métricas agregadas entre os buckets de sua conta.
- Bucket (nível não organizacional): métricas de um bucket específico. No CloudWatch, o S3 Storage Lens publica essas métricas na Conta da AWS que criou a configuração do S3 Storage Lens. O S3 Storage Lens publica essas métricas somente de configurações não organizacionais.

Dimensões

Ao enviar dados ao CloudWatch, o S3 Storage Lens anexa dimensões a cada métrica. As dimensões são categorias que descrevem as características das métricas. Você pode usar dimensões para filtrar os resultados que o CloudWatch retorna.

Por exemplo, todas as métricas do S3 Storage Lens no CloudWatch têm a dimensão `configuration_id`. É possível usar essa dimensão para diferenciar as métricas associadas a uma

configuração específica do S3 Storage Lens. O `organization_id` identifica métricas no nível da organização. Para obter mais informações sobre dimensões no CloudWatch, consulte [Dimensões](#) no Guia do usuário do CloudWatch.

Há diferentes dimensões disponíveis para métricas do S3 Storage Lens, conforme o detalhamento das métricas. Por exemplo, você pode usar a dimensão `organization_id` para filtrar métricas no nível da organização pelo ID do AWS Organizations. No entanto, não é possível usar essa dimensão para métricas no nível da conta e do bucket. Para obter mais informações, consulte [. Filtrar métricas usando dimensões \(p. 1148\)](#).

Para ver quais dimensões estão disponíveis para sua configuração do S3 Storage Lens, consulte a tabela a seguir.

Dimensão	Descrição	Bucket	Organização	Organização e e					
<code>configuration_id</code>	Nome do painel para configuração do S3 Storage Lens relatado nas métricas	✓	✓	✓	✓	✓	✓	✓	✓
<code>metrics_version</code>	Versão das métricas do S3 Storage Lens. A versão de métricas tem um valor fixo de 1.0.	✓	✓	✓	✓	✓	✓	✓	✓
<code>organization_id</code>	AWS OrganizationsID do das métricas	✗	✗	✓	✓	✓	✓	✓	✓
<code>aws_account_number</code>	Conta da AWS associada às métricas	✓	✓	✗	✓	✓	✓	✓	✓
<code>aws_region</code>	Região da AWS das métricas	✓	✓	✓	✓	✓	✓	✓	✓
<code>bucket_name</code>	Nome do bucket do S3 relatado nas métricas	✓	✗	✗	✓	✓	✓	✓	✗
<code>storage_class</code>	Classe de armazenamento do bucket relatado nas métricas	✓	✓	✓	✓	✓	✓	✓	✓
<code>record_type</code>	Detalhamento das métricas: ORGANIZATION, ACCOUNT, BUCKET	✓	✓	✓	✓	✓	✓	✓	✓
		BUCKET ACCOUNT ORGANIZATION							

Habilitar publicação no CloudWatch para o S3 Storage Lens

É possível publicar métricas de uso e atividade do S3 Storage Lens no Amazon CloudWatch para criar uma visão unificada de sua integridade operacional nos [painéis](#) do CloudWatch. Também é possível usar os recursos do CloudWatch, como alarmes e ações acionadas, matemática métrica e detecção de anomalias, para monitorar e realizar ações sobre as métricas do S3 Storage Lens. Além disso, as APIs do CloudWatch permitem que aplicações, inclusive provedores de terceiros, acessem as métricas do S3 Storage Lens. Para obter mais informações sobre os recursos do CloudWatch, consulte o [Guia do usuário do Amazon CloudWatch Events](#).

As métricas do S3 Storage Lens são publicadas no CloudWatch na conta proprietária da configuração do S3 Storage Lens. Depois de habilitar a opção de publicação no CloudWatch em métricas e recomendações avançadas, você pode acessar as métricas de atividade e de uso em nível de organização, conta e bucket no CloudWatch. As métricas de nível prefixo não estão disponíveis no CloudWatch.

Você pode habilitar o suporte do CloudWatch a configurações de painel novas ou existentes usando o console do S3, as APIs REST do Amazon S3, a AWS CLI e os AWS SDKs. A opção de publicação no CloudWatch está disponível para painéis atualizados para métricas e recomendações avançadas do S3 Storage Lens. Para saber mais sobre o preço das métricas e recomendações avançadas do S3 Storage Lens, consulte [Preços do Amazon S3](#). Não são aplicadas cobranças adicionais para publicação

de métricas no CloudWatch, mas outras cobranças do CloudWatch, como painéis, alarmes e API, são aplicáveis.

Para habilitar a opção de publicação de métricas do S3 Storage Lens no CloudWatch, consulte os tópicos a seguir.

Note

As métricas do S3 Storage Lens são métricas diárias e são publicadas no CloudWatch uma vez por dia. Quando você consulta métricas do S3 Storage Lens no CloudWatch, o período para a consulta deve ser de 1 dia (86.400 segundos). Depois que as métricas diárias do S3 Storage Lens são exibidas no painel do S3 Storage Lens no console do Amazon S3, essas mesmas métricas podem demorar algumas horas para serem exibidas no CloudWatch. Quando você habilita a opção de publicação do CloudWatch para métricas do S3 Storage Lens pela primeira vez, as métricas podem levar até 24 horas para serem publicadas no CloudWatch.

Atualmente, as métricas do S3 Storage Lens não podem ser consumidas por fluxos do CloudWatch.

Uso do console do S3

Ao atualizar um painel do S3 Storage Lens, não é possível alterar o nome do painel nem a região inicial. Não é possível alterar o escopo do painel padrão, que tem escopo para o armazenamento de toda a sua conta.

Para atualizar o painel do S3 Storage Lens para habilitar a publicação no CloudWatch

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha S3 Storage Lens.
3. Escolha o painel que você deseja editar e escolha Edit (Editar).
4. Em Metrics selection (Seleção de métricas), escolha Advanced metrics and recommendations (Métricas e recomendações avançadas).

As métricas e recomendações avançadas estão disponíveis mediante cobranças adicionais. As métricas e recomendações avançadas incluem um período de 15 meses para consultas de dados, métricas de uso agregadas no nível de prefixo, métricas de atividade agregadas por bucket, opção de publicação no CloudWatch e recomendações contextuais que ajudam a otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados. Para obter mais informações, consulte [Definição de preço do Amazon S3](#).

5. Em Select Advanced metrics and recommendations features, (Selecionar recursos de métricas e recomendações avançadas), escolha CloudWatch publishing (Publicação no CloudWatch).

Important

Se a configuração permitir a agregação de prefixos para métricas de uso, as métricas de nível de prefixo não serão publicadas no CloudWatch. Somente métricas em nível de organização, conta e bucket do S3 Storage Lens são publicadas no CloudWatch.

6. Escolha Create dashboard (Criar painel).

Para criar um novo painel do S3 Storage Lens que habilite o suporte ao CloudWatch

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Storage Lens.
3. Escolha Create dashboard (Criar painel).
4. Em General (Geral), defina as opções de configuração:

- a. Em Dashboard name (Nome do painel), insira o nome do painel.

Os nomes do painel devem ter menos de 65 caracteres e não devem conter caracteres ou espaços especiais. Não é possível alterar o nome do painel depois que o painel é criado.

- b. Escolha a Home Region (Região inicial) do painel.

As métricas de todas as regiões incluídas neste escopo do painel são armazenadas centralmente nesta região inicial designada. No CloudWatch, as métricas do S3 Storage Lens também estão disponíveis na região inicial. Não é possível alterar a região inicial depois que o painel é criado.

5. (Opcional) Para adicionar etiquetas, escolha Add tag (Adicionar etiqueta) e insira a Key (Chave) e o Value (Valor) da etiqueta.

Note

Você pode adicionar até 50 tags à configuração do painel.

6. Defina o escopo de sua configuração:

- a. Se você estiver criando uma configuração no nível da organização, escolha as contas a serem incluídas na configuração: **Include all accounts in your configuration** (Incluir todas as contas na configuração) ou **Limit the scope to your signed-in account** (Limitar o escopo a sua conta conectada).

Note

Ao criar uma configuração no nível da organização que inclui todas as contas, você só poderá incluir ou excluir regiões, e não buckets.

- b. Escolha as regiões e os buckets que deseja que o S3 Storage Lens inclua na configuração do painel:

- Para incluir todas as regiões, selecione **Include Regions and buckets** (Incluir regiões e buckets).
- Para incluir regiões específicas, desmarque a opção **Include all Regions** (Incluir todas as regiões). Em **Choose Regions to include** (Escolher regiões a serem incluídas), escolha as regiões que deseja que o S3 Storage Lens inclua no painel.
- Para incluir bucket específicos, desmarque a opção **Include all buckets** (Incluir todos os buckets). Em **Choose buckets to include** (Escolher buckets a serem incluídos), escolha os buckets que deseja que o S3 Storage Lens inclua no painel.

Note

Você pode escolher até 50 buckets.

7. Em Metrics selection (Seleção de métricas), escolha Advanced metrics and recommendations (Métricas e recomendações avançadas).

Para obter mais informações sobre o preço das métricas e recomendações avançadas, consulte [Preços do Amazon S3](#).

8. Em Select advanced metrics and recommendations features (Selecionar recursos de métricas e recomendações avançadas), selecione as opções que se aplicam:

- Métricas de atividade
- Publicação no CloudWatch

Important

Se você habilitar a agregação de prefixos na configuração do S3 Storage Lens, as métricas de nível de prefixo não serão publicadas no CloudWatch. Somente métricas em nível de organização, conta e bucket do S3 Storage Lens são publicadas no CloudWatch

- Agregação de prefixo

9. (Opcional) Configure a exportação de métricas.

Para obter mais informações sobre como configurar a exportação de métricas, consulte [Criação de um painel do Amazon S3 Storage Lens \(p. 1161\)](#).

10. Escolha Create dashboard (Criar painel).

Uso do AWS CLI

O seguinte exemplo da AWS CLI habilita a opção de publicação no CloudWatch usando uma configuração de métricas e recomendações avançadas em nível de organização do S3 Storage Lens.

```
aws s3control put-storage-lens-configuration --account-id=EXAMPLE-AWS-ACCOUNT-ID --config-id=your-configuration-id --region=us-east-1 --storage-lens-configuration=file://./config.json

config.json
{
    "Id": "SampleS3StorageLensConfiguration", //Use this property to identify S3 S3 Storage Lens configuration.
    "AwsOrg": { //Use this property when enabling S3 S3 Storage Lens for AWS Organizations
        "Arn": "arn:aws:organizations::222222222222:organization/o-abcdefg"
    },
    "AccountLevel": {
        "ActivityMetrics": {
            "IsEnabled": true
        },
        "BucketLevel": {
            "ActivityMetrics": { //Enables Activity Metrics
                "IsEnabled": true
            },
            "PrefixLevel": { //Enables Prefix Level Metrics
                "StorageMetrics": {
                    "IsEnabled": true,
                    "SelectionCriteria": {
                        "MaxDepth": 5,
                        "MinStorageBytesPercentage": 1.25,
                        "Delimiter": "/"
                    }
                }
            }
        }
    },
    "Exclude": { //Replace with include if you prefer to include Regions.
        "Regions": [
            "eu-west-1"
        ],
        "Buckets": [ //This attribute is not supported for organization-level configurations.
            "arn:aws:s3:::source_bucket1"
        ]
    },
    "IsEnabled": true, //Whether the configuration is enabled
    "DataExport": {
        "CloudWatchMetrics": { //Enables publishing Lens metrics to CloudWatch
            "IsEnabled": true
        },
        "S3BucketDestination": { //Enable export of metrics report to na S3 bucket
            "OutputSchemaVersion": "V_1",
            "Format": "CSV", //You can add "Parquet" if you prefer.
            "AccountId": "ExampleAWSAccountNo8",
            "Arn": "arn:aws:s3:::destination-bucket-name", //The destination bucket for your metrics export must be in the same Region as your S3 S3 Storage Lens configuration.
            "Prefix": "prefix-for-your-export-destination",
        }
    }
}
```

```
        "Encryption": {
            "SSEs3": {}
        }
    }
}
```

Usar o AWS SDK for Java

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.ActivityMetrics;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.CloudWatchMetrics;
import com.amazonaws.services.s3control.model.Format;
import com.amazonaws.services.s3control.model.Include;
import com.amazonaws.services.s3control.model.OutputSchemaVersion;
import com.amazonaws.services.s3control.model.PrefixLevel;
import com.amazonaws.services.s3control.model.PrefixLevelStorageMetrics;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.S3BucketDestination;
import com.amazonaws.services.s3control.model.SSEs3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateAndUpdateDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        String exportAccountId = "Destination Account ID";
        String exportBucketArn = "arn:aws:s3:::destBucketName"; // The destination
        bucket for your metrics export must be in the same Region as your S3 S3 Storage Lens
        configuration.
        String awsOrgARN = "arn:aws:organizations::222222222222:organization/o-abcdefg";
        Format exportFormat = Format.CSV;

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                .withDelimiter("/")
                .withMaxDepth(5)
                .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
            PrefixLevelStorageMetrics()
                .withEnabled(true)
                .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                .withActivityMetrics(new ActivityMetrics().withEnabled(true))
        }
    }
}
```

```
.withPrefixLevel(new
PrefixLevel().withStorageMetrics(prefixStorageMetrics));
AccountLevel accountLevel = new AccountLevel()
.withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
.withBucketLevel(bucketLevel);

Include include = new Include()
.withBuckets(Arrays.asList("arn:aws:s3:::bucketName")
.withRegions(Arrays.asList("us-west-2"));

StorageLensDataExportEncryption exportEncryption = new
StorageLensDataExportEncryption()
.withSSES3(new SSES3());
S3BucketDestination s3BucketDestination = new S3BucketDestination()
.withAccountId(exportAccountId)
.withArn(exportBucketArn)
.withEncryption(exportEncryption)
.withFormat(exportFormat)
.withOutputSchemaVersion(OutputSchemaVersion.V_1)
.withPrefix("Prefix");
CloudWatchMetrics cloudWatchMetrics = new CloudWatchMetrics()
.with.IsEnabled(true);
StorageLensDataExport dataExport = new StorageLensDataExport()
.withCloudWatchMetrics(cloudWatchMetrics)
.withS3BucketDestination(s3BucketDestination);

StorageLensAwsOrg awsOrg = new StorageLensAwsOrg()
.withArn(awsOrgARN);

StorageLensConfiguration configuration = new StorageLensConfiguration()
.withId(configurationId)
.withAccountLevel(accountLevel)
.withInclude(include)
.withDataExport(dataExport)
.withAwsOrg(awsOrg)
.with.IsEnabled(true);

List<StorageLensTag> tags = Arrays.asList(
    new StorageLensTag().withKey("key-1").withValue("value-1"),
    new StorageLensTag().withKey("key-2").withValue("value-2")
);

AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
    .withCredentials(new ProfileCredentialsProvider())
    .withRegion(US_WEST_2)
    .build();

s3ControlClient.putStorageLensConfiguration(new
PutStorageLensConfigurationRequest()
    .withAccountId(sourceAccountId)
    .withConfigId(configurationId)
    .withStorageLensConfiguration(configuration)
    .withTags(tags)
);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Uso da API REST

Para habilitar a opção de publicação no CloudWatch usando a API REST do Amazon S3, use [PutStorageLensConfiguration](#).

Próximas etapas

Depois de habilitar a opção de publicação no CloudWatch, você poderá acessar as métricas do S3 Storage Lens no CloudWatch. Você também pode utilizar os recursos do CloudWatch para monitorar e analisar os dados do S3 Storage Lens no CloudWatch. Para obter mais informações, consulte os tópicos a seguir:

- [Métricas e dimensões do S3 Storage Lens \(p. 1139\)](#)
- [Trabalhar com métricas do S3 Storage Lens no CloudWatch \(p. 1147\)](#)

Trabalhar com métricas do S3 Storage Lens no CloudWatch

É possível publicar métricas de uso e atividade do S3 Storage Lens no Amazon CloudWatch para criar uma visão unificada de sua integridade operacional nos [painéis](#) do CloudWatch. Também é possível usar os recursos do CloudWatch, como alarmes e ações acionadas, matemática métrica e detecção de anomalias, para monitorar e realizar ações sobre as métricas do S3 Storage Lens. Além disso, as APIs do CloudWatch permitem que aplicações, inclusive provedores de terceiros, acessem as métricas do S3 Storage Lens. Para obter mais informações sobre os recursos do CloudWatch, consulte o [Guia do usuário do Amazon CloudWatch Events](#).

Você pode habilitar a opção de publicação no CloudWatch para configurações de painel novas ou existentes usando o console do Amazon S3, as APIs REST do Amazon S3, a AWS CLI e os AWS SDKs. A opção de publicação no CloudWatch está disponível para painéis atualizados para métricas e recomendações avançadas do S3 Storage Lens. Para saber mais sobre o preço das métricas e recomendações avançadas do S3 Storage Lens, consulte [Preços do Amazon S3](#). Não são aplicadas cobranças adicionais para publicação de métricas no CloudWatch, mas outras cobranças do CloudWatch, como painéis, alarmes e API, são aplicáveis. Para obter mais informações, consulte [Amazon CloudWatch Pricing](#) (Definição de preços do Amazon CloudWatch).

As métricas do S3 Storage Lens são publicadas no CloudWatch na conta proprietária da configuração do S3 Storage Lens. Depois de habilitar a opção de publicação no CloudWatch em métricas e recomendações avançadas, você pode acessar as métricas de atividade e de uso em nível de organização, conta e bucket no CloudWatch. As métricas de nível prefixo não estão disponíveis no CloudWatch.

Note

As métricas do S3 Storage Lens são métricas diárias e são publicadas no CloudWatch uma vez por dia. Quando você consulta métricas do S3 Storage Lens no CloudWatch, o período para a consulta deve ser de 1 dia (86.400 segundos). Depois que as métricas diárias do S3 Storage Lens são exibidas no painel do S3 Storage Lens no console do Amazon S3, essas mesmas métricas podem demorar algumas horas para serem exibidas no CloudWatch. Quando você habilita a opção de publicação do CloudWatch para métricas do S3 Storage Lens pela primeira vez, as métricas podem levar até 24 horas para serem publicadas no CloudWatch.

Atualmente, as métricas do S3 Storage Lens não podem ser consumidas por fluxos do CloudWatch.

Para obter mais informações sobre como trabalhar com métricas do S3 Storage Lens no CloudWatch, consulte os tópicos a seguir.

Tópicos

- [Trabalhar com painéis do CloudWatch \(p. 1148\)](#)
- [Definir alarmes, acionar ações e usar detecção de anomalias \(p. 1148\)](#)
- [Filtrar métricas usando dimensões \(p. 1148\)](#)

- [Calcular novas métricas com matemática métrica \(p. 1148\)](#)
- [Usar expressões de pesquisa em grafos \(p. 1149\)](#)

Trabalhar com painéis do CloudWatch

É possível usar os painéis do CloudWatch para monitorar as métricas do S3 Storage Lens juntamente com métricas de outras aplicações e criar uma visão unificada de sua integridade operacional. Os painéis são páginas iniciais personalizáveis no console do CloudWatch que você pode usar para monitorar seus recursos em uma única visualização.

O CloudWatch tem amplo controle de permissões que não oferece suporte ao limite de acesso a um conjunto específico de métricas ou dimensões. Os usuários de sua conta ou organização que têm acesso ao CloudWatch terão acesso a métricas para todas as configurações do S3 Storage Lens em que a opção de suporte do CloudWatch estiver habilitada. Não é possível gerenciar permissões para painéis específicos, como pode ser feito no S3 Storage Lens. Para obter mais informações sobre permissões do CloudWatch, consulte [Gerenciamento de permissões de acesso a seus recursos do CloudWatch](#) no Guia do usuário do Amazon CloudWatch.

Para obter mais informações sobre como usar painéis do CloudWatch e configurar permissões, consulte [Usar painéis do Amazon CloudWatch](#) e [Compartilhar painéis do CloudWatch](#) no Guia do usuário do Amazon CloudWatch.

Definir alarmes, acionar ações e usar detecção de anomalias

Você pode configurar alarmes do CloudWatch que monitoram as métricas do S3 Storage Lens no CloudWatch e realizam ações quando um limite é violado. Por exemplo, é possível configurar um alarme que envia uma notificação do Amazon SNS quando os bytes de carregamentos fracionados incompletos excederem 1 GB por três dias consecutivos.

Também é possível habilitar a detecção de anomalias para analisar continuamente as métricas do S3 Storage Lens, determinar linhas de base normais e apontar anomalias. Você pode criar um alarme de detecção de anomalias com base no valor esperado de uma métrica. Por exemplo, é possível monitorar anomalias para bytes habilitados para bloqueio de objeto para detectar a remoção não autorizada das configurações do bloqueio.

Para obter mais informações e exemplos, consulte [Usar alarmes do Amazon CloudWatch](#) e [Criar um alarme da métrica em um grafo](#) no Guia do usuário do Amazon CloudWatch.

Filtrar métricas usando dimensões

Use dimensões para filtrar métricas do S3 Storage Lens no console do CloudWatch. Por exemplo, é possível filtrar por `configuration_id`, `aws_account_number`, `aws_region`, `bucket_name` e muito mais.

O S3 Storage Lens oferece suporte a várias configurações de painel por conta. Isso significa que configurações diferentes podem incluir o mesmo bucket. Quando essas métricas forem publicadas no CloudWatch, o intervalo terá métricas duplicadas no CloudWatch. Para exibir métricas somente para uma configuração específica do S3 Storage Lens no CloudWatch, você pode usar a dimensão `configuration_id`. Ao filtrar por `configuration_ID`, você visualiza apenas métricas associadas à configuração que você identificou.

Para obter mais informações sobre filtro por ID de configuração, veja o procedimento a seguir, ou consulte [Procurar as métricas disponíveis](#) no Guia do usuário do Amazon CloudWatch.

Calcular novas métricas com matemática métrica

Você pode usar a matemática métrica para consultar várias métricas do S3 Storage Lens e usar expressões matemáticas para criar novas séries temporais de acordo com essas métricas. Por exemplo,

é possível criar uma nova métrica para objetos não criptografados subtraindo os objetos criptografados da contagem de objetos. Também é possível criar uma métrica para obter a média de tamanho do objeto dividindo StorageBytes por ObjectCount ou os bytes numéricos acessados em um dia dividindo BytesDownloaded por StorageBytes.

Para obter mais informações, consulte [Usar matemática métrica](#) no Guia do usuário do Amazon CloudWatch.

[Usar expressões de pesquisa em grafos](#)

Com métricas do S3 Storage Lens, é possível criar uma expressão de pesquisa para todas as métricas chamada IncompleteMultipartUploadStorageBytes e adicionar SUM à expressão. Com essa expressão de pesquisa, você visualiza o total de bytes de MPU incompletos em todas as dimensões do armazenamento em uma única métrica.

Este exemplo mostra a sintaxe que você usaria para criar uma expressão de pesquisa para todas as métricas chamada IncompleteMultipartUploadStorageBytes.

```
SUM(SEARCH('AWS/S3/Storage-
Lens,aws_account_number,aws_region,configuration_id,metrics_version,record_type,storage_class'
MetricName="IncompleteMultipartUploadStorageBytes", 'Average',86400))
```

Para obter mais informações sobre essa sintaxe, consulte [Sintaxe de expressão de pesquisa do CloudWatch](#) no Guia do usuário do Amazon CloudWatch. Para criar um grafo do CloudWatch com uma expressão de pesquisa, consulte [Criar um grafo do CloudWatch com uma expressão de pesquisa](#) no Guia do usuário do Amazon CloudWatch.

[Como usar o Amazon S3 Storage Lens para otimizar seus custos de armazenamento](#)

O Amazon S3 Storage Lens agrupa as suas métricas de uso e atividade e exibe as informações no snapshot da conta na página (inicial) Buckets do console do Amazon S3. Você pode usar o painel do S3 Storage Lens para visualizar insights e tendências, sinalizar discrepâncias e receber recomendações para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados.

Os casos de uso a seguir fornecem estratégias para usar o painel do S3 Storage Lens para otimizar o seu armazenamento de forma mais eficaz.

Tópicos

- [Identifique os seus maiores buckets do S3 \(p. 1149\)](#)
- [Localize uploads fracionados incompletos \(p. 1150\)](#)
- [Reduza o número de versões não atuais retidas \(p. 1151\)](#)
- [Descubra buckets inativos do Amazon S3 \(p. 1151\)](#)

[Identifique os seus maiores buckets do S3](#)

Você paga para armazenar objetos em buckets do S3. A taxa cobrada depende do tamanho dos seus objetos, por quanto tempo você os armazena e quais são as suas classes de armazenamento. Com o Amazon S3 Storage Lens, você adquire uma visualização centralizada de todos os buckets em sua conta. Para ver todos os buckets em todas as contas da sua organização, é possível configurar um painel do S3 Storage Lens no nível do AWS Organizations. Nessa visualização do painel, você pode identificar os seus maiores buckets.

Para identificar os seus maiores buckets

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Storage Lens, Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o painel que deseja visualizar.
4. No canto superior direito, você verá a data mais recente para a qual o S3 Storage Lens coletou métricas de armazenamento. Seu painel sempre carrega a data mais recente para a qual as métricas estão disponíveis.
 - Para ajustar o escopo dos dados do painel que você está visualizando, escolha Filters (Filtros) para aplicar filtros temporários.
 - Para remover todos os filtros, escolha Reset (Redefinir) e, em seguida, Apply (Aplicar).
5. Na guia Overview (Visão geral) do painel, role para baixo até a seção Top N overview for **date** (Visão geral de top N para data) para ver uma classificação dos seus maiores buckets de acordo com a métrica de armazenamento total para um intervalo de datas selecionado.

Você pode alternar a ordem da classificação para exibir os buckets menores e ajustar a métrica para classificar os seus buckets de acordo com qualquer uma das mais de 30 métricas disponíveis. Essa visualização também mostra a mudança no percentual do dia ou da semana anterior, bem como um minigráfico com as suas tendências de 14 dias (ou de 30 dias, se você tiver feito o upgrade para as métricas e recomendações avançadas).

6. Para obter insights mais detalhados sobre os seus buckets, escolha a guia Buckets deste painel. Na guia Buckets você pode ver detalhes como a taxa de crescimento recente, o tamanho médio do objeto, os maiores prefixos e o número de objetos.
7. Para saber sobre os seus maiores buckets, basta navegar até cada bucket dentro do console do S3 para entender os seus objetos e a workload associada ou identificar os proprietários internos do bucket. Ao identificar os proprietários do bucket, você pode descobrir se o crescimento é esperado ou se precisa de monitoramento e controle adicionais.

Localize uploads fracionados incompletos

Você pode usar o recurso de upload fracionado para fazer upload de objetos muito grandes (até 5 TB) como um conjunto de partes para obter uma taxa de transferência aprimorada e uma recuperação mais rápida diante de problemas de rede. Quando o processo de upload fracionado não termina, as partes incompletas permanecem no bucket (em um estado inutilizável) e incorrem em custos de armazenamento até que o processo de upload seja concluído ou as peças incompletas sejam removidas. Para obter mais informações, consulte [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).

Com o S3 Storage Lens, você pode identificar o número de bytes de upload fracionado incompleto em sua conta ou em toda a sua organização.

Para identificar bytes de uploads fracionados incompletos

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Storage Lens, Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o painel que deseja visualizar.
4. Na parte superior da guia Overview (Visão geral), na seção Snapshot, escolha Cost efficiency (Custo-benefício) para ver a métrica % incomplete MPU bytes (% de bytes de MPU incompletos).

Você também pode selecionar Incomplete multipart upload bytes (Bytes de upload fracionado incompleto) como uma métrica para qualquer gráfico no painel do S3 Storage Lens. Em seguida, você pode avaliar

ainda mais o impacto dos bytes do upload fracionado incompleto no seu armazenamento, incluindo a sua contribuição para as tendências gerais de crescimento. Você também pode identificar buckets específicos que estão acumulando uploads fracionados incompletos.

Para gerenciar uploads fracionados incompletos automaticamente, [crie uma política de ciclo de vida para expirar bytes de upload fracionado incompleto](#) no bucket após um determinado número de dias.

Reduza o número de versões não atuais retidas

Quando habilitado, o recurso de versionamento do S3 retém várias versões do mesmo objeto que podem ser usadas para recuperar dados rapidamente se um objeto for acidentalmente excluído ou substituído. O versionamento do S3 pode resultar em custos de armazenamento se um grande número de versões anteriores e não atuais forem acumuladas. Para obter mais informações, consulte [Usando o versionamento em buckets do S3 \(p. 678\)](#).

Para identificar o acúmulo dos seus objetos versionados não atuais

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Storage Lens, Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o painel que deseja visualizar.
4. Na parte superior da guia Overview (Visão geral), na seção Snapshot (Snapshot), escolha Cost efficiency (Eficiência de custos). A métrica para % noncurrent version bytes (% de bytes da versão não atual) representa a proporção do seu total de bytes de armazenamento (dentro do escopo do painel) que é atribuída a versões não atuais para a data selecionada.

Note

Se a sua % noncurrent version bytes (% de bytes da versão não atual) for superior a 10% do seu armazenamento no nível da conta, isso pode ser um indicador de que você está armazenando versões demais.

5. Para identificar buckets específicos que estão acumulando um grande número de versões não atuais, role para baixo até a seção Top N overview for **date** (Visão geral de top N para data) e selecione a métrica % noncurrent version bytes (% de bytes de versão não atual).

Depois de determinar quais buckets exigem investigação adicional, você pode navegar até os buckets no console do S3 e habilitar uma política de ciclo de vida para expirar versões não atuais após um número especificado de dias. Se preferir, para reduzir custos mantendo as versões desatualizadas, você pode configurar uma política de ciclo de vida para fazer a transição de versões desatualizadas para o Amazon S3 Glacier Flexible Retrieval. Para obter mais informações, consulte [Exemplo 6: Especificar uma regra de ciclo de vida para um bucket com versionamento habilitado \(p. 782\)](#).

Descubra buckets inativos do Amazon S3

Se você tiver [métricas avançadas do S3 Storage Lens](#) habilitadas, poderá usar [métricas de atividade](#) para entender o quanto inativos seus buckets do S3 estão. Um bucket “inativo” é aquele cujo armazenamento deixou de ter acesso (ou é acessado raramente). Essa falta de atividade normalmente indica que os objetos do bucket não são acessados com frequência.

As métricas de atividade, como solicitações GET e bytes de download, indicam com que frequência os seus buckets são acessados todos os dias. Para entender a consistência do padrão de acesso e identificar buckets que deixaram de ser acessados, você pode fazer uma tendência desses dados ao longo de vários meses. A métrica de taxa de recuperação, computada como bytes de download/armazenamento total, indica a proporção de armazenamento em um bucket acessado diariamente.

Note

Os bytes de download são duplicados nos casos em que há download do mesmo objeto várias vezes em um dia.

Para ver o quanto ativos os seus buckets estão

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Storage Lens, Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o painel que deseja visualizar.
4. Escolha a guia Bucket (Bucket) do painel e role para baixo até os gráficos de análise de bolhas por buckets para **datas**.
5. Na seção Bubble analysis (Análise em bolhas), você pode traçar seus buckets em várias dimensões usando três métricas quaisquer para representar o eixo x, o eixo y e o tamanho da bolha. Selecione % retrieval rate (% de taxa de recuperação) como uma das métricas.
6. Para encontrar os buckets que se tornaram inativos, faça uma análise em bolhas usando as métricas Total storage (Armazenamento total), % retrieval rate (% de taxa de recuperação) e Average object size (Tamanho médio do objeto). Procure por buckets com taxas zero de recuperação (ou próximas a zero) e um tamanho maior de armazenamento relativo.

Daqui, você pode identificar os proprietários do bucket em sua conta ou organização e descobrir se esse armazenamento ainda é necessário. Em seguida, você pode otimizar os custos configurando as [políticas de validade de ciclo de vida](#) para os buckets ou arquivando os dados no [Amazon S3 Glacier](#).

Para impedir que o problema de buckets inativos prossiga, você pode [fazer a transição dos seus dados de forma automática usando políticas de S3 Lifecycle](#) em seus buckets, ou habilitar o [arquivamento automático com o S3 Intelligent-Tiering](#).

Por outro lado, usando o exemplo anterior, você pode identificar buckets de acesso frequente e ver se eles foram otimizados para atender às suas solicitações de forma mais eficaz, garantindo que a [classe de armazenamento do S3](#) correta seja usada para eles.

Glossário de métricas de lente de armazenamento do Amazon S3

Por padrão, todos os painéis são configurados com métricas gratuitas, que incluem métricas de uso agregadas no nível do bucket, e os dados ficam disponíveis para consultas por 14 dias. Isso significa que você pode visualizar todas as métricas de uso que o S3 Storage Lens agrupa, e suas métricas ficam disponíveis para consulta 14 dias a partir do dia em que os dados foram agregados.

Métricas e recomendações avançadas incluem métricas de uso que podem ser agregadas por prefixo e métricas de atividade. As métricas de atividade podem ser agregadas por bucket. Os dados ficam disponíveis para consultas por 15 meses. Há cobranças adicionais quando você usa o S3 Storage Lens com métricas e recomendações avançadas. Para obter mais informações, consulte [Definição de preço do Amazon S3](#).

Opção de publicação no Amazon CloudWatch

É possível publicar métricas de uso e atividade do S3 Storage Lens no Amazon CloudWatch para criar uma visão unificada de sua integridade operacional nos [painéis](#) do CloudWatch. A publicação no CloudWatch é uma opção que você pode habilitar em Advanced metrics and recommendations (Métricas e recomendações avançadas). As métricas do Storage Lens são publicadas no namespace `AWS/S3/`

Storage-Lens do CloudWatch. Para obter uma lista das dimensões do S3 Storage Lens no CloudWatch, consulte [Dimensões \(p. 1140\)](#). Para obter uma lista das métricas do S3 Storage Lens no CloudWatch, consulte a coluna Nome da métrica no CloudWatch e exportação na tabela a seguir.

Métricas derivadas

As métricas derivadas não estão disponíveis para a exportação de métricas e a opção de publicação no CloudWatch. No entanto, você pode usar a fórmula de métricas mostrada na coluna Fórmula da métrica derivada para computá-las.

Interpretar os vários símbolos de prefixo das unidades métricas do Amazon S3 Storage Lens (K, M, G etc.)

Os múltiplos de unidade métrica do S3 Storage Lens são escritos usando símbolos de prefixo que são representados usando os símbolos do SI (Sistema Internacional de Unidades) que são padronizados pelo BIPM (Bureau International de Pesos e Medidas). Eles também são usados no UCUM (Unified Code for Units of Measure). Para obter mais informações, consulte a [Lista de símbolos de prefixes do SI](#).

Nome da métrica	Nome da métrica no CloudWatch e exportação	Descrição	Gr	Tipo	Categoria	DER	Fórmula da métrica derivada
Armazenamento total	StorageBytes	O armazenamento total	Y	Uso	Resumo	N	
Contagem de objetos	ObjectCount	A contagem total de objetos	Y	Uso	Resumo	N	
Tamanho do objeto médio	NA	O tamanho médio do objeto	Y	Uso	Resumo	Y	$\text{Sum}(StorageBytes) / \text{sum}(ObjectCount)$
Nº de buckets ativos	NA	O número total de buckets em uso ativo com armazenamento > 0 bytes	Y	Uso	Resumo	Y	$\text{DistinctCount}[\text{Bucketname}]$
Número de contas	NA	O número de contas cujo armazenamento está no escopo	Y	Uso	Resumo	Y	$\text{DistinctCount}[\text{AccountID}]$
Bytes de armazenamento de versão	NA	O número de bytes de uma versão atual	Y	Uso	Proteção de dados, eficiência de custos	N	
% bytes da versão atual	NA	A porcentagem de bytes no escopo da versão atual	Y	Uso	Proteção de dados, eficiência de custos	Y	$\text{Sum}(\text{CurrentVersion Bytes}) / \text{sum}(Storage Bytes)$
Contagem de objetos da versão atual	CurrentVersionObjects	A contagem dos objetos de versão atual	Y	Uso	Proteção de dados, eficiência de custos	N	
% objetos da versão atual	NA	A porcentagem de objetos no escopo	Y	Uso	Proteção de dados,	Y	$\text{Soma}(\text{CurrentVersionObjects}) /$

Nome da métrica	Nome da métrica no CloudWatch e exportação	Descrição	Gr	Tipo	Categoria	DER	Fórmula da métrica derivada
		que são de uma versão atual			eficiência de custos		soma (ObjectCount)
Bytes de armazenamento de versão não atual	NonCurrentVersionStorageBytes	Bytes de bytes versionados não atuais	Y	Uso	Proteção de dados, eficiência de custos	N	
% bytes de versão não atual	NA	A porcentagem de bytes no escopo que são da versão não atual	Y	Uso	Proteção de dados, eficiência de custos	Y	$\text{Sum}(\text{NonCurrentVersionStorageBytes}) / \text{Sum}(\text{StorageBytes})$
Contagem de objetos de versão não atual	NonCurrentVersionObjectCount	Objetos contados objetos de versão não atual	Y	Uso	Proteção de dados, eficiência de custos	N	
% objetos de versão não atual	NA	A porcentagem de objetos no escopo que são uma versão não atual	Y	Uso	Proteção de dados, eficiência de custos	Y	$\text{Sum}(\text{NonCurrentVersionObjectCount}) / \text{Sum}(\text{ObjectCount})$
Contagem de objetos do marcador de exclusão	DeleteMarkerObjectCount	Número total de objetos com um marcador de exclusão	Y	Uso	Custo da eficiência	N	
% objetos de marcador de exclusão	NA	A porcentagem de objetos no escopo com um marcador de exclusão	Y	Uso	Custo da eficiência	Y	
Bytes de armazenamento criptografados	EncryptedStorageBytes	Bytes número total de bytes criptografados usando criptografia do lado do servidor do Amazon S3	Y	Uso	Proteção de dados	N	
% de bytes criptografados	NA	A porcentagem do total de bytes no escopo que são criptografados usando criptografia no lado do servidor do Amazon S3	Y	Uso	Proteção de dados	Y	$\text{Sum}(\text{EncryptedStorageBytes}) / \text{Sum}(\text{StorageBytes})$

Nome da métrica	Nome da métrica no CloudWatch e exportação	Descrição	Gr	Tipo	Categoria	DER	Fórmula da métrica derivada
Contagem de objetos criptografados	EncryptedObjectCount	As contagens totais de objetos criptografadas usando criptografia no lado do servidor do Amazon S3	Y	Uso	Proteção de dados	N	
% objetos criptografados	NA	A porcentagem de objetos no escopo que são criptografados usando criptografia do lado do servidor do Amazon S3	Y	Uso	Proteção de dados	Y	$\frac{\text{Sum}(\text{EncryptedStorageBytes})}{\text{Sum}(\text{ObjectCount})}$
Bytes de armazenamento sem criptografia	NA	O número de bytes no escopo que não são criptografados	Y	Uso	Proteção de dados	Y	$\text{Sum}(\text{StorageBytes}) - \text{sum}(\text{EncryptedStorageBytes})$
% de bytes não criptografados	NA	A porcentagem de bytes no escopo que não são criptografados	Y	Uso	Proteção de dados	Y	$\frac{\text{Sum}(\text{UnencryptedStorageBytes})}{\text{Sum}(\text{StorageBytes})}$
Contagem de objetos não criptografados	NA	A contagem dos objetos que não são criptografados	Y	Uso	Proteção de dados	Y	$\text{Sum}(\text{ObjectCounts}) - \text{sum}(\text{EncryptedObjectCounts})$
Objetos não criptografados.	NA	A porcentagem de objetos não criptografados	Y	Uso	Proteção de dados	Y	$\frac{\text{Sum}(\text{UnencryptedStorageBytes})}{\text{Sum}(\text{ObjectCount})}$
Bytes de armazenamento replicados	ReplicatedStorageBytes	Bytes total de bytes no escopo que são replicados	Y	Uso	Proteção de dados	N	
% bytes replicados	NA	A porcentagem do total de bytes no escopo que são replicados	Y	Uso	Proteção de dados	Y	$\frac{\text{Sum}(\text{ReplicatedStorageBytes})}{\text{Sum}(\text{StorageBytes})}$
Contagem de objetos replicados	ReplicatedObjectCount	As contagens de objetos replicados	Y	Uso	Proteção de dados	N	
% objetos replicados	NA	A porcentagem do total de objetos que são replicados	Y	Uso	Proteção de dados	Y	$\frac{\text{Sum}(\text{ReplicatedObjects})}{\text{Sum}(\text{ObjectCount})}$
Bytes de armazenamento habilitados para bloqueio de objetos	ObjectLockEnabledStorageBytes	Total de bytes no escopo que têm o bloqueio de objetos ativado	Y	Uso	Proteção de dados	N	

Nome da métrica	Nome da métrica no CloudWatch e exportação	Descrição	Gr	Tipo	Categoria	DER	Fórmula da métrica derivada
% bytes de bloqueio de objetos	NA	A porcentagem do total de bytes no escopo que têm o bloqueio de objetos ativado	Y	Uso	Proteção de dados	Y	$\text{Sum(ObjectLockBytes)} / \text{Sum(StorageBytes)}$
Contagem de objetos com bloqueio de objetos ativado	ObjectLockEnabled	ObjectCount de objetos no escopo que têm bloqueio de objetos ativado	Y	Uso	Proteção de dados	N	
% objetos com bloqueio de objetos	NA	A porcentagem de objetos no escopo que têm bloqueio de objetos ativado	Y	Uso	Proteção de dados	Y	$\text{Sum(ObjectLockObjects)} / \text{Sum(ObjectCount)}$
Bytes de armazenamento com multipart upload incompletos	IncompleteMultipartUploadStorageBytes	UploadStorageBytes no escopo com multipart uploads incompletos	Y	Uso	Custo da eficiência	N	
% bytes MPU incompletos	NA	A porcentagem de bytes no escopo que são resultados de multipart uploads incompletos	Y	Uso	Custo da eficiência	Y	$\text{Sum(IncompleteMPUbytes)} / \text{Sum(StorageBytes)}$
Contagem de objetos de multipart upload incompleto	IncompleteMultipartUploadObjectCount	UploadObjectCount de objetos no escopo com multipart uploads incompletos	Y	Uso	Custo da eficiência	N	
% objetos MPU incompletos	NA	A porcentagem de objetos no escopo com multipart uploads incompletos	Y	Uso	Custo da eficiência	Y	$\text{Sum(IncompleteMPUobjects)} / \text{Sum(ObjectCount)}$
Todas as solicitações	AllRequests	O número total de solicitações feitas	N	Atividade	Resumo, atividade	N	
Solicitações GET	GetRequests	O número total de solicitações GET feitas	N	Atividade	Atividades	N	
Solicitações PUT	PutRequests	O número total de solicitações PUT feitas	N	Atividade	Atividades	N	

Nome da métrica	Nome da métrica no CloudWatch e exportação	Descrição	Gr	Tipo	Categoria	DER	Fórmula da métrica derivada
Solicitações de cabeçalho	HeadRequests	O número total de solicitações de cabeçalho feitas	N	Atividades	Atividades	N	
Solicitações de exclusão	DeleteRequests	O número total de solicitações de exclusão feitas	N	Atividades	Atividades	N	
Solicitações de LIST	ListRequests	O número total de solicitações de LIST feitas	N	Atividades	Atividades	N	
Solicitações de Post	PostRequests	O número total de solicitações de post feitas	N	Atividades	Atividades	N	
Solicitações de SELECT	SelectRequests	O número total de solicitações de SELECT	N	Atividades	Atividades	N	
Bytes verificados selecionados	SelectBytesScanned	O número de bytes verificados selecionados	N	Atividades	Atividades	N	
Bytes selecionados retornados	SelectBytesReturned	O número de bytes selecionados retornados	N	Atividades	Atividades	N	
Bytes obtidos por download	BytesDownloaded	O número de bytes no escopo que foram obtidos por download	N	Atividades	Atividades	N	
% taxa de recuperação	NA	A porcentagem da taxa de recuperação	N	Atividade, eficiência de custo	Atividade, eficiência de custo	Y	$\text{Sum}(\text{BytesDownloaded}) / \text{Sum}(\text{StorageBytes})$
Bytes enviados por upload	BytesUploaded	O número de bytes enviados por upload	N	Atividades	Atividades	N	
% relação de consumo	NA	O número de bytes carregados como uma porcentagem do total de bytes de armazenamento no escopo	N	Atividade, eficiência de custo	Atividade, eficiência de custo	Y	$\text{Sum}(\text{BytesUploaded}) / \text{Sum}(\text{Storage Bytes})$
Erros 4xx	4xxErros	O total de erros 4xx no escopo	N	Atividades	Atividades	N	
Erros 5xx	5xxErros	O total de erros 5xx no escopo	N	Atividades	Atividades	N	

Nome da métrica	Nome da métrica no CloudWatch e exportação	Descrição	Gr	Tipo	Categoria	DER	Fórmula da métrica derivada
Total de erros	NA	A soma de todos os erros (4xx) e (5xx)	N	Atividade	Atividades	Y	$\text{Sum}(4xxErrors) + \text{Sum}(5xxErrors)$
Taxa de erro 504	NA	O total de erros como uma porcentagem do total de solicitações	N	Atividade	Atividades	Y	$\text{Sum}(TotalErrors) / \text{Sum}(TotalRequests)$

Trabalhar com o Amazon S3 Storage Lens usando o console e a API

O Amazon S3 Storage Lens agrupa suas métricas de uso e atividade e exibe as informações no snapshot da conta na página inicial do console (Buckets) do Amazon S3, em painéis interativos ou por meio de uma exportação de métricas que você pode obter por download no formato CSV ou Parquet. Você pode usar o painel para visualizar insights e tendências, sinalizar discrepâncias e receber recomendações para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados. Você pode usar o S3 Storage Lens por meio do AWS Management Console, da AWS CLI, de AWS SDKs ou da API REST.

Esta seção contém exemplos de criação, atualização e exibição de configurações do S3 Storage Lens e execução de operações relacionadas ao recurso. Se você estiver usando o S3 Storage Lens com o AWS Organizations, esses exemplos também abordam esses casos de uso. Nos exemplos, substitua todos os valores de variável por valores adequados às suas necessidades.

Tópicos

- [Usar o Amazon S3 Storage Lens no console \(p. 1158\)](#)
- [Exemplos de Amazon S3 Storage Lens usando a AWS CLI \(p. 1170\)](#)
- [Exemplos do Amazon S3 Storage Lens usando o SDK for Java \(p. 1175\)](#)

Usar o Amazon S3 Storage Lens no console

O Amazon S3 Storage Lens agrupa suas métricas de uso e atividade e exibe as informações no snapshot da conta na página inicial do console (Buckets) do Amazon S3, em painéis interativos ou por meio de uma exportação de métricas que você pode obter por download no formato CSV ou Parquet. Você pode usar o painel para visualizar insights e tendências, sinalizar discrepâncias e receber recomendações para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados. Você pode usar o S3 Storage Lens por meio do AWS Management Console, da AWS CLI, de AWS SDKs ou da API REST.

Tópicos

- [Exibição de um painel do Amazon S3 Storage Lens \(p. 1159\)](#)
- [Criação e atualização dos painéis do Amazon S3 Storage Lens \(p. 1161\)](#)
- [Desabilitar ou excluir painéis do Amazon S3 Storage Lens \(p. 1166\)](#)
- [Trabalhar com AWS Organizations para criar painéis no nível da organização \(p. 1167\)](#)

Exibição de um painel do Amazon S3 Storage Lens

O Amazon S3 Storage Lens agrupa suas métricas de uso e atividade e exibe as informações no snapshot da conta na página inicial do console (Buckets) do Amazon S3, em painéis interativos ou por meio de uma exportação de métricas que você pode obter por download no formato CSV ou Parquet. Você pode usar o painel para visualizar insights e tendências, sinalizar discrepâncias e receber recomendações para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados. Você pode usar o S3 Storage Lens por meio do AWS Management Console, da AWS CLI, de AWS SDKs ou da API REST.

O painel padrão do Amazon S3 Storage Lens é o painel de conta padrão. Esse painel é pré-configurado pelo Amazon S3 para ajudá-lo a visualizar insights resumidos e tendências para o uso agregado de armazenamento e as métricas de atividade de toda a sua conta no console. Você não pode modificar seu escopo de configuração, mas pode atualizar a seleção de métricas das Métricas gratuitas para as Métricas e recomendações avançadas pagas, configurar a exportação de métricas opcionais ou até mesmo desativá-la. O painel padrão não pode ser excluído.

Você também pode criar painéis adicionais do S3 Storage Lens focados em Regiões da AWS específicas, buckets do S3 ou outras contas em suas organizações.

O painel do Amazon S3 fornece um recurso avançado de informações sobre seu escopo de armazenamento que representa mais de 30 métricas. Essas métricas representam tendências e outras informações, incluindo resumo do armazenamento, economia de custos, proteção de dados e atividade.

O painel sempre é carregado para a data mais recente para a qual as métricas estão disponíveis.

Para exibir um painel do S3 Storage Lens

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Storage Lens, Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o painel que deseja visualizar.

No canto superior direito, você deve ver a data mais recente para a qual o S3 Storage Lens coletou métricas de armazenamento. Você também tem acesso a filtros temporários para limitar ainda mais o escopo dos dados do painel que você está exibindo. Há também uma opção de redefinição que você pode usar para remover todos os filtros.

Seu painel sempre carrega a data mais recente para a qual as métricas estão disponíveis.

Note

Você não pode usar as credenciais de usuário root da sua conta para visualizar os painéis do Amazon S3 Storage Lens. Para acessar os painéis do S3 Storage Lens, você deve conceder as permissões do IAM necessárias a um usuário do IAM novo ou existente. Em seguida, faça login com essas credenciais de usuário para acessar os painéis do S3 Storage Lens. Para obter mais informações, consulte [Permissões do Amazon S3 Storage Lens \(p. 1130\)](#).

Noções básicas sobre seu painel do S3 Storage Lens

O painel do S3 Storage Lens consiste em uma guia Overview (Visão geral) principal e até cinco guias adicionais que representam cada nível de agregação:

- Account (Conta) (apenas para painéis no nível da organização)
- Região
- Classe de armazenamento
- Bucket

- Prefix (Prefixo) (somente se inscrito em métricas e recomendações avançadas)

Os dados do seu painel são agregados em três seções diferentes.

Snapshot

A primeira seção é a seção Snapshot , que mostra as métricas que o S3 Storage Lens agregou para a data anterior selecionada. Ela mostra dados agregados para as cinco métricas a seguir do escopo de configuração do painel do S3 Storage Lens:

- Total de bytes de armazenamento
- Contagem total de objetos
- Tamanho médio do objeto
- Accounts (Contas): esse valor é 1, a menos que você esteja usando o AWS Organizations, e seu S3 Storage Lens tenha acesso confiável com uma função vinculada a serviços válida. Para obter mais informações, consulte [Usar funções vinculadas a serviços para o Amazon S3 Storage Lens \(p. 582\)](#).
- Buckets
- Solicitações — Se você optar por usar métricas avançadas e recomendações para este painel.

A seção Metrics (Métricas) da seção Snapshot mostra dados agregados das métricas de uso de armazenamento e atividade agrupadas nas seguintes categorias:

- Resumo
- Eficiência de custos
- Proteção de dados
- Atividades

Você pode exibir as propriedades relevantes para essas métricas, incluindo totais, % tendências de alteração (dia/dia, semana/semana e mês/mês) e recomendações.

Tendências e distribuição

A segunda seção da guia Visão geral é Tendências e distribuição.

As tendências fornecem duas métricas que você pode optar por comparar em um intervalo de datas de sua escolha agregado por um período de sua escolha. Ele ajuda você a ver a relação entre as duas tendências de métricas em relação ao escopo de armazenamento do painel. Você pode ver a classe de armazenamento e a distribuição de região entre as duas tendências que você está rastreando.

Com as três maneiras diferentes de comparar métricas, você pode obter mais informações sobre seu armazenamento que podem ajudá-lo a otimizar seu uso ao longo do tempo.

Visão geral da top N

A terceira seção do painel S3 Storage Lens é a Top N overview (visão geral do Top N) (classificada em ordem crescente ou decrescente). Isso permite que você veja suas métricas selecionadas nas principais N contas (se você habilitou o S3 Storage Lens para trabalhar com o AWS Organizations).

As guias Nível de dimensão fornecem uma exibição detalhada de todos os valores dentro de uma determinada dimensão. Por exemplo, a guia Region (Região) mostra métricas para todas as Regiões da AWS, e a guia Bucket mostra métricas para todos os buckets. Cada guia de dimensão contém um layout idêntico composto por quatro seções:

- Um gráfico de tendências exibindo seus N itens principais dentro da dimensão nos últimos 30 dias para a métrica selecionada. Por padrão, esse gráfico exibe os 10 principais itens, mas você pode aumentá-lo para qualquer número desejado.

- Um gráfico de histograma mostra um gráfico de barras verticais para a data e a métrica selecionadas. Talvez seja necessário rolar horizontalmente se tiver um número grande de itens a serem exibidos neste gráfico.
- O gráfico de análise de bolhas representa todos os itens dentro da dimensão representando a primeira métrica no eixo x, uma segunda métrica no eixo y e uma terceira métrica representada pelo tamanho da bolha.
- A exibição de grade de métrica contém cada item na dimensão listada em linhas. As colunas representam cada métrica disponível, organizadas em guias de categoria de métricas para facilitar a navegação.

Note

Para fornecer uma experiência fluida na condução de sua análise, o painel do S3 Storage Lens fornece um menu de ação detalhar, que aparece quando você escolhe qualquer valor de gráfico. Escolha qualquer valor de gráfico para ver os valores de métricas associados e escolha entre duas opções:

- A ação detalhar aplica o valor selecionado como um filtro em todas as guias do painel. Você pode então detalhar esse valor para uma análise mais profunda.
- A ação analisar por leva você à guia de dimensão selecionada no painel e aplica esse valor como um filtro. Em seguida, você pode visualizar esse valor no contexto da nova dimensão para uma análise mais profunda.

As ações detalhar e analisar talvez não apareçam se o resultado produzir resultados ilógicos ou não tiver valor. As ações detalhar e analisar por resultam na aplicação de filtros sobre quaisquer filtros existentes em todas as guias do painel. Se necessário, você pode remover os filtros ou usar a opção de redefinição para remover todos os filtros.

Criação e atualização dos painéis do Amazon S3 Storage Lens

O Amazon S3 Storage Lens agrupa suas métricas de uso e atividade e exibe as informações no snapshot da conta na página inicial do console (Buckets) do Amazon S3, em painéis interativos ou por meio de uma exportação de métricas que você pode obter por download no formato CSV ou Parquet. Você pode usar o painel para visualizar insights e tendências, sinalizar discrepâncias e receber recomendações para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados. Você pode usar o S3 Storage Lens por meio do AWS Management Console, da AWS CLI, de AWS SDKs ou da API REST.

O painel padrão do Amazon S3 Storage Lens é o painel de conta padrão. Esse painel é pré-configurado pelo Amazon S3 para ajudá-lo a visualizar insights resumidos e tendências para o uso agregado de armazenamento e as métricas de atividade de toda a sua conta no console. Você não pode modificar seu escopo de configuração, mas pode atualizar a seleção de métricas das Métricas gratuitas para as Métricas e recomendações avançadas pagas, configurar a exportação de métricas opcionais ou até mesmo desativá-la. O painel padrão não pode ser excluído.

Você também pode criar painéis personalizados adicionais do S3 Storage Lens que podem ser escopo para cobrir o AWS Organizations ou regiões ou buckets específicos em uma conta.

Tópicos

- [Criação de um painel do Amazon S3 Storage Lens \(p. 1161\)](#)
- [Atualização de um painel do Amazon S3 Storage Lens \(p. 1164\)](#)

Criação de um painel do Amazon S3 Storage Lens

Use as etapas a seguir para criar um painel do Amazon S3 Storage Lens no console do Amazon S3.

Para definir o escopo do painel e a seleção de métricas

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha S3 Storage Lens.
3. Escolha Create dashboard (Criar painel).
4. Na página Dashboard (Painel), na seção General (Geral), faça o seguinte:
 - a. Insira um nome de painel.

Os nomes do painel devem ter menos de 65 caracteres e não devem conter caracteres ou espaços especiais.

Note

Não é possível alterar o nome do painel depois que o painel for criado.

- b. Escolha a região inicial do seu painel. As métricas do painel para todas as regiões incluídas neste escopo do painel são armazenadas centralmente nesta região inicial designada.
- c. Opcionalmente, você pode optar por adicionar Tags ao seu painel. Você pode usar tags para gerenciar permissões para seu painel e rastrear os custos para o S3 Storage Lens.

Para obter mais informações, consulte [Controle do acesso usando tags de recurso](#) no Guia do usuário do IAM e [Tags de alocação de custos geradas pela AWS](#) no Guia do usuário do AWS Billing.

Note

Você pode adicionar até 50 tags à configuração do painel.

5. Na seção Dashboard scope (Escopo do painel), faça o seguinte:
 - a. Escolha as regiões e os buckets que deseja que o S3 Storage Lens inclua ou exclua no painel.
 - b. Escolha os buckets nas regiões selecionadas que deseja que o S3 Storage Lens inclua ou exclua. Você pode incluir ou excluir buckets, mas não ambos. Essa opção não está disponível quando você cria painéis no nível da organização.

Note

- Você pode incluir ou excluir regiões e buckets. Essa opção é limitada a regiões somente ao criar painéis no nível da organização em contas de membro na sua organização.
- Você pode escolher até 50 buckets para incluir ou excluir.

6. Na seção Metrics selection (Seleção de métricas), escolha o tipo de métricas que você deseja agregar para este painel.
 - Escolha Free Metrics (Métricas gratuitas) para incluir métricas de uso agregadas no nível de bucket e disponíveis para consultas por 14 dias.
 - Para obter um custo adicional, escolha Métricas e recomendações avançadas. Com métricas e recomendações avançadas, você obtém recomendações contextuais que ajudam a otimizar ainda mais os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados, e os dados ficam disponíveis para consultas por 15 meses. As métricas e recomendações avançadas também incluem métricas de uso agregadas no nível de prefixo, métricas de atividade agregadas por bucket e publicação do Amazon CloudWatch. Para obter mais informações, consulte [Definição de preço do Amazon S3](#).
7. Ao habilitar as métricas e recomendações avançadas, você poderá escolher Advanced metrics and recommendations features (Recursos de métricas e recomendações avançadas):

- a. Escolha Activity metrics (Métricas de atividade) para rastrear solicitações e erros de objetos no escopo do painel.
- b. Escolha CloudWatch publishing (Publicação no CloudWatch) para publicar suas métricas do S3 Storage Lens no CloudWatch.

Para obter mais informações sobre as opções de publicação no CloudWatch, consulte [Monitorar métricas do S3 Storage Lens no CloudWatch \(p. 1138\)](#).

- c. Escolha Prefix aggregation (Agregação de prefixos) para agrregar suas métricas de uso no nível do prefixo e poder receber insights detalhados para seus principais prefixos em cada bucket.

Note

No momento, você só pode receber agregações de prefixo para métricas de uso. As métricas de nível de prefixo não são publicadas no CloudWatch.

8. Caso tenha optado por habilitar a agregação de prefixos, configure o seguinte:

- a. Escolha o tamanho mínimo de limite de prefixo que o S3 Storage Lens coletará para este painel. Por exemplo, um limite de prefixo de 5% indica que os prefixos que compõem 5% ou mais em tamanho do armazenamento do bucket serão agregados.
- b. Escolha o nome do prefixo. Essa configuração indica o número máximo de níveis até os quais os prefixos são avaliados. A profundidade do prefixo deve ser inferior a 10.
- c. Insira um caractere delimitador de prefixo. Este é o valor usado para identificar cada nível de prefixo. O valor padrão no Amazon S3 é o caractere /, mas sua estrutura de armazenamento pode usar outros caracteres delimitadores.

Para exportar métricas para o painel

1. Na seção Metrics Export (Exportação de métricas), escolha Enable (Ativar) para criar uma exportação de métricas que será colocada diariamente em um bucket de destino de sua escolha.

A exportação de métricas está no formato CSV ou Apache Parquet. Ele representa o mesmo escopo de dados que os dados do painel do S3 Storage Lens sem as recomendações.

2. Se ativado, escolha o formato de saída da exportação de métricas diárias. Você pode escolher entre CSV ou Apache Parquet. Parquet é um formato de arquivo de código aberto para Hadoop que armazena dados aninhados em um formato colunar plano.
3. Escolha o bucket do S3 de destino para sua exportação de métricas. Você pode escolher um bucket na conta atual do painel do S3 Storage Lens. Ou você poderá escolher outra Conta da AWS se tiver as permissões do bucket de destino e o ID da conta do proprietário do bucket de destino.
4. Escolha o destino (formato: s3://bucket/prefix) do bucket do S3 de destino. O endereço do bucket deve estar no formato S3 na região inicial do painel de controle do S3 Storage Lens.

Note

- O Amazon S3 atualizará a política de permissões no bucket de destino para permitir que o S3 coloque dados nesse bucket.
 - O console do S3 mostrará a permissão explícita do bucket de destino que será adicionada pelo Amazon S3 à política de bucket de destino na caixa de permissão do bucket de destino.
 - Se o bucket do S3 de destino de exportação de métricas já tiver a criptografia no lado do servidor habilitada, todos os arquivos de exportação colocados lá também deverão ter a criptografia do lado do servidor ativada.
5. Se você optar por ativar a criptografia do lado do servidor para seu painel, você deve escolher um tipo de chave de criptografia. Você pode escolher entre uma [chave do Amazon S3 \(SSE-S3\)](#) e uma [chave do AWS Key Management Service \(AWS KMS\) \(SSE-KMS\)](#).

6. Se você escolheu uma chave do AWS KMS, deverá escolher entre as chaves do KMS ou inserir uma chave do nome do recurso da Amazon (ARN).
7. Escolha Create dashboard (Criar painel).

Atualização de um painel do Amazon S3 Storage Lens

Use as etapas a seguir para atualizar um painel do Amazon S3 Storage Lens no console do Amazon S3.

Para atualizar o escopo do painel e a seleção de métricas

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha S3 Storage Lens.
3. Escolha o painel que deseja editar e escolha Edit (Editar) na parte superior da lista.

Note

Você não pode alterar o seguinte:

- O nome do painel
 - A região inicial
 - O escopo do painel do painel padrão, que tem escopo para o armazenamento de toda a sua conta.
4. Na página de configuração do painel, na seção General (Geral), você pode atualizar e adicionar tags ao seu painel.

Você pode usar tags para gerenciar permissões para seu painel e rastrear os custos para o S3 Storage Lens. Para obter mais informações, consulte [Controle do acesso usando tags de recurso](#) no Guia do usuário do IAM e [Tags de alocação de custos geradas pela AWS](#) no Guia do usuário do AWS Billing.

Note

Você pode adicionar até 50 tags à configuração do painel.

5. Na seção Dashboard scope (Escopo do painel), faça o seguinte:
 - Atualize as regiões e os buckets que deseja que o S3 Storage Lens inclua ou exclua no painel.

Note

- Você pode incluir ou excluir regiões e buckets. Essa opção é limitada a regiões somente ao criar painéis no nível da organização em contas de membro na sua organização.
- Você pode escolher até 50 buckets para incluir ou excluir.

Atualize os buckets nas regiões selecionadas que deseja que o S3 Storage Lens inclua ou exclua. Você pode incluir ou excluir buckets, mas não ambos. Esta opção não está presente ao criar painéis no nível da organização.

6. Na seção Metrics selection (Seleção de métricas), escolha o tipo de métricas que você deseja agregar para este painel.
 - Escolha Free Metrics (Métricas gratuitas) para incluir métricas de uso agregadas no nível de bucket e 14 dias de retenção de dados.
 - Para obter um custo adicional, escolha Métricas e recomendações avançadas. Com métricas e recomendações avançadas, você obtém recomendações contextuais que ajudam a otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados, e os dados

ficam disponíveis para consultas por 15 meses. As métricas e recomendações avançadas também incluem métricas de uso agregadas no nível de prefixo, métricas de atividade agregadas por bucket e publicação do Amazon CloudWatch. Para obter mais informações, consulte [Definição de preço do Amazon S3](#).

7. Ao habilitar as métricas e recomendações avançadas, você poderá escolher Advanced metrics and recommendations features (Recursos de métricas e recomendações avançadas):
 - a. Escolha Activity metrics (Métricas de atividade) para rastrear solicitações e erros de objetos no escopo do painel.
 - b. Escolha CloudWatch publishing (Publicação no CloudWatch) para publicar suas métricas do S3 Storage Lens no CloudWatch.

Para obter mais informações sobre as opções de publicação no CloudWatch, consulte [Monitorar métricas do S3 Storage Lens no CloudWatch \(p. 1138\)](#).

- c. Escolha Prefix aggregation (Agregação de prefixos) para agrregar suas métricas de uso no nível do prefixo e poder receber insights detalhados para seus principais prefixos em cada bucket.

Note

No momento, você só pode receber agregações de prefixo para métricas de uso. As métricas de nível prefixo não são publicadas no CloudWatch

8. Caso tenha optado por habilitar a agregação de prefixos, configure o seguinte:

- a. Escolha o tamanho mínimo de limite de prefixo que o S3 Storage Lens coletará para este painel. Por exemplo, um limite de prefixo de 5% indica que serão agregados os prefixos que compõem 5% ou mais em tamanho do armazenamento do bucket.
- b. Escolha o nome do prefixo. Essa configuração indica o número máximo de níveis até os quais os prefixos são avaliados. A profundidade do prefixo deve ser inferior a 10.
- c. Insira um caractere delimitador de prefixo. Este é o valor usado para identificar cada nível de prefixo. O valor padrão no Amazon S3 é o caractere /, mas sua estrutura de armazenamento pode usar outros caracteres delimitadores.

Para configurar a exportação de métricas

1. Em Metrics Export (Exportação de métricas), escolha Enable (Habilitar) para criar uma exportação de métricas que será colocada diariamente em um bucket de destino de sua preferência. A exportação de métricas está no formato CSV ou Apache Parquet e representa o mesmo escopo de dados que os dados do painel do S3 Storage Lens, sem as recomendações.
2. Se ativado, escolha o formato de saída da exportação de métricas diárias. Você pode escolher entre CSV ou Apache Parquet. Parquet é um formato de arquivo de código aberto para Hadoop que armazena dados aninhados em um formato colunar plano.
3. Atualize o bucket do S3 de destino da exportação de métricas. Você pode escolher um bucket na conta atual do painel do S3 Storage Lens ou outra Conta da AWS se tiver as permissões do bucket de destino e o ID da conta do proprietário do bucket de destino.
4. Atualize o destino (formato: s3://bucket/prefix) do bucket do S3 de destino. O endereço do bucket deve estar no formato S3 na região inicial do painel de controle do S3 Storage Lens.

Note

- O Amazon S3 atualizará a política de permissões no bucket de destino para permitir que o S3 coloque dados nesse bucket.
- O console do S3 mostrará a permissão explícita do bucket de destino que será adicionada pelo Amazon S3 à política de bucket de destino na caixa de permissão do bucket de destino.

- Se o bucket do S3 de destino de exportação de métricas já tiver a criptografia no lado do servidor habilitada, todos os arquivos de exportação colocados lá também devem ter a criptografia do lado do servidor ativada.
5. Se você optar por ativar a criptografia do lado do servidor para seu painel, você deve escolher um tipo de chave de criptografia. Você pode escolher entre uma [chave do Amazon S3 \(SSE-S3\)](#) e uma [chave do AWS Key Management Service \(AWS KMS\) \(SSE-KMS\)](#).
 6. Se você escolheu uma chave do AWS KMS, deverá escolher entre as chaves do KMS ou inserir uma chave do nome do recurso da Amazon (ARN).
 7. Selecione Save changes.

Em seguida, você pode visualizar as métricas incluídas para este painel.

Desabilitar ou excluir painéis do Amazon S3 Storage Lens

O Amazon S3 Storage Lens agrega suas métricas de uso e atividade e exibe as informações no snapshot da conta na página inicial do console (Buckets) do Amazon S3, em painéis interativos ou por meio de uma exportação de métricas que você pode obter por download no formato CSV ou Parquet. Você pode usar o painel para visualizar insights e tendências, sinalizar discrepâncias e receber recomendações para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados. Você pode usar o S3 Storage Lens por meio do AWS Management Console, da AWS CLI, de AWS SDKs ou da API REST.

O painel padrão do Amazon S3 Storage Lens é o painel de conta padrão. Esse painel é pré-configurado pelo Amazon S3 para ajudá-lo a visualizar insights resumidos e tendências para o uso agregado de armazenamento e as métricas de atividade de toda a sua conta no console. Você não pode modificar seu escopo de configuração, mas pode atualizar a seleção de métricas das Métricas gratuitas para as Métricas e recomendações avançadas pagas, configurar a exportação de métricas opcionais ou até mesmo desativá-la. O painel padrão não pode ser excluído.

Você pode excluir ou desativar um painel do Amazon S3 Storage Lens do console do Amazon S3. Desativar ou excluir um painel impede que ele gere métricas no futuro. Um painel desativado ainda mantém suas informações de configuração, para que possa ser facilmente retomado quando reativado. Um painel desabilitado retém seus dados históricos até que não estejam mais disponíveis para consultas.

As seleções de dados para métricas gratuitas são mantidas por 14 dias, e os dados das seleções de métricas e recomendações avançadas são mantidos por 15 meses.

Tópicos

- [Desativação de um painel do Amazon S3 Storage Lens \(p. 1166\)](#)
- [Exclusão de um painel do Amazon S3 Storage Lens \(p. 1167\)](#)

Desativação de um painel do Amazon S3 Storage Lens

Para desativar um painel do S3 Storage Lens

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Storage Lens (Storage Lens), Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o painel que deseja desativar e, em seguida, escolha Disable (Desativar) na parte superior da lista.
4. Na página de confirmação, confirme se deseja desativar o painel digitando o nome do painel no campo de texto e escolha Confirm (Confirmar).

Exclusão de um painel do Amazon S3 Storage Lens

Note

Antes de excluir um painel, considere o seguinte:

- Como alternativa à exclusão de um painel, você pode desativar o painel para que ele esteja disponível para ser reativado no futuro. Para obter mais informações, consulte [Desativação de um painel do Amazon S3 Storage Lens \(p. 1166\)](#).
- A exclusão do painel excluirá todas as configurações associadas a ele.
- A exclusão de um painel tornará todos os dados de métricas históricas indisponíveis. Esses dados históricos ainda são mantidos por 15 meses. Se você quiser acessar esses dados novamente, crie um painel com o mesmo nome na mesma região inicial daquele que foi excluído.

Para excluir um painel do S3 Storage Lens

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Storage Lens (Storage Lens), Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o painel que deseja excluir e escolha Delete (Excluir) na parte superior da lista.
4. Na página Delete dashboards (Excluir painéis), confirme se deseja excluir o painel inserindo o nome do painel no campo de texto. Depois, selecione Confirm (Confirmar).

Trabalhar com AWS Organizations para criar painéis no nível da organização

O Amazon S3 Storage Lens agrega suas métricas de uso e atividade e exibe as informações no snapshot da conta na página inicial do console (Buckets) do Amazon S3, em painéis interativos ou por meio de uma exportação de métricas que você pode obter por download no formato CSV ou Parquet. Você pode usar o painel para visualizar insights e tendências, sinalizar discrepâncias e receber recomendações para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados. Você pode usar o S3 Storage Lens por meio do AWS Management Console, da AWS CLI, de AWS SDKs ou da API REST.

O painel padrão do Amazon S3 Storage Lens é o painel de conta padrão. Esse painel é pré-configurado pelo Amazon S3 para ajudá-lo a visualizar insights resumidos e tendências para o uso agregado de armazenamento e as métricas de atividade de toda a sua conta no console. Você não pode modificar seu escopo de configuração, mas pode atualizar a seleção de métricas das Métricas gratuitas para as Métricas e recomendações avançadas pagas, configurar a exportação de métricas opcionais ou até mesmo desativá-la. O painel padrão não pode ser excluído.

Você também pode criar painéis adicionais do S3 Storage Lens focados em Regiões da AWS específicas, buckets do S3 ou outras Contas da AWS em suas organizações.

O painel do Amazon S3 fornece um rico recurso de informações sobre seu escopo de armazenamento, representando mais de 30 métricas que representam tendências e informações, incluindo resumo de armazenamento, economia de custos, proteção de dados e atividade.

O Amazon S3 Storage Lens pode ser usado para coletar métricas de armazenamento e dados de uso de todas as contas que fazem parte da hierarquia do AWS Organizations. Para fazer isso, você deve usar o AWS Organizations e habilitar o acesso confiável do S3 Storage Lens usando sua conta de gerenciamento do AWS Organizations.

Quando o acesso confiável estiver habilitado, você pode adicionar acesso de administrador delegado às contas em sua organização. Essas contas podem criar painéis e configurações para toda a organização para o S3 Storage Lens. Para obter mais informações sobre como habilitar o acesso confiável, consulte [Amazon S3 Storage Lens e AWS Organizations](#) no Guia do usuário do AWS Organizations.

Os controles de console a seguir estão disponíveis apenas para as contas de gerenciamento do AWS Organizations.

Tópicos

- [Ativação do acesso confiável para o S3 Storage Lens em sua organização \(p. 1168\)](#)
- [Desativação de acesso confiável do S3 Storage Lens em sua organização \(p. 1168\)](#)
- [Registro de administradores delegados para o S3 Storage Lens \(p. 1169\)](#)
- [Cancelamento do registro de administradores delegados para o S3 Storage Lens \(p. 1169\)](#)

Ativação do acesso confiável para o S3 Storage Lens em sua organização

A ativação do acesso confiável permite que o Amazon S3 Storage Lens acesse sua hierarquia, associação e estrutura do AWS Organizations por meio de APIs do AWS. O S3 Storage Lens se torna um serviço confiável para toda a estrutura de sua organização. Ele pode criar funções vinculadas a serviços nas contas de gerenciamento ou administrador delegado da sua organização sempre que uma configuração de painel é criada.

A função vinculada ao serviço concede permissões do S3 Storage Lens para descrever organizações, listar contas, verificar uma lista de acesso de serviço para as organizações e obter administradores delegados para a organização. Isso permite que o S3 Storage Lens colete métricas de uso de armazenamento entre contas e atividades para painéis dentro de contas em suas organizações.

Para obter mais informações, consulte [Usar funções vinculadas a serviços para o Amazon S3 Storage Lens \(p. 582\)](#).

Note

- O acesso confiável só pode ser ativado pela conta de gerenciamento.
- Somente a conta de gerenciamento e os administradores delegados podem criar painéis ou configurações do S3 Storage Lens para sua organização.

Para permitir que o S3 Storage Lens tenha acesso confiável

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Storage Lens, Organization settings (Configurações da organização).
3. Em Organization access(Acesso às organizações), escolha Edit (Editar).

A página Organization access (Acesso à organização) é aberta. Aqui você pode Habilitar o acesso confiável para o S3 Storage Lens. Isso permite que você e quaisquer outros titulares de conta que você adicionar como administradores delegados criem painéis para todas as contas e armazenamento em sua organização.

Desativação de acesso confiável do S3 Storage Lens em sua organização

A desativação do acesso confiável limitará o S3 Storage Lens a funcionar apenas em nível de conta. Cada titular da conta só poderá ver os benefícios do S3 Storage Lens limitados ao escopo de sua conta e não à sua organização. Os painéis que requerem acesso confiável não serão mais atualizados, mas poderão consultar seus dados históricos de acordo com o [período em que os dados ficarão disponíveis para consulta](#).

A remoção de uma conta como administrador delegado limita o acesso às métricas do painel do S3 Storage Lens para funcionar apenas em nível de conta. Os painéis organizacionais que foram criados não serão mais atualizados, mas conseguirão consultar seus dados históricos de acordo com o [período em que os dados ficarão disponíveis para consulta](#).

Note

- A desativação do acesso confiável também desativa automaticamente todos os painéis no nível da organização, pois o S3 Storage Lens não terá mais acesso confiável às contas da organização para coletar e agrregar métricas de armazenamento.
- As contas de gerenciamento e de administrador delegado ainda poderão visualizar os dados históricos desses painéis desabilitados e consultar esses dados enquanto estiverem disponíveis.

Para desativar o acesso confiável para o S3 Storage Lens

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Storage Lens, Organization settings (Configurações da organização).
3. Em Organization access(Acesso às organizações), escolha Edit (Editar).

A página Organization access (Acesso à organização) é aberta. Agora você pode desativar o acesso confiável para o S3 Storage Lens.

Registro de administradores delegados para o S3 Storage Lens

Depois de habilitar o acesso confiável, você pode registrar o acesso de administrador delegado às contas em sua organização. Quando uma conta é registrada como administrador delegado, a conta recebe autorização para acessar todas as APIs do AWS Organizations somente leitura. Isso fornece visibilidade aos membros e estruturas da sua organização para que eles possam criar painéis do S3 Storage Lens em seu nome.

Para registrar administradores delegados para o S3 Storage Lens

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Storage Lens, Organization settings (Configurações da organização).
3. Na seção delegated access (acesso delegado), em Accounts (Contas), escolha Add account (Adicionar conta).

A página de Delegated admin access (Acesso admin delegado) é aberta. Aqui você pode adicionar o ID da Conta da AWS como administrador delegado para criar painéis no nível da organização para todas as contas e armazenamentos em sua organização.

Cancelamento do registro de administradores delegados para o S3 Storage Lens

Você pode cancelar o registro de acesso de administrador delegado a contas em sua organização. Quando uma conta é cancelada de registro como administrador delegado, a conta perde a autorização para acessar todas as APIs somente leitura do AWS Organizations que fornecem visibilidade aos membros e estruturas de sua organização.

Note

- Cancelando o registro de um administrador delegado também desativa automaticamente todos os painéis no nível da organização criados pelo administrador delegado.
- As contas de administrador delegado ainda podem visualizar os dados históricos desses painéis desabilitados de acordo com o período em que os dados ficarão disponíveis para consulta.

Para cancelar o registro de contas para acesso de administrador delegado

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação, escolha Storage Lens, Organization settings (Configurações da organização).
3. Na seção Accounts with delegated access(Contas com acesso delegado), escolha o ID da conta que deseja cancelar o registro e escolha Remove (Remover).

Exemplos de Amazon S3 Storage Lens usando a AWS CLI

O Amazon S3 Storage Lens agrega suas métricas de uso e atividade e exibe as informações no snapshot da conta na página inicial do console (Buckets) do Amazon S3, em painéis interativos ou por meio de uma exportação de métricas que você pode obter por download no formato CSV ou Parquet. Você pode usar o painel para visualizar insights e tendências, sinalizar discrepâncias e receber recomendações para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados. Você pode usar o S3 Storage Lens por meio do AWS Management Console, da AWS CLI, de AWS SDKs ou da API REST. Para obter mais informações, consulte [Avaliação da atividade de armazenamento e uso com o Amazon S3 Storage Lens](#).

Os exemplos a seguir mostram como você pode usar o S3 Storage Lens com o AWS Command Line Interface.

Tópicos

- [Arquivos auxiliares para usar o Amazon S3 Storage Lens \(p. 1170\)](#)
- [Uso de configurações do Amazon S3 Storage Lens com a AWS CLI \(p. 1172\)](#)
- [Usar o Amazon S3 Storage Lens com AWS Organizations usando a AWS CLI \(p. 1174\)](#)

Arquivos auxiliares para usar o Amazon S3 Storage Lens

Use os seguintes arquivos JSON para entradas de chave para seus exemplos.

JSON de configuração de exemplo do S3 Storage Lens

Example config.json

Contém detalhes de uma configuração de métricas e recomendações avançadas no nível das organizações do S3 Storage Lens.

Note

Aplicam-se cobranças adicionais para Métricas e recomendações avançadas. Para obter mais informações, consulte [Métricas e recomendações avançadas](#).

```
{  
    "Id": "SampleS3StorageLensConfiguration", //Use this property to identify S3 Storage  
    //Lens configuration.  
    "AwsOrg": { //Use this property when enabling S3 Storage Lens for AWS Organizations  
        "Arn": "arn:aws:organizations::222222222222:organization/o-abcdefgh"  
    },  
    "AccountLevel": {  
        "ActivityMetrics": {  
            "IsEnabled": true  
        },  
        "BucketLevel": {  
            "IsEnabled": true  
        }  
    }  
}
```

```
"ActivityMetrics": {  
    "Enabled":true //Mark this as false if you only want Free Metrics metrics.  
},  
"PrefixLevel":{  
    "StorageMetrics":{  
        "Enabled":true, //Mark this as false if you only want Free Metrics metrics.  
        "SelectionCriteria":{  
            "MaxDepth":5,  
            "MinStorageBytesPercentage":1.25,  
            "Delimiter":"/"  
        }  
    }  
},  
"Exclude": { //Replace with include if you prefer to include regions.  
    "Regions": [  
        "eu-west-1"  
    ],  
    "Buckets": [ //This attribute is not supported for Organizations-level configurations.  
        "arn:aws:s3:::source_bucket1"  
    ]  
},  
"Enabled": true, //Whether the configuration is enabled  
"DataExport": { //Details about the metrics export  
    "S3BucketDestination": {  
        "OutputSchemaVersion": "V_1",  
        "Format": "CSV", //You can add "Parquet" if you prefer.  
        "AccountId": "ExampleAWSAccountNo8",  
        "Arn": "arn:aws:s3:::destination-bucket-name", // The destination bucket for your metrics export must be in the same Region as your S3 Storage Lens configuration.  
        "Prefix": "prefix-for-your-export-destination",  
        "Encryption": {  
            "SSE": {}  
        }  
    }  
},  
}
```

Tags JSON de configuração de exemplo do S3 Storage Lens.

Example tags.json

```
[  
    {  
        "Key": "key1",  
        "Value": "value1"  
    },  
    {  
        "Key": "key2",  
        "Value": "value2"  
    }]
```

Permissões do IAM de configuração do S3 Storage Lens

Example permissions.json: nome específico do painel

Este exemplo de política mostra permissões do IAM do S3 Storage Lens com um nome de painel específico especificado. Substitua `your-dashboard-name` e `example-account-id` pelos seus próprios valores.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetStorageLensConfiguration",  
                "s3>DeleteStorageLensConfiguration",  
                "s3:PutStorageLensConfiguration"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "aws:ResourceTag/key1": "value1"  
                }  
            },  
            "Resource": "arn:aws:s3:us-east-1:example-account-id:storage-lens/your-  
dashboard-name"  
        }  
    ]  
}
```

Example permissions.json: nenhum nome de painel específico

Este exemplo de política mostra permissões do IAM do S3 Storage Lens sem um nome de painel específico especificado. Substituir **example-account-id** por seu ID da Conta da AWS.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetStorageLensConfiguration",  
                "s3>DeleteStorageLensConfiguration",  
                "s3:PutStorageLensConfiguration"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "aws:ResourceTag/key1": "value1"  
                }  
            },  
            "Resource": "arn:aws:s3:us-east-1:example-account-id:storage-lens/*"  
        }  
    ]  
}
```

Uso de configurações do Amazon S3 Storage Lens com a AWS CLI

Você pode usar a AWS CLI para listar, criar, obter e atualizar suas configurações do S3 Storage Lens. Os exemplos a seguir usam os arquivos JSON auxiliares para entradas de chave.

Tópicos

- [Colocar uma configuração do S3 Storage Lens \(p. 1173\)](#)
- [Colocar uma configuração de S3 Storage Lens sem tags \(p. 1173\)](#)
- [Obter uma configuração do S3 Storage Lens \(p. 1173\)](#)
- [Listar as configurações do S3 Storage Lens sem próximo token \(p. 1173\)](#)
- [Listar configurações do S3 Storage Lens \(p. 1173\)](#)

- Excluir uma configuração do S3 Storage Lens (p. 1173)
- Colocar tags em uma configuração do S3 Storage Lens (p. 1174)
- Obter tags para uma configuração do S3 Storage Lens (p. 1174)
- Excluir tags para uma configuração do S3 Storage Lens (p. 1174)

Colocar uma configuração do S3 Storage Lens

Example Coloca uma configuração do S3 Storage Lens

```
aws s3control put-storage-lens-configuration --account-id=222222222222 --config-id=your-configuration-id --region=us-east-1 --storage-lens-configuration=file://./config.json --tags=file://./tags.json
```

Colocar uma configuração de S3 Storage Lens sem tags

Example Colocar uma configuração do S3 Storage Lens

```
aws s3control put-storage-lens-configuration --account-id=222222222222 --config-id=your-configuration-id --region=us-east-1 --storage-lens-configuration=file://./config.json
```

Obter uma configuração do S3 Storage Lens

Example Obter uma configuração do S3 Storage Lens

```
aws s3control get-storage-lens-configuration --account-id=222222222222 --config-id=your-configuration-id --region=us-east-1
```

Listar as configurações do S3 Storage Lens sem próximo token

Example Listar as configurações do S3 Storage Lens sem próximo token

```
aws s3control list-storage-lens-configurations --account-id=222222222222 --region=us-east-1
```

Listar configurações do S3 Storage Lens

Example Listar configurações do S3 Storage Lens

```
aws s3control list-storage-lens-configurations --account-id=222222222222 --region=us-east-1 --next-token=abcdefgij1234
```

Excluir uma configuração do S3 Storage Lens

Example Excluir uma configuração do S3 Storage Lens

```
aws s3control delete-storage-lens-configuration --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id
```

Colocar tags em uma configuração do S3 Storage Lens

Example Colocar tags em uma configuração do S3 Storage Lens

```
aws s3control put-storage-lens-configuration-tagging --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id --tags=file://./tags.json
```

Obter tags para uma configuração do S3 Storage Lens

Example Obter tags para uma configuração do S3 Storage Lens

```
aws s3control get-storage-lens-configuration-tagging --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id
```

Excluir tags para uma configuração do S3 Storage Lens

Example Excluir tags para uma configuração do S3 Storage Lens

```
aws s3control delete-storage-lens-configuration-tagging --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id
```

Usar o Amazon S3 Storage Lens com AWS Organizations usando a AWS CLI

Use o Amazon S3 Storage Lens para coletar métricas de armazenamento e dados de uso de todas as contas que fazem parte da hierarquia do AWS Organizations. Para obter mais informações, consulte [Uso do Amazon S3 Storage Lens com o AWS Organizations](#).

Tópicos

- [Habilitar o acesso confiável das organizações para o S3 Storage Lens \(p. 1174\)](#)
- [Desativar o acesso confiável das organizações para o S3 Storage Lens \(p. 1174\)](#)
- [Registrar administradores delegados das Organizações para o S3 Storage Lens \(p. 1175\)](#)
- [Cancelar o registro de administradores delegados do Organizations para o S3 Storage Lens \(p. 1175\)](#)

Habilitar o acesso confiável das organizações para o S3 Storage Lens

Example Habilitar o acesso confiável das organizações para o S3 Storage Lens

```
aws organizations enable-aws-service-access --service-principal storage-lens.s3.amazonaws.com
```

Desativar o acesso confiável das organizações para o S3 Storage Lens

Example Desativar o acesso confiável das organizações para o S3 Storage Lens

```
aws organizations disable-aws-service-access --service-principal storage-lens.s3.amazonaws.com
```

Registrar administradores delegados das Organizações para o S3 Storage Lens

Example Registrar administradores delegados das Organizações para o S3 Storage Lens

```
aws organizations register-delegated-administrator --service-principal storage-lens.s3.amazonaws.com --account-id 123456789012
```

Cancelar o registro de administradores delegados do Organizations para o S3 Storage Lens

Example Cancelar o registro de administradores delegados do Organizations para o S3 Storage Lens

```
aws organizations deregister-delegated-administrator --service-principal storage-lens.s3.amazonaws.com --account-id 123456789012
```

Exemplos do Amazon S3 Storage Lens usando o SDK for Java

O Amazon S3 Storage Lens agrega suas métricas de uso e atividade e exibe as informações no snapshot da conta na página inicial do console (Buckets) do Amazon S3, em painéis interativos ou por meio de uma exportação de métricas que você pode obter por download no formato CSV ou Parquet. Você pode usar o painel para visualizar insights e tendências, sinalizar discrepâncias e receber recomendações para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados. Você pode usar o S3 Storage Lens por meio do AWS Management Console, da AWS CLI, de AWS SDKs ou da API REST. Para obter mais informações, consulte [Avaliação da atividade de armazenamento e uso com o Amazon S3 Storage Lens](#).

Os exemplos a seguir mostram como você pode usar o S3 Storage Lens com o AWS SDK for Java.

Tópicos

- [Usando configurações do Amazon S3 Storage Lens usando o SDK for Java \(p. 1175\)](#)
- [Uso de configurações do Amazon S3 Storage Lens com o AWS Organizations usando o SDK for Java \(p. 1183\)](#)

Usando configurações do Amazon S3 Storage Lens usando o SDK for Java

Você pode usar o SDK for Java para listar, criar, obter e atualizar suas configurações do S3 Storage Lens. Os exemplos a seguir usam os arquivos json auxiliares para entradas de chave.

Tópicos

- [Criar e atualizar uma configuração do S3 Storage Lens \(p. 1176\)](#)
- [Excluir uma configuração do S3 Storage Lens \(p. 1177\)](#)
- [Obtém uma configuração do S3 Storage Lens \(p. 1178\)](#)
- [Lista as configurações do S3 Storage Lens \(p. 1179\)](#)
- [Colocar tags em uma configuração do S3 Storage Lens \(p. 1180\)](#)
- [Obter tags para uma configuração do S3 Storage Lens \(p. 1181\)](#)
- [Excluir tags para uma configuração do S3 Storage Lens \(p. 1181\)](#)
- [Atualizar a configuração padrão do S3 Storage Lens com Métricas e recomendações avançadas \(p. 1182\)](#)

Criar e atualizar uma configuração do S3 Storage Lens

Example Criar e atualizar uma configuração do S3 Storage Lens

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSControl;
import com.amazonaws.services.s3control.AWSControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.ActivityMetrics;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.Format;
import com.amazonaws.services.s3control.model.Include;
import com.amazonaws.services.s3control.model.OutputSchemaVersion;
import com.amazonaws.services.s3control.model.PrefixLevel;
import com.amazonaws.services.s3control.model.PrefixLevelStorageMetrics;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.S3BucketDestination;
import com.amazonaws.services.s3control.model.SSES3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateAndUpdateDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        String exportAccountId = "Destination Account ID";
        String exportBucketArn = "arn:aws:s3:::destBucketName"; // The destination bucket
for your metrics export must be in the same Region as your S3 Storage Lens configuration.
        String awsOrgARN = "arn:aws:organizations::222222222222:organization/o-abcdefg";
        Format exportFormat = Format.CSV;

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                .withDelimiter("/")
                .withMaxDepth(5)
                .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
PrefixLevelStorageMetrics()
                .withIsEnabled(true)
                .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                .withPrefixLevel(new
PrefixLevel().withStorageMetrics(prefixStorageMetrics));
            AccountLevel accountLevel = new AccountLevel()
                .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                .withBucketLevel(bucketLevel);

            Include include = new Include()
                .withBuckets(Arrays.asList("arn:aws:s3:::bucketName"))
                .withRegions(Arrays.asList("us-west-2"));
        }
    }
}
```

```
StorageLensDataExportEncryption exportEncryption = new
StorageLensDataExportEncryption()
    .withSSES3(new SSES3());
S3BucketDestination s3BucketDestination = new S3BucketDestination()
    .withAccountId(exportAccountId)
    .withArn(exportBucketArn)
    .withEncryption(exportEncryption)
    .withFormat(exportFormat)
    .withOutputSchemaVersion(OutputSchemaVersion.V_1)
    .withPrefix("Prefix");
StorageLensDataExport dataExport = new StorageLensDataExport()
    .withS3BucketDestination(s3BucketDestination);

StorageLensAwsOrg awsOrg = new StorageLensAwsOrg()
    .withArn(awsOrgARN);

StorageLensConfiguration configuration = new StorageLensConfiguration()
    .withId(configurationId)
    .withAccountLevel(accountLevel)
    .withInclude(include)
    .withDataExport(dataExport)
    .withAwsOrg(awsOrg)
    .withIsEnabled(true);

List<StorageLensTag> tags = Arrays.asList(
    new StorageLensTag().withKey("key-1").withValue("value-1"),
    new StorageLensTag().withKey("key-2").withValue("value-2")
);

AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
    .withCredentials(new ProfileCredentialsProvider())
    .withRegion(US_WEST_2)
    .build();

s3ControlClient.putStorageLensConfiguration(new
PutStorageLensConfigurationRequest()
    .withAccountId(sourceAccountId)
    .withConfigId(configurationId)
    .withStorageLensConfiguration(configuration)
    .withTags(tags)
);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Excluir uma configuração do S3 Storage Lens

Example Exclua uma configuração do S3 Storage Lens.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
```

```
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.DeleteStorageLensConfigurationRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class DeleteDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            s3ControlClient.deleteStorageLensConfiguration(new
DeleteStorageLensConfigurationRequest()
                .withAccountId(sourceAccountId)
                .withConfigId(configurationId)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Obtém uma configuração do S3 Storage Lens

Example Obter uma configuração do S3 Storage Lens

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.GetStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.GetStorageLensConfigurationResult;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class GetDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            final StorageLensConfiguration configuration =

```

```
s3ControlClient.getStorageLensConfiguration(new
GetStorageLensConfigurationRequest()
    .withAccountId(sourceAccountId)
    .withConfigId(configurationId)
).getStorageLensConfiguration();

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

[Lista as configurações do S3 Storage Lens](#)

Example Lista as configurações do S3 Storage Lens

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.ListStorageLensConfigurationEntry;
import com.amazonaws.services.s3control.model.ListStorageLensConfigurationsRequest;

import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class ListDashboard {

    public static void main(String[] args) {
        String sourceAccountId = "Source Account ID";
        String nextToken = "nextToken";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            final List<ListStorageLensConfigurationEntry> configurations =
                s3ControlClient.listStorageLensConfigurations(new
ListStorageLensConfigurationsRequest()
                    .withAccountId(sourceAccountId)
                    .withNextToken(nextToken)
                ).getStorageLensConfigurationList();

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

```
        e.printStackTrace();
    }
}
```

Colocar tags em uma configuração do S3 Storage Lens

Example Colocar tags em uma configuração do S3 Storage Lens

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWS3Control;
import com.amazonaws.services.s3control.AWS3ControlClient;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationTaggingRequest;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class PutDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";

        try {
            List<StorageLensTag> tags = Arrays.asList(
                new StorageLensTag().withKey("key-1").withValue("value-1"),
                new StorageLensTag().withKey("key-2").withValue("value-2")
            );

            AWS3Control s3ControlClient = AWS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            s3ControlClient.putStorageLensConfigurationTagging(new
PutStorageLensConfigurationTaggingRequest()
                .withAccountId(sourceAccountId)
                .withConfigId(configurationId)
                .withTags(tags)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Obter tags para uma configuração do S3 Storage Lens

Example Obter tags para uma configuração do S3 Storage Lens

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.DeleteStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.GetStorageLensConfigurationTaggingRequest;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class GetDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            final List<StorageLensTag> s3Tags = s3ControlClient
                .getStorageLensConfigurationTagging(new
                    GetStorageLensConfigurationTaggingRequest()
                        .withAccountId(sourceAccountId)
                        .withConfigId(configurationId)
                ).getTags();

            System.out.println(s3Tags.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Excluir tags para uma configuração do S3 Storage Lens

Example Excluir tags para uma configuração do S3 Storage Lens

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.DeleteStorageLensConfigurationTaggingRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;
```

```
public class DeleteDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            s3ControlClient.deleteStorageLensConfigurationTagging(new
DeleteStorageLensConfigurationTaggingRequest()
                .withAccountId(sourceAccountId)
                .withConfigId(configurationId)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Atualizar a configuração padrão do S3 Storage Lens com Métricas e recomendações avançadas

Example Atualizar a configuração padrão do S3 Storage Lens com Métricas e recomendações avançadas

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.ActivityMetrics;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.Format;
import com.amazonaws.services.s3control.model.Include;
import com.amazonaws.services.s3control.model.OutputSchemaVersion;
import com.amazonaws.services.s3control.model.PrefixLevel;
import com.amazonaws.services.s3control.model.PrefixLevelStorageMetrics;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.S3BucketDestination;
import com.amazonaws.services.s3control.model.SSSE3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;
```

```
public class UpdateDefaultConfigWithPaidFeatures {

    public static void main(String[] args) {
        String configurationId = "default-account-dashboard"; // This configuration ID
        cannot be modified
        String sourceAccountId = "Source Account ID";

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                .withDelimiter("/")
                .withMaxDepth(5)
                .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
            PrefixLevelStorageMetrics()
                .with.IsEnabled(true)
                .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                .withActivityMetrics(new ActivityMetrics().with.IsEnabled(true))
                .withPrefixLevel(new
            PrefixLevel().withStorageMetrics(prefixStorageMetrics));
            AccountLevel accountLevel = new AccountLevel()
                .withActivityMetrics(new ActivityMetrics().with.IsEnabled(true))
                .withBucketLevel(bucketLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                .withId(configurationId)
                .withAccountLevel(accountLevel)
                .with.IsEnabled(true);

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion("US_WEST_2")
                .build();

            s3ControlClient.putStorageLensConfiguration(new
            PutStorageLensConfigurationRequest()
                .withAccountId(sourceAccountId)
                .withConfigId(configurationId)
                .withStorageLensConfiguration(configuration)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Note

Aplicam-se cobranças adicionais para Métricas e recomendações avançadas. Para obter mais informações, consulte [Métricas e recomendações avançadas](#).

Uso de configurações do Amazon S3 Storage Lens com o AWS Organizations usando o SDK for Java

Use o Amazon S3 Storage Lens para coletar métricas de armazenamento e dados de uso de todas as contas que fazem parte da hierarquia do AWS Organizations. Para obter mais informações, consulte [Uso do Amazon S3 Storage Lens com o AWS Organizations](#).

Tópicos

- [Habilitar o acesso confiável das organizações para o S3 Storage Lens \(p. 1184\)](#)
- [Desativar o acesso confiável das organizações para o S3 Storage Lens \(p. 1184\)](#)
- [Registrar administradores delegados das Organizações para o S3 Storage Lens \(p. 1185\)](#)
- [Cancelar o registro de administradores delegados do Organizations para o S3 Storage Lens \(p. 1186\)](#)

Habilitar o acesso confiável das organizações para o S3 Storage Lens

Example Habilitar o acesso confiável das organizações para o S3 Storage Lens

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import com.amazonaws.services.organizations.model.EnableAWSServiceAccessRequest;

public class EnableOrganizationsTrustedAccess {
    private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

    public static void main(String[] args) {
        try {
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.enableAWSServiceAccess(new EnableAWSServiceAccessRequest()
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
        }
    }
}
```

Desativar o acesso confiável das organizações para o S3 Storage Lens

Example Desativar o acesso confiável das organizações para o S3 Storage Lens

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import com.amazonaws.services.organizations.model.DisableAWSServiceAccessRequest;

public class DisableOrganizationsTrustedAccess {
    private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";
```

```
public static void main(String[] args) {
    try {
        AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
            .withCredentials(new ProfileCredentialsProvider())
            .withRegion(Regions.US_EAST_1)
            .build();

        // Make sure to remove any existing delegated administrator for S3 Storage Lens
        // before disabling access, otherwise the request will fail.
        organizationsClient.disableAWSServiceAccess(new
DisableAWSServiceAccessRequest()
            .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but AWS Organizations couldn't
process
        // it and returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // AWS Organizations couldn't be contacted for a response, or the client
        // couldn't parse the response from AWS Organizations.
        e.printStackTrace();
    }
}
```

Registrar administradores delegados das Organizações para o S3 Storage Lens

Example Registrar administradores delegados das Organizações para o S3 Storage Lens

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import com.amazonaws.services.organizations.model.RegisterDelegatedAdministratorRequest;

public class RegisterOrganizationsDelegatedAdministrator {
    private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

    public static void main(String[] args) {
        try {
            String delegatedAdminAccountId = "253880222538"; // Account Id for the
delegated administrator
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.registerDelegatedAdministrator(new
RegisterDelegatedAdministratorRequest()
                .withAccountId(delegatedAdminAccountId)
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
        }
    }
}
```

```
}
```

CANCELAR O REGISTRO DE ADMINISTRADORES DELEGADOS DO ORGANIZATIONS PARA O S3 STORAGE LENS

Example Cancelar o registro de administradores delegados do Organizations para o S3 Storage Lens

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import com.amazonaws.services.organizations.model.DeregisterDelegatedAdministratorRequest;

public class DeregisterOrganizationsDelegatedAdministrator {
    private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

    public static void main(String[] args) {
        try {
            String delegatedAdminAccountId = "Account ID"; // Account Id for the delegated
            administrator
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.deregisterDelegatedAdministrator(new
DeregisterDelegatedAdministratorRequest()
                .withAccountId(delegatedAdminAccountId)
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
            process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
        }
    }
}
```

Rastreamento de solicitações do Amazon S3 usando o AWS X-Ray

AWS X-Ray coleta dados sobre solicitações que sua aplicação atende. Em seguida, você pode visualizar e filtrar os dados para identificar e solucionar problemas de performance e erros em suas aplicações distribuídas e arquitetura de microsserviços. Para qualquer solicitação rastreada para a aplicação, é possível ver informações detalhadas sobre a solicitação, a resposta e as chamadas que a aplicação faz para recursos da AWS downstream, microsserviços, bancos de dados e APIs HTTP Web.

Para obter mais informações, consulte [O que é o AWS X-Ray?](#) no Guia do desenvolvedor do AWS X-Ray.

Tópicos

- [Como o X-ray funciona com o Amazon S3 \(p. 1187\)](#)
- [Regiões disponíveis \(p. 1187\)](#)

Como o X-ray funciona com o Amazon S3

AWS X-Ray oferece suporte à propagação de contexto de rastreamento para o Amazon S3, para que você possa visualizar solicitações completas à medida que elas percorrem toda a aplicação. O X-Ray agrupa os dados gerados pelos serviços individuais, como o Amazon S3, o AWS Lambda e o Amazon EC2, e os muitos recursos que compõem sua aplicação. Ele fornece uma visão geral de como sua aplicação está funcionando.

O Amazon S3 integra-se ao X-Ray para propagar o [contexto de rastreamento](#) e fornecer uma cadeia de solicitações com nós [upstream e downstream](#). Se um serviço upstream incluir um cabeçalho de rastreamento formatado válido com sua solicitação do S3, o Amazon S3 passará o cabeçalho de rastreamento ao entregar notificações de eventos para serviços downstream, como Lambda, Amazon SQS e Amazon SNS. Se você tiver todos esses serviços ativamente integrados ao X-Ray, eles serão vinculados em uma cadeia de solicitações para fornecer os detalhes completos de suas solicitações do Amazon S3.

Para enviar cabeçalhos de rastreamento de X-Ray por meio do Amazon S3, você deve incluir um [Id de rastreamento de X-Amzn formatado](#) em suas solicitações. Você também pode instrumentar o cliente do Amazon S3 usando os SDKs do AWS X-Ray. Para obter uma lista dos SDKs compatíveis, consulte a [documentação do AWS X-Ray](#).

Mapas de serviço

Os mapas de serviço do X-Ray mostram as relações entre o Amazon S3 e outros serviços e recursos da AWS em sua aplicação em tempo quase real. Para ver as solicitações completas usando os mapas de serviço X-Ray, você pode usar o console X-Ray para visualizar um mapa das conexões entre o Amazon S3 e outros serviços que sua aplicação usa. Você pode detectar facilmente onde estão ocorrendo altas latências, visualizar a distribuição de nós para esses serviços e, em seguida, detalhar os serviços e caminhos específicos que afetam a performance da aplicação.

Análise de X-Ray

Você também pode usar o console [Analytics](#) (Análises) do X-Ray para analisar rastreamentos, visualizar métricas como latência e taxas de falha e [gerar Insights](#) para ajudar a identificar e solucionar problemas. Este console também mostra métricas como latência média e taxas de falha. Para obter mais informações, consulte [Console do AWS X-Ray](#) no Guia do desenvolvedor do AWS X-Ray.

Regiões disponíveis

O suporte da AWS X-Ray para o Amazon S3 está disponível em todas as [regiões da AWS X-Ray](#). Para obter mais informações, consulte [Amazon S3 e AWS X-Ray](#) no Guia do desenvolvedor do AWS X-Ray.

Hospedagem de um site estático usando o Amazon S3

Você pode usar o Amazon S3 para hospedar um site estático. Em um site estático, as páginas da Web individuais incluem conteúdo estático. Elas também podem conter scripts do lado do cliente.

Ao contrário, um site dinâmico depende do processamento no servidor, incluindo scripts executados no servidor, como PHP, JSP ou ASP.NET. O Amazon S3 não oferece suporte a scripts no lado do servidor, mas a AWS tem outros recursos para hospedar sites dinâmicos. Para saber mais sobre a hospedagem de sites na AWS, consulte [Hospedagem web](#).

Note

É possível usar o console do AWS Amplify para hospedar uma aplicação Web de página única. O console do AWS Amplify é compatível com aplicações de página única criadas com frameworks de aplicações de página única (por exemplo, React JS, Vue JS, Angular JS e Nuxt) e geradores de sites estáticos (por exemplo, Gatsby JS, React-static, Jekyll e Hugo). Para obter mais informações, consulte [Conceitos básicos](#) no Guia do usuário do console do AWS Amplify. Os endpoints de site do Amazon S3 não são compatíveis com HTTPS. Se quiser usar HTTPS, você poderá utilizar o Amazon CloudFront para atender a um site estático hospedado no Amazon S3. Para obter mais informações, consulte [Como uso o CloudFront para atender a solicitações HTTPS para meu bucket do Amazon S3?](#) Para usar HTTPS com um domínio personalizado, consulte [Configuração de um site estático usando um domínio personalizado registrado no Route 53](#).

Para obter mais informações sobre a hospedagem de um site estático no Simple Storage Service (Amazon S3), incluindo instruções e demonstrações detalhadas, consulte os tópicos a seguir.

Tópicos

- [Endpoints de site \(p. 1188\)](#)
- [Habilitar a hospedagem de sites \(p. 1190\)](#)
- [Configurar um documento de índice \(p. 1194\)](#)
- [Configurar um documento de erro personalizado \(p. 1196\)](#)
- [Configuração de permissões para acesso ao site \(p. 1199\)](#)
- [\(Opcional\) Registrar em log o tráfego da web \(p. 1202\)](#)
- [\(Opcional\) Configurar um redirecionamento de uma página da Web \(p. 1203\)](#)

Endpoints de site

Quando você configura seu bucket como um site estático, o site fica disponível no endpoint de site específico da Região da AWS do bucket. 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 um endpoint de site e um endpoint de API REST \(p. 1190\)](#).

Dependendo da Região, os endpoints de site do Amazon S3 seguem um destes dois formatos.

- Região s3-website dash - `http://bucket-name.s3-website-Region.amazonaws.com`
- s3-website ponto(.) Região - `http://bucket-name.s3-website.Region.amazonaws.com`

Esses URLs retornam o documento de indexação padrão configurado para o site. Para obter uma lista completa dos endpoints do site do Amazon S3, consulte [Endpoints de site do Amazon S3](#).

Se você quiser que seu site seja público, será necessário fazer com que seu conteúdo seja publicamente legível para que seus clientes possam acessá-lo no endpoint de site. Para mais informações, consulte [Configuração de permissões para acesso ao site \(p. 1199\)](#).

Important

Os endpoints de site do Amazon S3 não oferecem suporte a HTTPS ou pontos de acesso. Se quiser usar HTTPS, você poderá usar o Amazon CloudFront para servir um site estático hospedado no Amazon S3. Para obter mais informações, consulte [Como uso o CloudFront para atender a solicitações HTTPS para meu bucket do Amazon S3?](#) Para usar HTTPS com um domínio personalizado, consulte [Configuração de um site estático usando um domínio personalizado registrado no Route 53](#).

Os buckets de pagamento pelo solicitante não permitem acesso por um endpoint de site. Qualquer solicitação para tal bucket recebe uma resposta 403 Acesso negado. Para obter mais informações, consulte [Configuração de buckets de Pagamento pelo solicitante para transferências de armazenamento e uso \(p. 155\)](#).

Tópicos

- [Exemplos de endpoint de site \(p. 1189\)](#)
- [Adicionar um registro DNS CNAME \(p. 1189\)](#)
- [Usar um domínio personalizado com o Route 53 \(p. 1190\)](#)
- [Principais diferenças entre um endpoint de site e um endpoint de API REST \(p. 1190\)](#)

Exemplos de endpoint de site

Os exemplos a seguir mostram como você pode acessar um bucket do Amazon S3 configurado como um site estático.

Example - solicitação de um objeto no nível raiz

Para solicitar um objeto específico que é armazenado no nível raiz do bucket, use a seguinte estrutura de URL.

```
http://bucket-name.s3-website.Region.amazonaws.com/object-name
```

Por exemplo, o URL a seguir solicita o objeto photo.jpg que está armazenado no nível raiz do bucket.

```
http://example-bucket.s3-website.us-west-2.amazonaws.com/photo.jpg
```

Example - solicitação de um objeto em um prefixo

Para solicitar um objeto armazenado em uma pasta em seu bucket, use a seguinte estrutura de URL.

```
http://bucket-name.s3-website.Region.amazonaws.com/folder-name/object-name
```

O URL a seguir solicita o objeto docs/doc1.html em seu bucket.

```
http://example-bucket.s3-website.us-west-2.amazonaws.com/docs/doc1.html
```

Adicionar um registro DNS CNAME

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 [Personalizar URLs do Amazon S3 com registros CNAME \(p. 1252\)](#).

Usar um domínio personalizado com o Route 53

Em vez de acessar o site usando um endpoint de site do Amazon S3, você pode usar seu próprio domínio registrado com Amazon Route 53 para veicular conteúdo, por exemplo, `example.com`. Você pode usar o Amazon S3 com o Route 53 para hospedar 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 [Configurar um site estático usando um domínio personalizado registrado no Route 53 \(p. 105\)](#).

Principais diferenças entre um endpoint de site e um endpoint de API REST

Um endpoint de site do Amazon S3 é otimizado para acesso de um navegador da Web. A tabela a seguir resume as principais diferenças entre um endpoint de API REST e um 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 completa dos endpoints do Amazon S3, consulte [Endpoints e cotas do Amazon S3](#) na Referência geral da AWS.

Habilitar a hospedagem de sites

Ao configurar um bucket como um site estático, você deve habilitar a hospedagem de sites estáticos, configurar um documento de índice e definir permissões.

Você pode habilitar a hospedagem estática de sites usando o console do Amazon S3, a API REST, os AWS SDKs, a AWS CLI ou o AWS CloudFormation.

Para configurar seu site com um domínio personalizado, consulte [Configurar um site estático usando um domínio personalizado registrado no Route 53 \(p. 105\)](#).

Uso do console do S3

Como habilitar a hospedagem de sites estáticos

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket para o qual você deseja habilitar a hospedagem de site estático.
3. Escolha Properties (Propriedades).
4. Em Static website hosting (Hospedagem estática de sites), escolha Edit (Editar).
5. Escolha Use this bucket to host a website (Usar este bucket para hospedar um site).
6. Em Static website hosting (Hospedagem estática de sites), escolha Enable (Ativar).
7. Em Index Document (Documento de índice), insira o nome do arquivo do documento de índice, que geralmente é `index.html`.

O nome do documento de índice diferencia letras maiúsculas de minúsculas e deve corresponder exatamente ao nome do arquivo do documento de índice HTML do qual você planeja fazer upload para o bucket do S3. Quando você configura um bucket para hospedagem de site, deve especificar um documento de índice. O Amazon S3 retorna esse documento de índice quando as solicitações são feitas para o domínio raiz ou alguma subpasta. Para obter mais informações, consulte [Configurar um documento de índice \(p. 1194\)](#).

8. Para fornecer seu próprio documento de erros personalizado para erros da classe 4XX, em Error document (Documento de erros), insira o nome de arquivo do documento de erros personalizado.

O nome do documento de erro diferencia letras maiúsculas de minúsculas e deve corresponder exatamente ao nome do arquivo do documento de erro HTML do qual você planeja fazer upload para o bucket do S3. Se você não especificar um documento de erro personalizado e ocorrer um erro, o Amazon S3 retornará um documento de erro HTML padrão. Para mais informações, consulte [Configurar um documento de erro personalizado \(p. 1196\)](#).

9. (Opcional) Se você quiser especificar regras avançadas de redirecionamento em Redirection rules (Regras de redirecionamento), use JSON para descrevê-las.

Por exemplo, você pode encaminhar solicitações condicionalmente de acordo com nomes de chave de objeto ou prefixos específicos na solicitação. Para obter mais informações, consulte [Configurar regras de redirecionamento para usar redirecionamentos condicionais avançados \(p. 1204\)](#).

10. Selecione Save changes.

O Amazon S3 permite a hospedagem estática de sites para seu bucket. Na parte inferior da página, em Static website hosting (Hospedagem estática de sites), você verá o endpoint do site do seu bucket.

11. Em Static website hosting (Hospedagem de sites estáticos), anote o Endpoint.

O Endpoint é o endpoint do site do Amazon S3 para o bucket. Depois de concluir a configuração do bucket como um site estático, é possível usar esse endpoint para testar o site.

Uso dos REST API

Para obter mais informações sobre o envio de solicitações REST diretamente para habilitar a hospedagem estática de sites, consulte as seções a seguir na Referência da API do Amazon Simple Storage Service:

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

- Site de DELETE Bucket

Uso da SDKs AWS

Para hospedar um site estático no Amazon S3, você configura um bucket do Amazon S3 para hospedagem de sites e faz upload do conteúdo do seu site no bucket. Você também pode usar os AWS SDKs para criar, atualizar e excluir a configuração do site de forma programática. 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.

.NET

O exemplo a seguir mostra como usar o AWS SDK for .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 Bucket website](#). Para obter mais informações sobre o recurso de site do Amazon S3, consulte [Hospedagem de um site estático usando o Amazon S3 \(p. 1188\)](#).

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 do Amazon S3 .NET \(p. 1268\)](#).

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

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

        static async Task AddWebsiteConfigurationAsync(string bucketName,
                                                       string indexDocumentSuffix,
                                                       string errorDocument)
        {
            try
            {
                // 1. Put the website configuration.
                PutBucketWebsiteRequest putRequest = new PutBucketWebsiteRequest()
                {
                    BucketName = bucketName,
                    WebsiteConfiguration = new WebsiteConfiguration()
                
```

```
        IndexDocumentSuffix = indexDocumentSuffix,
        ErrorDocument = errorDocument
    }
};

PutBucketWebsiteResponse response = await
client.PutBucketWebsiteAsync(putRequest);

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

PHP

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 usando o Amazon S3 \(p. 1188\)](#).

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Usar o AWS SDK for PHP e executar exemplos do PHP \(p. 1268\)](#).

```
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  
]);
```

Usar a AWS CLI

Para obter mais informações sobre como usar a AWS CLI para configurar um bucket do S3 como um site estático, consulte [site](#) na Referência de comandos da AWS CLI.

Depois, é necessário configurar o documento de índice e definir permissões. Para obter informações, consulte [Configurar um documento de índice \(p. 1194\)](#) e [Configuração de permissões para acesso ao site \(p. 1199\)](#).

Também é possível configurar um [documento de erro \(p. 1196\)](#), o [registro em log do tráfego da web \(p. 1202\)](#) ou um [redirecionamento \(p. 1203\)](#).

Configurar um documento de índice

Ao habilitar a hospedagem de sites, você também deve configurar e fazer upload de um 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.

Quando você habilita a hospedagem de sites estáticos para seu bucket, insere o nome do documento de índice (por exemplo, `index.html`). Depois de habilitar a hospedagem de sites estáticos para seu bucket, faça upload de um arquivo HTML com o nome do documento de índice para o bucket.

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 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 \(p. 1188\)](#).

Pastas e documentos de índice

No Amazon S3, um bucket é um contêiner de objetos simples. Ele não fornece nenhuma organização hierárquica, já que o sistema de arquivos em seu computador faz isso. No entanto, você pode criar uma hierarquia lógica usando nomes de chaves que envolvam uma estrutura de pastas.

Por exemplo, considere um bucket com três objetos e que tenha os nomes de chaves a seguir. Embora esses objetos sejam armazenados sem uma organização hierárquica física, você pode pressupor a seguinte estrutura lógica de pastas com base nos nomes das chaves.

- `sample1.jpg`: o objeto está na raiz do bucket.
- `photos/2006/Jan/sample2.jpg`: o objeto está na subpasta `photos/2006/Jan`.

- `photos/2006/Feb/sample3.jpg`: o objeto está na subpasta `photos/2006/Feb`.

No console do Amazon S3, você também pode criar uma pasta em um bucket. Por exemplo, você pode criar uma pasta chamada `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 do objeto será `photos/sample.jpg`.

Se você criar uma estrutura de pastas em seu bucket, deverá ter um documento de índice em cada nível. Em cada pasta, o documento de índice deve ter o mesmo nome, por exemplo, `index.html`. 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://bucket-name.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.

Configurar um documento de índice

Para configurar um documento de índice usando o console do S3, use o procedimento a seguir. Você também pode configurar um documento de índice usando a API REST, os AWS SDKs, a AWS CLI ou o AWS CloudFormation.

Quando você habilita a hospedagem de sites estáticos para seu bucket, insere o nome do documento de índice (por exemplo, `index.html`). Depois de habilitar a hospedagem de sites estáticos para seu bucket, faça upload de um arquivo HTML com esse nome de documento de índice para o bucket.

Como configurar o documento de índice

1. Criar um arquivo `index.html`

Se você não tiver um arquivo `index.html`, poderá usar o HTML a seguir para criar um:

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <title>My Website Home Page</title>
</head>
<body>
    <h1>Welcome to my website</h1>
    <p>Now hosted on Amazon S3!</p>
</body>
</html>
```

2. Salve o arquivo de índice localmente.

O nome do documento de índice deve corresponder exatamente ao nome do documento de índice que você inseriu na caixa de diálogo Hospedagem de site estático. O nome do documento de índice diferencia maiúsculas de minúsculas. Por exemplo, se você inserir `index.html` no nome do Documento de índice na caixa de diálogo Hospedagem de site estático, o nome do arquivo do documento de índice também deverá ser `index.html` e não `Index.html`.

3. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
4. Na lista Buckets, selecione o nome do bucket que você deseja usar para hospedar um site estático.

5. Habilite a hospedagem de sites estáticos para seu bucket e insira o nome exato do documento de índice (por exemplo, `index.html`). Para obter mais informações, consulte [Habilitar a hospedagem de sites \(p. 1190\)](#).

Depois de habilitar a hospedagem estática do site, vá para a etapa 6.

6. Para fazer upload do documento de índice para o bucket, siga um destes procedimentos:

- Arraste e solte o arquivo de índice na listagem de buckets do console.
- Escolha Upload (Fazer upload) e siga as instruções para escolher e fazer upload do arquivo de índice.

Para obter instruções detalhadas, consulte [Fazer upload de objetos \(p. 171\)](#).

7. (Opcional) Faça upload do conteúdo de outro site para o seu bucket.

Depois, é necessário definir permissões para acesso ao site. Para mais informações, consulte [Configuração de permissões para acesso ao site \(p. 1199\)](#).

Também é possível configurar um [documento de erro \(p. 1196\)](#), o [registro em log do tráfego da web \(p. 1202\)](#) ou um [redirecionamento \(p. 1203\)](#).

Configurar um documento de erro personalizado

Depois de configurar o bucket como um site estático, quando ocorre um erro, o Amazon S3 retorna um documento de erro HTML. Opcionalmente, você pode configurar seu bucket com um documento de erro personalizado para que o Amazon S3 retorne esse documento quando ocorrer um erro.

Note

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.

Tópicos

- [Códigos de resposta HTTP do Amazon S3 \(p. 1196\)](#)
- [Configurar um documento de erro personalizado \(p. 1198\)](#)

Códigos de resposta HTTP do Amazon S3

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.Region.amazonaws.com/</code>), o Amazon S3 retornará uma resposta 301 Moved Permanently 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://bucket-name.s3-website.Region.amazonaws.com/x</code> , sem uma barra no final, ele tenta localizar o objeto com o nome de chave x. Se o objeto não

Código de erro HTTP	Descrição
	for encontrado, o Amazon S3 determinará que a solicitação é para a subpasta <code>x</code> e redirecionará a solicitação adicionando uma barra no final, e retornará 302 Found.
304 Not Modified (304 Não modificado)	Os usuários do Amazon S3 usam cabeçalhos de solicitação <code>If-Modified-Since</code> , <code>If-Unmodified-Since</code> , <code>If-Match</code> e/ou <code>If-None-Match</code> para determinar se o objeto solicitado é igual ao da cópia em cache mantida pelo cliente. Se o objeto for o mesmo, o endpoint do site retornará uma resposta 304 Not Modified (304 não modificados).
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"> • O Amazon S3 determina que o URL do site refere-se a uma chave de objeto que não existe. • O Amazon S3 pressupõe que a solicitação é para um documento de índice que não existe. • Um bucket especificado no URL não existe. • Um bucket especificado no URL existe, mas não é configurado como um site. <p>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.</p> <p>Para obter informações sobre como o Amazon S3 interpreta o URL como uma solicitação de um objeto ou um documento de índice, consulte Configurar um documento de índice (p. 1194).</p>
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 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: DdQezl9vkuw5luD5HKsFaTDm9KH4PZzCPRkW3igimILbTu1DiYlvXjgyd7pVxq32

An Error Occurred While Attempting to Retrieve a Custom Error Document

- Code: AccessDenied
- Message: Access Denied

Configurar um documento de erro personalizado

Quando configura seu bucket como um site estático, você pode, opcionalmente, fornecer um documento de erro personalizado que contém uma mensagem de erro fácil de entender e ajuda adicional. O Amazon S3 retorna seu documento de erro personalizado somente para códigos de erro HTTP classe 4XX.

Para configurar um documento de erro personalizado usando o console do S3, siga as etapas abaixo. Você também pode configurar um documento de erro usando a API REST, os AWS SDKs, a AWS CLI ou o AWS CloudFormation. Para obter mais informações, consulte:

- [PutBucketWebsite](#) na Referência da API do Amazon Simple Storage Service
- [AWS::S3::Bucket WebsiteConfiguration](#) no Guia do usuário do AWS CloudFormation
- [put-bucket-website](#) na Referência de comando da AWS CLI

Ao habilitar a hospedagem de site estático para o bucket, insira o nome do documento de erro (por exemplo, **404.html**). Depois de habilitar a hospedagem de sites estáticos para seu bucket, faça upload de um arquivo HTML com esse nome de documento de erros para o bucket.

Para configurar um documento de erros

1. Crie um documento de erro, por exemplo **404.html**.
2. Salve o arquivo de documento de erros localmente.

O nome do documento de erros diferencia maiúsculas e minúsculas e deve corresponder exatamente ao nome que você insere ao habilitar a hospedagem estática do site. Por exemplo, se você inserir **404.html** como o nome do Error document (Documento de erro) na caixa de diálogo Static website hosting (Hospedagem de site estático), o nome de arquivo do documento de erro também deve ser **404.html**.

3. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
4. Na lista Buckets, selecione o nome do bucket que você deseja usar para hospedar um site estático.
5. Habilite a hospedagem de site estático para seu bucket e insira o nome exato do documento de erro (por exemplo, **404.html**). Para obter mais informações, consulte [Habilitar a hospedagem de sites \(p. 1190\)](#).

Depois de habilitar a hospedagem estática do site, vá para a etapa 6.

6. Para fazer upload do documento de erros para o bucket, siga um destes procedimentos:
 - Arraste e solte o arquivo de documento de erros na lista de buckets do console.
 - Escolha Upload (Fazer upload) e siga as instruções para escolher e fazer upload do arquivo de índice.

Para obter instruções detalhadas, consulte [Fazer upload de objetos \(p. 171\)](#).

Configuração de permissões para acesso ao site

Ao configurar um bucket como um site estático, se você desejar que seu site seja público, poderá conceder acesso público de leitura. Para tornar o bucket publicamente legível, é necessário desabilitar as configurações de bloqueio de acesso público do bucket e gravar uma política de bucket que conceda acesso público de leitura. Se o bucket contiver objetos que não são do proprietário do bucket, talvez também seja necessário adicionar uma lista de controle de acesso (ACL) de objeto que conceda acesso de leitura a todos.

Se você não quiser desativar as configurações de acesso público de bloqueio para seu bucket, mas ainda quiser que seu site seja público, poderá criar uma distribuição do Amazon CloudFront para servir seu site estático. Para obter mais informações, consulte [Usar uma distribuição do Amazon CloudFront para servir um site estático](#) no Guia do desenvolvedor do Amazon Route 53.

Note

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.

Tópicos

- [Etapa 1: Editar configurações de bloqueio de acesso público do S3 \(p. 1199\)](#)
- [Etapa 2: Adicionar uma política de bucket \(p. 1200\)](#)
- [Listas de controle de acesso de objetos \(p. 1201\)](#)

Etapa 1: Editar configurações de bloqueio de acesso público do S3

Se quiser configurar um bucket existente como um site estático que tenha acesso público, você deverá editar as configurações de bloqueio de acesso público desse bucket. Você também pode ter que editar suas configurações de bloqueio de acesso público no nível da conta. O Amazon S3 aplica a combinação mais restritiva das configurações de bloqueio de acesso público no nível do bucket e no nível da conta.

Por exemplo, se você permitir o acesso público a um bucket, mas bloquear todo o acesso público no nível da conta, o Amazon S3 continuará a bloquear o acesso público ao bucket. Neste cenário, você precisaria editar as configurações de bloqueio de acesso público no nível do bucket e no nível da conta. Para obter mais informações, consulte [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#).

Por padrão, o Amazon S3 bloqueia o acesso público à sua conta e aos seus buckets. Se quiser usar um bucket para hospedar um site estático, use estas etapas para editar as configurações de bloqueio de acesso público.

Warning

Antes de concluir esta etapa, revise [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#) para garantir que você entenda e aceite os riscos envolvidos em permitir o acesso público. Ao desativar as configurações de bloqueio de acesso público para tornar seu bucket

público, qualquer pessoa na Internet pode acessá-lo. Recomendamos que você bloquee todo o acesso público aos buckets.

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Escolha o nome do bucket configurado como um site estático.
3. Escolha Permissions (Permissões).
4. Em Block public access (bucket settings) (Bloqueio de acesso público (configurações de bucket)), escolha Edit (Editar).
5. Desmarque Block all public access (Bloquear todo acesso público) e escolha Save changes (Salvar alterações).

Warning

Antes de concluir esta etapa, revise [Bloquear o acesso público ao armazenamento do Amazon S3 \(p. 618\)](#) para garantir que você entenda e aceite os riscos envolvidos em permitir o acesso público. Ao desativar as configurações de bloqueio de acesso público para tornar seu bucket público, qualquer pessoa na Internet pode acessá-lo. Recomendamos que você bloquee todo o acesso público aos buckets.

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)



Account settings for Block Public Access are currently turned on

Account settings for Block Public Access that are enabled apply even if they are disabled for this bucket.

Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

Block public access to buckets and objects granted through new access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

Block public access to buckets and objects granted through any access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

Block public access to buckets and objects granted through new public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

Block public and cross-account access to buckets and objects through any public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

O Amazon S3 desativa as configurações do Bloqueio de acesso público para seu bucket. Para criar um site público e estático, você também pode ter que [editar as configurações de Bloqueio de acesso público](#) para sua conta antes de adicionar uma política de bucket. Se as configurações da conta para bloquear acesso público estiverem ativadas no momento, você verá uma observação em Block public access (bucket settings) (Bloqueio de acesso público (configurações de bucket)).

Etapa 2: Adicionar uma política de bucket

Para tornar os objetos no bucket publicamente legíveis, você deverá gravar uma política de bucket que conceda a todos a permissão s3:GetObject.

Depois de editar as configurações do Bloqueio de acesso público do S3, é possível adicionar uma política de bucket para conceder acesso público de leitura ao bucket. Ao conceder um acesso público de leitura, qualquer pessoa na Internet pode acessar seu bucket.

Important

A política a seguir é somente um exemplo e concede acesso total aos conteúdos do bucket. Antes de prosseguir com esta etapa, revise [Como posso proteger os arquivos no meu bucket do Amazon S3?](#) para garantir que você entende as práticas recomendadas a fim de proteger os arquivos no bucket do S3 e os riscos envolvidos na concessão de acesso público.

1. Em Buckets, escolha o nome do seu bucket.
2. Escolha Permissions (Permissões).
3. Em Bucket Policy (Política de bucket), escolha Edit (Editar).
4. Para conceder acesso público de leitura ao site, copie a política de bucket a seguir e cole-a no Bucket policy editor (Editor de política de bucket).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "PublicReadGetObject",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::Bucket-Name/*"  
            ]  
        }  
    ]  
}
```

5. Atualize o Resource para o nome do bucket.

Na política de bucket do exemplo anterior, **Bucket-Name** é um espaço reservado para o nome do bucket. Para usar essa política de bucket com seu próprio bucket, você deve atualizar esse nome para corresponder ao nome do seu bucket.

6. Selecione Save changes.

Uma mensagem é exibida indicando que a política de bucket foi adicionada com sucesso.

Se você vir um erro que diz `Policy has invalid resource`, confirme se o nome do bucket na política de bucket corresponde ao nome do seu bucket. Para obter informações sobre como adicionar uma política de bucket, consulte [Como adicionar uma política de bucket do S3?](#)

Se você receber uma mensagem de erro e não puder salvar a política do bucket, verifique suas configurações de acesso público para confirmar que você permite acesso público ao bucket.

Listas de controle de acesso de objetos

É possível usar uma política de bucket para conceder permissão de leitura aos seus objetos. No entanto, a política de bucket se aplica somente a objetos que sejam do proprietário do bucket. Se o seu bucket contiver objetos que não sejam do proprietário do bucket, ele deverá usar a lista de controle de acesso (ACL) do objeto para conceder permissão READ pública nesses objetos.

Por padrão, quando outra Conta da AWS carrega um objeto no bucket do S3, essa conta (que gravou o objeto) é a proprietária do objeto, tem acesso a ele e pode conceder acesso a outros usuários por meio

de ACLs. É possível usar Object Ownership para alterar esse comportamento padrão para que as ACLs sejam desabilitadas e você, como proprietário do bucket, automaticamente seja proprietário de todos os objetos de seu bucket. Como resultado, o controle de acesso para seus dados é baseado em políticas, como políticas do IAM, políticas de bucket do S3, políticas de endpoint da Virtual Private Cloud (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations.

A maioria dos casos de uso modernos no Amazon S3 não requer mais o uso de ACLs, e recomendamos desabilitar as ACLs, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com Object Ownership, é possível desabilitar ACLs e confiar em políticas para controle de acesso. Ao desabilitar ACLs, você pode facilmente manter um bucket com objetos carregados por diferentes Contas da AWS. Como proprietário do bucket, você possui todos os objetos do bucket e pode gerenciar o acesso a eles usando políticas. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket \(p. 638\)](#).

Important

Se o bucket usar a configuração aplicada pelo proprietário do bucket em S3 Object Ownership, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Solicitações para definir ACLs ou atualizar ACLs falham e retornam o código de erro `AccessControlListNotSupported`. Ainda há suporte para solicitações de leitura de ACLs.

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 [Visão geral da lista de controle de acesso \(ACL\) \(p. 587\)](#).

```
<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) Registrar em log o tráfego da web

Opcionalmente, é possível habilitar o registro em log de acesso ao servidor do Amazon S3 para um bucket configurado como um site estático. O registro em log de acesso ao servidor fornece registros detalhados para as solicitações que são feitas ao bucket. Para obter mais informações, consulte [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#). Se planeja usar o Amazon CloudFront para [acelerar seu site \(p. 118\)](#), você também pode usar o registro em log do CloudFront. Para obter mais informações, consulte [Configurar e usar logs de acesso](#) no Guia do desenvolvedor do Amazon CloudFront.

Como habilitar o registro em log do acesso ao servidor para o bucket de site estático

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na mesma região onde você criou o bucket configurado como um site estático, crie um bucket para registro em log, por exemplo `logs.example.com`.
3. Crie uma pasta para os arquivos de log do registro em log de acesso ao servidor (por exemplo, `logs`).

Quando você agrupa seus arquivos de dados de log em uma pasta, eles são mais fáceis de localizar.

4. (Opcional) Se você quiser usar o CloudFront para melhorar a performance do seu site, crie uma pasta para os arquivos de log do CloudFront (por exemplo, `cfn`).

Para mais informações, consulte [Acelerar seu site com o Amazon CloudFront \(p. 118\)](#).

5. Na lista Buckets, escolha o bucket.
6. Escolha Properties (Propriedades).
7. Em Server access logging (Registro de acesso ao servidor), selecione Edit (Editar).
8. Escolha Habilitar.
9. No Target bucket (Bucket de destino), escolha o destino do bucket e da pasta para os logs de acesso ao servidor:
 - Navegue até o local da pasta e do bucket:
 1. Escolha Browse S3 (Navegar no S3).
 2. Escolha o nome do bucket e, depois, escolha a pasta de logs.
 3. Selecione Choose path (Escolher caminho).
 - Insira o caminho do bucket do S3, por exemplo, `s3://logs.example.com/logs/`.
10. Selecione Save changes.

No bucket de log, agora você pode acessar seus logs. O Amazon S3 grava os logs de acesso ao site no bucket de log a cada duas horas.

(Opcional) Configurar um redirecionamento de uma página da Web

Se o bucket do Amazon S3 estiver configurado para hospedagem de site estático, você poderá configurar redirecionamentos para seu bucket ou para os objetos nele contidos. As opções a seguir estão disponíveis para a configuração de redirecionamentos.

Tópicos

- [Redirecionar solicitações para o endpoint do site do seu bucket para outro bucket ou domínio \(p. 1203\)](#)
- [Configurar regras de redirecionamento para usar redirecionamentos condicionais avançados \(p. 1204\)](#)
- [Redirecionar solicitações para um objeto \(p. 1209\)](#)

Redirecionar solicitações para o endpoint do site do seu bucket para outro bucket ou domínio

É possível redirecionar todas as solicitações de um endpoint de site de um bucket para outro bucket ou domínio. Se você redirecionar todas as solicitações, qualquer solicitação feita ao endpoint do site será redirecionada ao bucket ou domínio especificado.

Por exemplo, se o seu domínio raiz for `example.com` e você quiser atender a solicitações para `http://example.com` e `http://www.example.com`, é necessário criar dois buckets chamados `example.com` e `www.example.com`. Depois, mantenha o conteúdo no bucket `example.com` e configure o outro bucket `www.example.com` para redirecionar todas as solicitações ao bucket `example.com`. Para obter mais informações, consulte [Configurar um site estático usando um nome de domínio personalizado](#).

Como redirecionar solicitações para um endpoint de site do bucket

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Em Buckets, escolha o nome do bucket do qual você deseja redirecionar solicitações (por exemplo, `www.example.com`).

3. Escolha Properties (Propriedades).
4. Em Static website hosting (Hospedagem estática de sites), escolha Edit (Editar).
5. Selecione Redirect requests for an object (Redirecionar solicitações de um objeto).
6. Na caixa Host name (Nome do host) insira o endpoint do site para seu bucket ou seu domínio personalizado.

Por exemplo, se você estiver redirecionando para um endereço de domínio raiz, insira `example.com`.

7. Para Protocol (Protocolo), escolha o protocolo para as solicitações redirecionadas (none,http, ou https).

Se você não especificar um protocolo, a opção padrão será none.

8. Selecione Save changes.

Configurar regras de redirecionamento para usar 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 habilitar a hospedagem de site estático para seu bucket, você pode especificar opcionalmente regras de redirecionamento avançadas. O Amazon S3 tem uma limitação de 50 regras de roteamento por configuração de site. Se você precisar de mais de 50 regras de roteamento, poderá usar o redirecionamento de objetos. Para obter mais informações, consulte [Uso do console do S3 \(p. 1209\)](#).

Para obter mais informações sobre como configurar regras de roteamento usando a API REST, consulte [PutBucketWebsite](#) na Referência da API do Amazon Simple Storage Service.

Important

Para criar regras de redirecionamento no novo console do Amazon S3, você deve usar o JSON. Para exemplos de JSON, consulte [Exemplos de regras de redirecionamento \(p. 1207\)](#).

Para configurar regras de redirecionamento para um site estático

Para adicionar regras de redirecionamento para um bucket que já tem alojamento de site estático ativado, siga estas etapas.

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Bucket, escolha o nome do bucket configurado como um site estático.
3. Escolha Properties (Propriedades).
4. Em Static website hosting (Hospedagem estática de sites), escolha Edit (Editar).
5. Na caixa Redirection rules (Regras de redirecionamento), insira suas regras de redirecionamento em JSON.

No console do S3, você descreve as regras usando o JSON. Para exemplos de JSON, consulte [Exemplos de regras de redirecionamento \(p. 1207\)](#). O Amazon S3 tem uma limitação de 50 regras de roteamento por configuração de site.

6. Selecione Save changes.

Elementos de regra de roteamento

Veja a seguir a sintaxe geral para definir as regras de roteamento em uma configuração de site em XML. Para configurar regras de redirecionamento no novo console do S3, use o JSON. Para exemplos de JSON, consulte [Exemplos de regras de redirecionamento \(p. 1207\)](#).

JSON

```
[  
  {  
    "Condition": {  
      "HttpErrorCodeReturnedEquals": "string",  
      "KeyPrefixEquals": "string"  
    },  
    "Redirect": {  
      "HostName": "string",  
      "HttpRedirectCode": "string",  
      "Protocol": "http"|"https",  
      "ReplaceKeyPrefixWith": "string",  
      "ReplaceKeyWith": "string"  
    }  
  }  
]
```

Note: Redirect must each have at least one child element. You can have either ReplaceKeyPrefix with or ReplaceKeyWith but not both.

XML

```
<RoutingRules> =  
  <RoutingRules>  
    <RoutingRule>...</RoutingRule>  
    [<RoutingRule>...</RoutingRule>  
     ...]  
  </RoutingRules>  
  
<RoutingRule> =  
  <RoutingRule>  
    [ <Condition>...</Condition> ]  
    <Redirect>...</Redirect>  
  </RoutingRule>  
  
<Condition> =  
  <Condition>  
    [ <KeyPrefixEquals>...</KeyPrefixEquals> ]  
    [ <HttpErrorCodeReturnedEquals>...</HttpErrorCodeReturnedEquals> ]  
  </Condition>  
  Note: <Condition> must have at least one child element.  
  
<Redirect> =  
  <Redirect>  
    [ <HostName>...</HostName> ]  
    [ <Protocol>...</Protocol> ]  
    [ <ReplaceKeyPrefixWith>...</ReplaceKeyPrefixWith> ]  
    [ <ReplaceKeyWith>...</ReplaceKeyWith> ]  
    [ <HttpRedirectCode>...</HttpRedirectCode> ]  
  </Redirect>
```

Note: <Redirect> must have at least one child element. You can have either ReplaceKeyPrefix with or ReplaceKeyWith but not both.

A tabela a seguir descreve os elementos na regra de roteamento.

Nome	Descrição
<code>RoutingRules</code>	Contêiner para um conjunto de elementos <code>RoutingRule</code> .
<code>RoutingRule</code>	<p>Uma regra que identifica uma condição e o redirecionamento que é aplicado quando a condição é satisfeita.</p> <p>Condição:</p> <ul style="list-style-type: none"> Um contêiner <code>RoutingRules</code> deve ter pelo menos uma regra de roteamento.
<code>Condition</code>	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.
<code>KeyPrefixEquals</code>	<p>O prefixo do nome da chave de objeto do qual as solicitações são redirecionadas.</p> <p><code>KeyPrefixEquals</code> será obrigatório se <code>HttpErrorCodeReturnedEquals</code> não for especificado. Se <code>KeyPrefixEquals</code> e <code>HttpErrorCodeReturnedEquals</code> forem especificados, ambos deverão ser verdadeiros para a condição ser satisfeita.</p>
<code>HttpErrorCodeReturnedEquals</code>	<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><code>HttpErrorCodeReturnedEquals</code> será obrigatório se <code>KeyPrefixEquals</code> não for especificado. Se <code>KeyPrefixEquals</code> e <code>HttpErrorCodeReturnedEquals</code> forem especificados, ambos deverão ser verdadeiros para a condição ser satisfeita.</p>
<code>Redirect</code>	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 <code>RoutingRule</code> deve ter um elemento <code>Redirect</code> . Um elemento <code>Redirect</code> deve conter pelo menos um dos seguintes elementos irmãos: <code>Protocol</code> , <code>HostName</code> , <code>ReplaceKeyPrefixWith</code> , <code>ReplaceKeyWith</code> ou <code>HttpRedirectCode</code> .
<code>Protocol</code>	<p>O protocolo, <code>http</code> ou <code>https</code>, que será usado no cabeçalho <code>Location</code> que é retornado na resposta.</p> <p>Se um dos irmãos for fornecido, <code>Protocol</code> não será obrigatório.</p>
<code>HostName</code>	<p>O nome do host a ser usado no cabeçalho <code>Location</code> que é retornado na resposta.</p> <p>Se um dos irmãos for fornecido, <code>HostName</code> não será obrigatório.</p>
<code>ReplaceKeyPrefixWith</code>	O prefixo do nome de chave de objeto que substitui o valor de <code>KeyPrefixEquals</code> na solicitação de redirecionamento.

Nome	Descrição
	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 Location 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 Location que é retornado na resposta. Se um dos irmãos for fornecido, HttpRedirectCode não será obrigatório.

Exemplos de regras de redirecionamento

Os seguintes exemplos explicam tarefas comuns de redirecionamento:

Important

Para criar regras de redirecionamento no novo console do Amazon S3, você deve usar o JSON.

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, adicione a seguinte regra de roteamento à configuração do site:

JSON

```
[  
  {  
    "Condition": {  
      "KeyPrefixEquals": "docs/"  
    },  
    "Redirect": {  
      "ReplaceKeyPrefixWith": "documents/"  
    }  
  }  
]
```

XML

```
<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`.

JSON

```
[{
    {
        "Condition": {
            "KeyPrefixEquals": "images/"
        },
        "Redirect": {
            "ReplaceKeyWith": "folderdeleted.html"
        }
    }
}]
```

XML

```
<RoutingRules>
    <RoutingRule>
        <Condition>
            <KeyPrefixEquals>images/</KeyPrefixEquals>
        </Condition>
        <Redirect>
            <ReplaceKeyWith>folderdeleted.html</ReplaceKeyWith>
        </Redirect>
    </RoutingRule>
</RoutingRules>
```

Example 3: Redirecionar para um erro HTTP

Suponha que, quando um objeto solicitado não for encontrado, você queira redirecionar 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.

JSON

```
[{
    {
```

```
        "Condition": {
            "HttpErrorCodeReturnedEquals": "404"
        },
        "Redirect": {
            "HostName": "ec2-11-22-333-44.compute-1.amazonaws.com",
            "ReplaceKeyPrefixWith": "report-404/"
        }
    }
]
```

XML

```
<RoutingRules>
    <RoutingRule>
        <Condition>
            <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals >
        </Condition>
        <Redirect>
            <HostName>ec2-11-22-333-44.compute-1.amazonaws.com</HostName>
            <ReplaceKeyPrefixWith>report-404</ReplaceKeyPrefixWith>
        </Redirect>
    </RoutingRule>
</RoutingRules>
```

Redirecionar solicitações para um objeto

Você pode redirecionar solicitações de um objeto para outro objeto ou URL configurando o local de redirecionamento do site nos metadados do objeto. Você define o redirecionamento adicionando a propriedade `x-amz-website-redirect-location` aos metadados do objeto. Você pode usar o console do Amazon S3 para definir o Website Redirect Location (Local de redirecionamento de site) nos metadados do objeto. Se você usar a [API do Amazon S3 \(p. 1210\)](#), defina `x-amz-website-redirect-location`. A seguir, 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 do objeto definidos pelo sistema \(p. 166\)](#).

Quando você define um redirecionamento de página, pode manter ou excluir o conteúdo de objeto de origem. Por exemplo, se você tiver um objeto `page1.html` em seu bucket, poderá redirecionar todas as solicitações para essa página para outro objeto, `page2.html`. Você tem duas opções:

- Mantenha o conteúdo do objeto do `page1.html` e redirecione solicitações de página.
- Exclua o conteúdo de `page1.html` e carregue um objeto de byte zero chamado `page1.html` para substituir o objeto existente e redirecionar solicitações de páginas.

Uso do console do S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Buckets, escolha o nome do bucket que você configurou como um site estático (por exemplo, `example.com`).
3. Em Objects (Objetos), selecione seu objeto.
4. Escolha Actions (Ações) e escolha Edit metadata (Editar metadados).
5. Escolha Metadata (Metadados).
6. Escolha Add Metadata (Adicionar metadados).

7. Em Type (Tipo), escolha System Defined (Definido pelo sistema).
8. Em Key (Chave), escolha x-amz-website-redirect-location.
9. Em Value (Valor), insira o nome da chave do objeto o qual deseja redirecionar, por exemplo, /page2.html.

Para outro objeto no mesmo bucket, o prefixo / no valor é obrigatório. Você também pode definir o valor para um URL externo, por exemplo, <http://www.example.com>.

10. Escolha Edit metadata (Editar metadados).

Uso dos REST API

As ações da 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

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 um endpoint de site e um endpoint de API REST \(p. 1190\)](#).

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 para a página, envie 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 [GET objeto](#) com outros metadados de objeto, o Amazon S3 retorna o cabeçalho x-amz-website-redirect-location na resposta.

Desenvolvimento com o Amazon S3

Esta seção aborda tópicos relacionados ao desenvolvedor para usar o Amazon S3. Para obter mais informações, consulte os tópicos abaixo.

Tópicos

- [Fazer solicitações \(p. 1211\)](#)
- [Desenvolvimento com o Amazon S3 usando a AWS CLI \(p. 1257\)](#)
- [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers \(p. 1258\)](#)
- [Desenvolver com o Amazon S3 usando a API REST \(p. 1272\)](#)
- [Tratar erros de REST e SOAP \(p. 1276\)](#)
- [Referência do desenvolvedor \(p. 1279\)](#)

Fazer solicitações

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.

Cada 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 da 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 como obter chaves de acesso, consulte [Como faço para obter credenciais de segurança?](#) na Referência geral da AWS.

Se você estiver usando o AWS SDK, 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.

Tópicos

- [Sobre as chaves de acesso \(p. 1211\)](#)
- [Endpoints de solicitações \(p. 1213\)](#)
- [Fazer solicitações para o Amazon S3 por meio do IPv6 \(p. 1213\)](#)
- [Fazer solicitações usando os AWS SDKs \(p. 1220\)](#)
- [Fazer solicitações usando a API REST \(p. 1247\)](#)

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 pertencem à conta. Veja a seguir exemplos de chaves de acessos:

- 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 da AWS individuais 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 da AWS pai e são pagos por 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 em sua 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 aplicações criadas 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 do AWS STS ou o AWS SDK 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 aplicações. 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 \(p. 1286\)](#). Para obter

informações sobre o envio de solicitações usando AWS SDKs, consulte [Fazer solicitações usando os AWS SDKs \(p. 1220\)](#).

Para obter mais informações sobre o suporte do IAM para credenciais de segurança temporárias, consulte [Credenciais de segurança temporárias](#) no Manual 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 mais informações, consulte [Adicionar uma política de bucket para exigir MFA \(p. 524\)](#). 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 do [AWS Multi-Factor Authentication e Configuração do acesso à API com proteção MFA](#) no Manual 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 endpoints correspondentes, acesse [Regiões e endpoints](#) na Referência geral da AWS.

Fazer solicitações para o 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 IPv6 (Internet Protocol versão 6), além do protocolo IPv4. Os endpoints de pilha dupla do Amazon S3 oferecem suporte para buckets do S3 por IPv6 e IPv4. Não há custo adicional para acessar o Amazon S3 por meio do IPv6. Para obter mais informações sobre a definição de preços, consulte [Definição de preços do Amazon S3](#).

Tópicos

- [Conceitos básicos para fazer solicitações por meio do IPv6 \(p. 1213\)](#)
- [Como usar endereços do IPv6 em políticas do IAM \(p. 1214\)](#)
- [Testar a compatibilidade com endereços IP \(p. 1215\)](#)
- [Usar endpoints de pilha dupla do Amazon S3 \(p. 1216\)](#)

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 \(p. 1216\)](#).
- 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 [Como usar endereços do IPv6 em políticas do IAM \(p. 1214\)](#).
- 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 dos logs de acesso ao servidor do Amazon S3 \(p. 1052\)](#) e [Registrar em log as solicitações com registro em log de acesso ao servidor \(p. 1041\)](#).

Note

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

Como 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 de API do Amazon S3 funcionam da mesma maneira se você estiver acessando o Amazon S3 por IPv6 ou por 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 \(p. 1216\)](#).

Ao usar a AWS Command Line Interface (AWS CLI) e os AWS SDKs, você pode utilizar 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 \(p. 1216\)](#).
- Os AWS SDKs, consulte [Usar endpoints de pilha dupla dos AWS SDKs \(p. 1217\)](#).
- A API REST, consulte [Fazer solicitações para endpoints de pilha dupla usando a API REST \(p. 1248\)](#).

Recursos não disponíveis por meio do IPv6

No momento, o recurso a seguir não é compatível ao acessar um bucket do S3 com IPv6: hospedagem de site estático proveniente de um bucket do S3.

Como 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 [Identity and Access Management no Amazon S3 \(p. 419\)](#).

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:*",  
            "Condition": {  
                "IpAddress": "54.240.143.0/24"  
            }  
        }  
    ]  
}
```

```
    "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 [Limitar o acesso a endereços IP específicos \(p. 521\)](#).

Você pode revisar suas políticas de usuário do IAM usando o console do IAM em <https://console.aws.amazon.com/iam/>. Para obter mais informações sobre o IAM, consulte o [Manual do usuário do IAM](#). Para obter informações sobre políticas de bucket do S3, consulte [Adicionar uma política de bucket usando o console do Amazon S3 \(p. 516\)](#).

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 oferecem suporte para buckets do S3 por IPv6 e IPv4. Esta seção descreve como usar os endpoints de pilha dupla.

Tópicos

- [Endpoints de pilha dupla do Amazon S3 \(p. 1216\)](#)
- [Usar endpoints de pilha dupla da AWS CLI \(p. 1216\)](#)
- [Usar endpoints de pilha dupla dos AWS SDKs \(p. 1217\)](#)
- [Usar endpoints de pilha dupla da API REST \(p. 1219\)](#)

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 para o Amazon S3 por meio do IPv6 \(p. 1213\)](#).

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 oferece 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 [Métodos de acesso a um bucket \(p. 135\)](#). Para obter uma lista de todos os endpoints da Amazon S3, consulte [Regiões e endpoints na Referência geral da AWS](#).

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 \(p. 149\)](#).

Ao usar a AWS Command Line Interface (AWS CLI) e os AWS SDKs, você pode utilizar 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 AWS SDKs.

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 [Desenvolvimento com o Amazon S3 usando a AWS CLI \(p. 1257\)](#).

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 do Amazon S3 da AWS CLI](#).

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 AWS CLI estará disponível em breve. Para obter mais informações, consulte [Usar a AWS CLI \(p. 151\)](#).

Usar endpoints de pilha dupla dos AWS SDKs

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

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 exemplos de código Java no Amazon S3 \(p. 1267\)](#).

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

```
public class DualStackEndpoints {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_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
```

AWSExemplo do endpoint de pilha dupla do SDK para .NET

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

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

namespace Amazon.DocSamples.S3
{
    class DualStackEndpointTest
    {
        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()
        {
            var config = new AmazonS3Config
            {
                UseDualstackEndpoint = true,
                RegionEndpoint = bucketRegion
            };
            client = new AmazonS3Client(config);
            Console.WriteLine("Listing objects stored in a bucket");
        }
}
```

```
        ListingObjectsAsync().Wait();
    }

    private static async Task ListingObjectsAsync()
    {
        try
        {
            var request = new ListObjectsV2Request
            {
                BucketName = bucketName,
                MaxKeys = 10
            };
            ListObjectsV2Response response;
            do
            {
                response = await client.ListObjectsV2Async(request);

                // 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 == true);
        }
        catch (AmazonS3Exception amazonS3Exception)
        {
            Console.WriteLine("An AmazonS3Exception was thrown. Exception: " +
amazonS3Exception.ToString());
        }
        catch (Exception e)
        {
            Console.WriteLine("Exception: " + e.ToString());
        }
    }
}
```

Para uma amostra completa da .NET para objetos de listagem, consulte [Listar chaves de objeto programaticamente \(p. 260\)](#).

Para obter informações sobre como criar e testar um exemplo funcional .NET, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

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 \(p. 1248\)](#).

Fazer solicitações usando os AWS SDKs

Tópicos

- [Fazer solicitações usando credenciais de usuário do IAM ou da Conta da AWS \(p. 1220\)](#)
- [Fazer solicitações usando credenciais temporárias de usuário do IAM \(p. 1228\)](#)
- [Fazer solicitações usando credenciais temporárias de usuário federado \(p. 1237\)](#)

Você pode enviar solicitações autenticadas para o Amazon S3 usando o AWS SDK ou fazendo chamadas de API REST diretamente em sua aplicação. A API do AWS SDK usa as credenciais fornecidas por você 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 obter uma lista de AWS SDKs disponíveis, acesse [Código de exemplo e bibliotecas](#).

Fazer solicitações usando credenciais de usuário do IAM ou da Conta da AWS

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

Cada um desses AWS SDKs 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.

Para obter mais informações, consulte os tópicos abaixo.

Tópicos

- [Para criar um arquivo de credenciais da AWS local \(p. 1220\)](#)
- [Enviar solicitações autenticadas usando os AWS SDKs \(p. 1221\)](#)
- [Recursos relacionados \(p. 1227\)](#)

Para criar um arquivo de credenciais da AWS local

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

1. Faça login no AWS Management Console e abra o console do 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 [Criar usuários do IAM \(console\)](#) e siga as instruções até a etapa 8.
3. Escolha Download .csv (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 AWS SDKs.

Enviar solicitações autenticadas usando os AWS SDKs

Use os AWS SDKs para enviar solicitações autenticadas.

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`.
- Execute um dos métodos do `AmazonS3Client` para enviar solicitações ao 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 exemplos de código Java no Amazon S3 \(p. 1267\)](#).

Example

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

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

public class MakingRequests {

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

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

```

```
// 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();
}
}
```

.NET

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

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

Para obter mais informações, consulte [Fazer solicitações usando credenciais de usuário do IAM ou da Conta da AWS \(p. 1220\)](#).

Note

- Você pode criar o cliente `AmazonS3Client` 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.
- Você pode criar uma Conta da AWS e criar as contas de usuário necessárias. Você também pode gerenciar credenciais para essas contas de usuário. Você precisa dessas credenciais para executar a tarefa no exemplo a seguir. Para obter mais informações, consulte [Configurar credenciais da AWS](#) no Guia do desenvolvedor do AWS SDK for .NET.

Em seguida, você também pode configurar seu aplicativo para recuperarativamente perfis e credenciais e, em seguida, usar explicitamente essas credenciais ao criar um cliente de serviço da AWS. Para obter mais informações, consulte [Acessando credenciais e perfis em uma aplicação](#) no Guia do desenvolvedor do AWS SDK for .NET.

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 do Amazon S3 .NET \(p. 1268\)](#).

Example

```
using Amazon;
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);
        }
    }
}
```

Para obter exemplos funcionais, consulte [Visão geral de objetos Amazon S3 \(p. 161\)](#) e [Visão geral dos buckets \(p. 123\)](#). É possível testar esses exemplos usando as credenciais de sua Conta da AWS ou de um usuário do IAM.

Por exemplo, para listar todas as chaves de objetos em seu bucket, consulte [Listar chaves de objeto programaticamente \(p. 260\)](#).

PHP

Esta seção explica como usar uma classe da versão 3 do AWS SDK for 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 for PHP e executar exemplos do PHP \(p. 1268\)](#) e tenha o AWS SDK for 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

```
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 obter mais informações, consulte [Criar clientes anônimos](#) na [Documentação do AWS SDK for PHP](#).

Para ver um exemplo funcional, consulte [Visão geral de objetos Amazon S3 \(p. 161\)](#). É possível testar esses exemplos usando as credenciais de sua Conta da AWS ou de um usuário do IAM.

Para um exemplo de listagem de chaves de objeto em um bucket, consulte [Listar chaves de objeto programaticamente \(p. 260\)](#).

Ruby

Para poder usar a versão 3 do AWS SDK for Ruby para fazer chamadas para o Amazon S3, defina as credenciais de acesso da AWS que o SDK usa para verificar acesso aos seus buckets e objetos. Se você compartilhou credenciais configuradas no perfil de credenciais da AWS no sistema local, a versão 3 do SDK for Ruby poderá usar essas credenciais sem que você precise declará-las no código. Para obter mais informações sobre como configurar credenciais compartilhadas, consulte [Fazer solicitações usando credenciais de usuário do IAM ou da Conta da AWS \(p. 1220\)](#).

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::Client`.
2. Faz uma solicitação para o Amazon S3, enumerando objetos em um bucket com o método `list_objects_v2` do `Aws::S3::Client`. 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

```
# Prerequisites:
# - An existing Amazon S3 bucket.

require "aws-sdk-s3"

# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if all operations succeed; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless list_bucket_objects?(s3_client, 'doc-example-bucket')
def list_bucket_objects?(s3_client, bucket_name)
  puts "Accessing the bucket named '#{bucket_name}'..."
  objects = s3_client.list_objects_v2(
    bucket: bucket_name,
    max_keys: 50
  )

  if objects.count.positive?
    puts "The object keys in this bucket are (first 50 objects):"
    objects.contents.each do |object|
      puts object.key
    end
  end
end
```

```
    else
      puts "No objects found in this bucket."
    end

    return true
rescue StandardError => e
  puts "Error while accessing the bucket named '#{bucket_name}': #{e.message}"
  return false
end

# Full example call:
# Replace us-west-2 with the AWS Region you're using for Amazon S3.
def run_me
  region = "us-west-2"
  bucket_name = "BUCKET_NAME"
  s3_client = Aws::S3::Client.new(region: region)

  exit 1 unless list_bucket_objects?(s3_client, bucket_name)
end

run_me if $PROGRAM_NAME == __FILE__
```

Se você não tiver um arquivo de credenciais da AWS local, ainda poderá criar o recurso `Aws::S3::Client` e executar o código nos buckets e objetos do Amazon S3. As solicitações 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 que não esteja disponível publicamente.

É possível usar e expandir o trecho de código anterior para aplicações do SDK para Ruby, como no seguinte exemplo mais robusto.

```
# Prerequisites:
# - An existing Amazon S3 bucket.

require "aws-sdk-s3"

# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if all operations succeed; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless list_bucket_objects?(s3_client, 'doc-example-bucket')
def list_bucket_objects?(s3_client, bucket_name)
  puts "Accessing the bucket named '#{bucket_name}'..."
  objects = s3_client.list_objects_v2(
    bucket: bucket_name,
    max_keys: 50
  )

  if objects.count.positive?
    puts "The object keys in this bucket are (first 50 objects):"
    objects.contents.each do |object|
      puts object.key
    end
  else
    puts "No objects found in this bucket."
  end

  return true
rescue StandardError => e
  puts "Error while accessing the bucket named '#{bucket_name}': #{e.message}"
  return false
end
```

```
end

# Full example call:
# Replace us-west-2 with the AWS Region you're using for Amazon S3.
def run_me
    region = "us-west-2"
    bucket_name = "BUCKET_NAME"
    s3_client = Aws::S3::Client.new(region: region)

    exit 1 unless list_bucket_objects?(s3_client, bucket_name)
end

run_me if $PROGRAM_NAME == __FILE__
```

Recursos relacionados

- Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers (p. 1258)
- AWS SDK for PHP para a classe Aws\S3\S3Client do Amazon S
- AWS SDK for PHP Documentação do

Fazer solicitações usando credenciais temporárias de usuário do IAM

Uma Conta da AWS ou um usuário do IAM podem solicitar credenciais de segurança temporárias e usá-las para enviar solicitações autenticadas ao 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.

Java

Um usuário do IAM ou uma Conta da AWS podem solicitar credenciais de segurança temporárias (consulte [Fazer solicitações \(p. 1211\)](#)) usando o AWS SDK for Java e usá-las para acessar o Amazon S3. Essas credenciais expiram após a duração da sessão especificada.

Por padrão, a duração da sessão é de uma hora. Se usar credenciais de usuário do IAM, você poderá especificar a duração ao solicitar as credenciais de segurança temporárias de 15 minutos até a duração máxima de sessão para a função. Para obter mais informações sobre credenciais de segurança temporárias, consulte [Credenciais de segurança temporárias](#) no Manual do usuário do IAM. Para mais informações sobre fazer solicitações, consulte [Fazer solicitações \(p. 1211\)](#).

Como obter credenciais de segurança temporárias e acesso ao Amazon S3

1. Crie uma instância da classe `AWSecurityTokenService`. Para mais informações sobre a concessão de credenciais, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers \(p. 1258\)](#).
2. Recupere as credenciais de segurança temporárias para a função desejada chamando o método `assumeRole()` do cliente STS (Security Token Service).
3. Empacote as credenciais de segurança temporárias em um objeto `BasicSessionCredentials`. Use esse objeto para fornecer as credenciais de segurança temporárias para o cliente do Amazon S3.
4. Crie uma instância da classe `AmazonS3Client` usando as credenciais de segurança temporárias. Envie 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 pode 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 e usa essas credenciais para enviar uma solicitação autenticada ao Amazon S3.

Para testar o exemplo usando credenciais de usuário do IAM, crie 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 grupo de administradores](#) no Manual do usuário do IAM.

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
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.AWSSecurityTokenService;
import com.amazonaws.services.securitytoken.AWSSecurityTokenServiceClientBuilder;
import com.amazonaws.services.securitytoken.model.AssumeRoleRequest;
import com.amazonaws.services.securitytoken.model.AssumeRoleResult;
import com.amazonaws.services.securitytoken.model.Credentials;

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.
            AWSSecurityTokenService stsClient =
                AWSSecurityTokenServiceClientBuilder.standard()
                    .withCredentials(new
                        ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Obtain credentials for 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);
            AssumeRoleResult roleResponse = stsClientassumeRole(roleRequest);
            Credentials sessionCredentials = roleResponse.getCredentials();

            // Create a BasicSessionCredentials object that contains the credentials
            // you just retrieved.
            BasicSessionCredentials awsCredentials = 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 sessionCredentials object.
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withCredentials(new
                    AWSStaticCredentialsProvider(awsCredentials))
                .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();
        }
    }
}
```

.NET

Um usuário do IAM ou uma Conta da AWS podem solicitar credenciais de segurança temporárias usando o AWS SDK for .NET 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, você poderá especificar a duração ao solicitar as credenciais de segurança temporárias de 15 minutos até a duração máxima de sessão para a função. Para obter mais informações sobre credenciais de segurança temporárias, consulte [Credenciais de segurança temporárias](#) no Manual do usuário do IAM. Para mais informações sobre fazer solicitações, consulte [Fazer solicitações \(p. 1211\)](#).

Como obter credenciais de segurança temporárias e acesso ao Amazon S3

1. Crie uma instância do cliente do AWS Security Token Service, `AmazonSecurityTokenServiceClient`. Para mais informações sobre a concessão de credenciais, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers \(p. 1258\)](#).
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`. Use 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. Envie 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 de 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 ao Amazon S3.

Para testar o exemplo usando credenciais de usuário do IAM, crie 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 grupo de administradores](#) no Manual do usuário do IAM. Para mais informações sobre fazer solicitações, consulte [Fazer solicitações \(p. 1211\)](#).

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
```

```
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;
    }
}
}
```

PHP

Este exemplo considera que você já está seguindo as instruções para [Usar o AWS SDK for PHP e executar exemplos do PHP \(p. 1268\)](#) e tem o AWS SDK for 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 for PHP. Posteriormente, as credenciais temporárias podem 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, você poderá especificar a duração ao solicitar as credenciais de segurança temporárias de 15 minutos até a duração máxima de sessão para a função. Para obter mais informações sobre credenciais de segurança temporárias, consulte [Credenciais de segurança temporárias](#) no Manual do usuário do IAM. Para mais informações sobre fazer solicitações, consulte [Fazer solicitações \(p. 1211\)](#).

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 pode 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 ao Amazon S3. Para obter informações sobre a execução dos exemplos de PHP neste guia, consulte [Executar exemplos do PHP \(p. 1269\)](#).

Para testar o exemplo usando credenciais de usuário do IAM, crie 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 grupo de administradores](#) no Manual do usuário do IAM. Para obter exemplos de definição de duração de sessão usando credenciais de usuário do IAM para solicitar uma sessão, consulte [Fazer solicitações usando credenciais temporárias de usuário do IAM \(p. 1228\)](#).

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

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

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

$sts = new StsClient([
    'region' => 'us-west-2',
    'version' => '2011-06-15',
    'credentials' => [
        'access_key_id' => 'AKIAJ4RQHBTWZG3P6L3A',
        'secret_access_key' => 'wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY',
        'session_token' => 'STT123456789012345678901234567890'
    ]
]);
```

```
'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;
}
```

Ruby

Um usuário do IAM ou uma Conta da AWS podem solicitar credenciais de segurança temporárias usando o AWS SDK for 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, você poderá especificar a duração ao solicitar as credenciais de segurança temporárias de 15 minutos até a duração máxima de sessão para a função. Para obter mais informações sobre credenciais de segurança temporárias, consulte [Credenciais de segurança temporárias](#) no Manual do usuário do IAM. Para mais informações sobre fazer solicitações, consulte [Fazer solicitações \(p. 1211\)](#).

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. É possível especificar a duração da sessão somente ao 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.

```
# Prerequisites:
# - A user in AWS Identity and Access Management (IAM). This user must
#   be able to assume the following IAM role. You must run this code example
```

```
#   within the context of this user.
# - An existing role in IAM that allows all of the Amazon S3 actions for all of the
#   resources in this code example. This role must also trust the preceding IAM user.
# - An existing S3 bucket.

require "aws-sdk-core"
require "aws-sdk-s3"
require "aws-sdk-iam"

# Checks whether a user exists in IAM.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The user's name.
# @return [Boolean] true if the user exists; otherwise, false.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-west-2')
#   exit 1 unless user_exists?(iam_client, 'my-user')
def user_exists?(iam_client, user_name)
  response = iam_client.get_user(user_name: user_name)
  return true if response.user.user_name
rescue Aws::IAM::Errors::NoSuchEntity
  # User doesn't exist.
rescue StandardError => e
  puts "Error while determining whether the user " \
    "'#{user_name}' exists: #{e.message}"
end

# Creates a user in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The user's name.
# @return [AWS::IAM::Types::User] The new user.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-west-2')
#   user = create_user(iam_client, 'my-user')
#   exit 1 unless user.user_name
def create_user(iam_client, user_name)
  response = iam_client.create_user(user_name: user_name)
  return response.user
rescue StandardError => e
  puts "Error while creating the user '#{user_name}': #{e.message}"
end

# Gets a user in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The user's name.
# @return [AWS::IAM::Types::User] The existing user.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-west-2')
#   user = get_user(iam_client, 'my-user')
#   exit 1 unless user.user_name
def get_user(iam_client, user_name)
  response = iam_client.get_user(user_name: user_name)
  return response.user
rescue StandardError => e
  puts "Error while getting the user '#{user_name}': #{e.message}"
end

# Checks whether a role exists in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] The role's name.
# @return [Boolean] true if the role exists; otherwise, false.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-west-2')
```

```
#   exit 1 unless role_exists?(iam_client, 'my-role')
def role_exists?(iam_client, role_name)
  response = iam_client.get_role(role_name: role_name)
  return true if response.role.role_name
rescue StandardError => e
  puts "Error while determining whether the role " \
    "'#{role_name}' exists: #{e.message}"
end

# Gets credentials for a role in IAM.
#
# @param sts_client [Aws::STS::Client] An initialized AWS STS client.
# @param role_arn [String] The role's Amazon Resource Name (ARN).
# @param role_session_name [String] A name for this role's session.
# @param duration_seconds [Integer] The number of seconds this session is valid.
# @return [AWS::AssumeRoleCredentials] The credentials.
# @example
#   sts_client = Aws::STS::Client.new(region: 'us-west-2')
#   credentials = get_credentials(
#     sts_client,
#     'arn:aws:iam::123456789012:role/AmazonS3ReadOnly',
#     'ReadAmazonS3Bucket',
#     3600
#   )
#   exit 1 if credentials.nil?
def get_credentials(sts_client, role_arn, role_session_name, duration_seconds)
  Aws::AssumeRoleCredentials.new(
    client: sts_client,
    role_arn: role_arn,
    role_session_name: role_session_name,
    duration_seconds: duration_seconds
  )
rescue StandardError => e
  puts "Error while getting credentials: #{e.message}"
end

# Checks whether a bucket exists in Amazon S3.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The name of the bucket.
# @return [Boolean] true if the bucket exists; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless bucket_exists?(s3_client, 'doc-example-bucket')
def bucket_exists?(s3_client, bucket_name)
  response = s3_client.list_buckets
  response.buckets.each do |bucket|
    return true if bucket.name == bucket_name
  end
rescue StandardError => e
  puts "Error while checking whether the bucket '#{bucket_name}' " \
    "exists: #{e.message}"
end

# Lists the keys and ETags for the objects in an Amazon S3 bucket.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if the objects were listed; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless list_objects_in_bucket?(s3_client, 'doc-example-bucket')
def list_objects_in_bucket?(s3_client, bucket_name)
  puts "Accessing the contents of the bucket named '#{bucket_name}'..."
  response = s3_client.list_objects_v2(
    bucket: bucket_name,
```

```
    max_keys: 50
)

if response.count.positive?
  puts "Contents of the bucket named '#{bucket_name}' (first 50 objects):"
  puts "Name => ETag"
  response.contents.each do |obj|
    puts "#{obj.key} => #{obj.etag}"
  end
else
  puts "No objects in the bucket named '#{bucket_name}'."
end
return true
rescue StandardError => e
  puts "Error while accessing the bucket named '#{bucket_name}': #{e.message}"
end
```

Recursos relacionados

- Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers (p. 1258)
- AWS SDK for PHP para a classe Aws\S3\S3Client do Amazon S
- AWS SDK for PHP Documentação do

Fazer solicitações usando credenciais temporárias de usuário federado

Solicite credenciais de segurança temporárias e forneça-as às aplicações ou aos usuários federados que precisam de acesso aos seus recursos da AWS. Esta seção fornece exemplos de como usar o AWS SDK para obter credenciais de segurança temporárias para as aplicações ou usuários federados e enviar solicitações autenticadas ao Amazon S3 usando essas credenciais. Para obter uma lista de AWS SDKs 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 melhor segurança, somente um usuário do IAM com as permissões necessárias deve solicitar essas credenciais temporárias para garantir que o usuário federado obtenha, no máximo, as mesmas permissões do usuário do IAM solicitante. Em algumas aplicações, pode ser apropriado criar um usuário do IAM com permissões específicas com o único propósito de conceder credenciais de segurança temporárias às aplicações e aos usuários federados.

Java

Forneça credenciais de segurança temporárias para as aplicações e os usuários federados para que eles possam enviar solicitações autenticadas para acessar os seus recursos da AWS. Ao solicitar essas credenciais temporárias, forneça um nome de usuário e uma política do IAM que descreva as permissões de recursos que você 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 aplicações, recomendamos usar 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 sobre o 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 mais informações sobre a concessão de credenciais, consulte [Usar a AWS SDK for Java \(p. 1266\)](#).
- 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 você 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`. Use 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. Envie 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 usuário federado e usa as credenciais

para enviar solicitações autenticadas ao Amazon S3. Para executar o exemplo, crie um usuário do IAM com a política anexada que permita 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 grupo de administradores](#) no Manual 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 exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
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.regions.Regions;  
import com.amazonaws.services.s3.AmazonS3;  
import com.amazonaws.services.s3.AmazonS3ClientBuilder;  
import com.amazonaws.services.s3.model.ObjectListing;  
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 java.io.IOException;  
  
public class MakingRequestsWithFederatedTempCredentials {  
  
    public static void main(String[] args) throws IOException {  
        Regions clientRegion = Regions.DEFAULT_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();
}
}
```

.NET

Forneça credenciais de segurança temporárias para as aplicações e os usuários federados para que eles possam enviar solicitações autenticadas para acessar os seus recursos da AWS. Ao solicitar essas credenciais temporárias, forneça um nome de usuário e uma política do IAM que descreva as permissões de recursos que você 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 \(p. 1211\)](#).

Note

Ao solicitar credenciais de segurança temporárias para usuários federados e aplicações, 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 sobre o AWS Identity and Access Management](#).

Faça o seguinte:

- Crie uma instância de cliente AWS Security Token Service, classe `AmazonSecurityTokenServiceClient`. Para mais informações sobre a concessão de credenciais, consulte [Usar a AWS SDK for .NET \(p. 1267\)](#).
- Inicie uma sessão chamando o método `GetFederationToken` do cliente STS. Forneça as informações da sessão, incluindo o nome de usuário e uma política do IAM que você 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`. Use 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 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 grupo de administradores](#) no Manual do usuário do IAM.

- Use as credenciais de segurança do usuário do IAM para testar o exemplo a seguir. O exemplo envia a solicitação autenticada ao 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 do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
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;
}
}
}
```

PHP

Este tópico explica como usar classes da versão 3 do AWS SDK for PHP para solicitar credenciais de segurança temporárias para aplicações 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 for PHP e executar exemplos do PHP \(p. 1268\)](#) e tenha o AWS SDK for PHP devidamente instalado.

Forneça credenciais de segurança temporárias para as aplicações e os usuários federados para que eles possam enviar solicitações autenticadas para acessar os recursos da AWS. Ao solicitar essas credenciais temporárias, forneça um nome de usuário e uma política do IAM que descreva as permissões de recursos que você 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](#) no Manual do usuário do IAM. Para obter informações sobre

como fornecer credenciais de segurança temporárias para aplicativos e usuários federados, consulte [Fazer solicitações \(p. 1211\)](#).

Para garantir segurança adicional ao solicitar credenciais de segurança temporárias para usuários federados e aplicações, recomendamos usar 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 sobre federação de identidades, consulte as [Perguntas frequentes do AWS Identity and Access Management](#).

Para obter informações sobre a execução dos exemplos de PHP neste guia, consulte [Executar exemplos do PHP \(p. 1269\)](#).

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). Depois, use as credenciais de segurança temporárias para enviar solicitações autenticadas ao Amazon S3.

Para maior segurança, ao solicitar credenciais temporárias para terceiros, 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 da aplicação ao usuário federado, você também 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 grupo de administradores](#) no Manual do usuário do IAM.

Agora é possível usar as credenciais de segurança do usuário do IAM para testar o exemplo a seguir. O exemplo envia uma solicitação autenticada ao 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:

```
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;
}
```

Ruby

Forneça credenciais de segurança temporárias para as aplicações e os usuários federados para que eles possam enviar solicitações autenticadas para acessar os seus 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 aplicações. Para obter informações

sobre credenciais de segurança temporárias para aplicativos e usuários federados, consulte [Fazer solicitações \(p. 1211\)](#).

Note

Para garantir segurança adicional ao solicitar credenciais de segurança temporárias para usuários federados e aplicações, 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 sobre o 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.

```
# Prerequisites:  
# - An existing Amazon S3 bucket.  
  
require "aws-sdk-s3"  
require "aws-sdk-iam"  
require "json"  
  
# Checks to see whether a user exists in IAM; otherwise,  
# creates the user.  
#  
# @param iam [Aws::IAM::Client] An initialized IAM client.  
# @param user_name [String] The user's name.  
# @return [Aws::IAM::Types::User] The existing or new user.  
# @example  
#   iam = Aws::IAM::Client.new(region: 'us-west-2')  
#   user = get_user(iam, 'my-user')  
#   exit 1 unless user.user_name  
#   puts "User's name: #{user.user_name}"  
def get_user(iam, user_name)  
    puts "Checking for a user with the name '#{user_name}'..."  
    response = iam.get_user(user_name: user_name)  
    puts "A user with the name '#{user_name}' already exists."  
    return response.user  
# If the user doesn't exist, create them.  
rescue Aws::IAM::Errors::NoSuchEntity  
    puts "A user with the name '#{user_name}' doesn't exist. Creating this user..."  
    response = iam.create_user(user_name: user_name)  
    iam.wait_until(:user_exists, user_name: user_name)  
    puts "Created user with the name '#{user_name}'."  
    return response.user  
rescue StandardError => e  
    puts "Error while accessing or creating the user named '#{user_name}': #{e.message}"  
end  
  
# Gets temporary AWS credentials for an IAM user with the specified permissions.  
#  
# @param sts [Aws::STS::Client] An initialized AWS STS client.  
# @param duration_seconds [Integer] The number of seconds for valid credentials.  
# @param user_name [String] The user's name.  
# @param policy [Hash] The access policy.  
# @return [Aws::STS::Types::Credentials] AWS credentials for API authentication.  
# @example  
#   sts = Aws::STS::Client.new(region: 'us-west-2')  
#   credentials = get_temporary_credentials(sts, duration_seconds, user_name,  
#   {  
#     'Version' => '2012-10-17',
```

```
#      'Statement' => [
#        'Sid' => 'Stmt1',
#        'Effect' => 'Allow',
#        'Action' => 's3>ListBucket',
#        'Resource' => 'arn:aws:s3:::doc-example-bucket'
#      ]
#    }
#
#    exit 1 unless credentials.access_key_id
#    puts "Access key ID: #{credentials.access_key_id}"
def get_temporary_credentials(sts, duration_seconds, user_name, policy)
  response = sts.get_federation_token(
    duration_seconds: duration_seconds,
    name: user_name,
    policy: policy.to_json
  )
  return response.credentials
rescue StandardError => e
  puts "Error while getting federation token: #{e.message}"
end

# Lists the keys and ETags for the objects in an Amazon S3 bucket.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if the objects were listed; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless list_objects_in_bucket?(s3_client, 'doc-example-bucket')
def list_objects_in_bucket?(s3_client, bucket_name)
  puts "Accessing the contents of the bucket named '#{bucket_name}'..."
  response = s3_client.list_objects_v2(
    bucket: bucket_name,
    max_keys: 50
  )

  if response.count.positive?
    puts "Contents of the bucket named '#{bucket_name}' (first 50 objects):"
    puts "Name => ETag"
    response.contents.each do |obj|
      puts "#{obj.key} => #{obj.etag}"
    end
  else
    puts "No objects in the bucket named '#{bucket_name}'."
  end
  return true
rescue StandardError => e
  puts "Error while accessing the bucket named '#{bucket_name}': #{e.message}"
end

# Full example call:
# Replace us-west-2 with the AWS Region you're using for Amazon S3.
def run_me
  region = "us-west-2"
  user_name = "my-user"
  bucket_name = "doc-example-bucket"

  iam = Aws::IAM::Client.new(region: region)
  user = get_user(iam, user_name)

  exit 1 unless user.user_name

  puts "User's name: #{user.user_name}"
  sts = Aws::STS::Client.new(region: region)
  credentials = get_temporary_credentials(sts, 3600, user_name,
  {
```

```
"Version" => "2012-10-17",
"Statement" => [
  "Sid" => "Stmt1",
  "Effect" => "Allow",
  "Action" => "s3>ListBucket",
  "Resource" => "arn:aws:s3:::#{bucket_name}"
]
}

exit 1 unless credentials.access_key_id

puts "Access key ID: #{credentials.access_key_id}"
s3_client = Aws::S3::Client.new(region: region, credentials: credentials)

exit 1 unless list_objects_in_bucket?(s3_client, bucket_name)
end

run_me if $PROGRAM_NAME == __FILE__
```

Recursos relacionados

- Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers (p. 1258)
- AWS SDK for PHP para a classe Aws\S3\S3Client do Amazon S
- AWS SDK for PHP Documentação do

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 de todos os endpoints da Amazon S3, consulte [Regiões e endpoints](#) na Referência geral da AWS.

Criar nomes de host do S3 para solicitações de API REST

Endpoints do Amazon S3 seguem a estrutura mostrada abaixo:

```
s3.Region.amazonaws.com
```

Os endpoints de pontos de acesso do Amazon S3 e endpoints de pilha dupla também seguem a estrutura padrão:

- Pontos de acesso do Amazon S3 -s3-accesspoint.*Region*.amazonaws.com
- Pilha dupla - s3.dualstack.*Region*.amazonaws.com

Para obter uma lista completa de regiões e endpoints do Amazon S3, consulte [Amazon S3 endpoints and quotas](#) (Endpoints e cotas do Amazon S3) na Amazon Web Services General Reference (Referência geral da Amazon Web Services).

Solicitações de estilo de hospedagem virtual e caminho

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 [Hospedagem virtual de buckets \(p. 1249\)](#).

Example Solicitação no estilo de hospedagem virtual

Veja a seguir um exemplo de uma solicitação de estilo de hospedagem virtual para excluir o arquivo `puppy.jpg` do bucket `examplebucket` na região Oeste dos EUA (Oregon). Para obter mais informações sobre solicitações de estilo hosted virtual, consulte [Solicitações no estilo de hospedagem virtual \(p. 1250\)](#).

```
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
```

Atualmente, o Amazon S3 é compatível com o acesso a URLs no estilo de hospedagem virtual e no estilo de caminho em todas as Regiões da AWS. No entanto, os URLs no estilo de caminho serão descontinuados no futuro. Para obter mais informações, consulte a observação Importante a seguir.

Para obter mais informações sobre solicitações de estilo virtual, consulte [Solicitações no estilo de caminho \(p. 1250\)](#).

Important

Atualização (23 de setembro de 2020): para garantir que os clientes tenham o tempo necessário para fazer a transição para URLs no estilo de hospedagem virtual, decidimos postergar a desativação de URLs no estilo de caminho. Para obter mais informações, consulte [Amazon S3 Path Deprecation Plan – The Rest of the Story](#) no Blog de notícias da AWS.

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` na região Oeste dos EUA (Oregon).

```
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 \(p. 1216\)](#).

Para obter mais informações sobre como fazer solicitações usando a API REST, consulte os tópicos abaixo.

Tópicos

- [Hospedagem virtual de buckets \(p. 1249\)](#)
- [Redirecionamento de solicitação e a API REST \(p. 1255\)](#)

Hospedagem virtual de buckets

Hospedagem virtual é a prática de atender vários sites a partir de um único servidor web. Uma maneira de diferenciar sites em suas solicitações de API REST do Amazon S3 é usar o nome de host aparente do URI da solicitação em vez de apenas a parte do nome do caminho do URI. Uma solicitação REST comum do Amazon S3 especifica um bucket usando o primeiro componente delimitado por barra do caminho do URI da solicitação. Em vez disso, você pode usar 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 são acessíveis automaticamente (para tipos limitados de solicitações) em `https://bucket-name.s3.region-code.amazonaws.com`. Para obter uma lista completa de regiões e endpoints do Amazon S3, consulte [Amazon S3 endpoints and quotas \(Endpoints e cotas do Amazon S3\)](#) na Amazon Web Services General Reference (Referência geral da Amazon Web Services).

Além disso, a hospedagem virtual tem outros benefícios. 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.bucket-name.com/`. Você também pode publicar 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` e `crossdomain.xml` serão todos encontrados na raiz.

Important

Ao usar buckets no estilo de hospedagem virtual com SSL, o certificado curinga SSL corresponde apenas a buckets que não contêm pontos (.). Para contornar essa limitação, use HTTP ou escreva sua própria lógica de verificação de certificado. Para obter mais informações, consulte [Amazon S3 Path Deprecation Plan](#) (Plano de depreciação de caminhos do Amazon S3) no AWS News Blog.

Tópicos

- [Solicitações no estilo de caminho \(p. 1250\)](#)
- [Solicitações no estilo de hospedagem virtual \(p. 1250\)](#)
- [Especificação de bucket do cabeçalho Host HTTP \(p. 1250\)](#)
- [Exemplos \(p. 1251\)](#)
- [Personalizar URLs do Amazon S3 com registros CNAME \(p. 1252\)](#)
- [Como associar um nome de host a um bucket do Amazon S3 \(p. 1253\)](#)

- [Limitações \(p. 1253\)](#)
- [Compatibilidade retroativa \(p. 1253\)](#)

Solicitações no estilo de caminho

Atualmente, o Amazon S3 é compatível com o acesso a URLs no estilo de hospedagem virtual e no estilo de caminho em todas as Regiões da AWS. No entanto, os URLs no estilo de caminho serão descontinuados no futuro. Para obter mais informações, consulte a observação Importante a seguir.

No Amazon S3, os URLs no estilo de caminho usam o seguinte formato:

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

Por exemplo, se você criar um bucket chamado DOC-EXAMPLE-BUCKET1 na região Oeste dos EUA (Oregon) e quiser acessar o objeto puppy.jpg nele, use o seguinte URL no estilo de caminho:

```
https://s3.us-west-2.amazonaws.com/DOC-EXAMPLE-BUCKET1/puppy.jpg
```

Important

Atualização (23 de setembro de 2020): para garantir que os clientes tenham o tempo necessário para fazer a transição para URLs no estilo de hospedagem virtual, decidimos postergar a desativação de URLs no estilo de caminho. Para obter mais informações, consulte [Amazon S3 Path Deprecation Plan – The Rest of the Story](#) no Blog de notícias da AWS.

Solicitações no estilo de hospedagem virtual

Em um URL de estilo hospedado virtual, o nome do bucket faz parte do nome do domínio no URL.

Os URLs no estilo de hospedagem virtual do Amazon S3 usam o seguinte formato:

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

Neste exemplo, DOC-EXAMPLE-BUCKET1 é o nome do bucket, Oeste dos EUA (Oregon) é a região, e puppy.png é o nome da chave.

```
https://DOC-EXAMPLE-BUCKET1.s3.us-west-2.amazonaws.com/puppy.png
```

Especificação de bucket do cabeçalho 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.region-code.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.region-code.amazonaws.com, o nome do bucket será o componente inicial do valor do cabeçalho Host até .s3.region-code.amazonaws.com. A chave da solicitação será o seu URI. Essa interpretação expõe buckets como subdomínios do .s3.region-code.amazonaws.com, conforme ilustrado pelo terceiro e pelo quarto exemplos 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) de nome canônico para o Amazon S3. O procedimento para registrar nomes de domínio e configurar o CNAME 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 – URLs no estilo de caminho e solicitações

Este exemplo usa o seguinte:

- Nome do bucket - example.com
- Região: Leste dos EUA (Norte da Virgínia)
- Nome da chave: homepage.html

O URL é o seguinte:

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

A solicitação é a seguinte:

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

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

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

Para obter informações sobre nomes compatíveis do DNS, consulte [Limitações \(p. 1253\)](#). Para obter mais informações sobre chaves, consulte [Chaves \(p. 5\)](#).

Example – Solicitações e URLs no estilo de hospedagem virtual

Este exemplo usa o seguinte:

- Nome do bucket: DOC-EXAMPLE-BUCKET1.eu
- Região: Europa (Irlanda)
- Nome da chave: homepage.html

O URL é o seguinte:

```
http://DOC-EXAMPLE-BUCKET1.eu.s3.eu-west-1.amazonaws.com/homepage.html
```

A solicitação é a seguinte:

```
GET /homepage.html HTTP/1.1
Host: DOC-EXAMPLE-BUCKET1.eu.s3.eu-west-1.amazonaws.com
```

Example – método de alias CNAME

Para usar este método, é necessário configurar o nome de DNS como um alias CNAME para **bucket-name**.s3.us-east-1.amazonaws.com. Para obter mais informações, consulte [Personalizar URLs do Amazon S3 com registros CNAME \(p. 1252\)](#).

Este exemplo usa o seguinte:

- Nome do bucket - `example.com`
- Nome da chave: `homepage.html`

O URL é o seguinte:

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

O exemplo é o seguinte:

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

Personalizar URLs do Amazon S3 com registros CNAME

Dependendo da necessidade, é possível que você não queira que `s3.region-code.amazonaws.com` apareça no site ou serviço. Por exemplo, se você estiver hospedando as imagens do site no Amazon S3, talvez prefira `http://images.example.com/` em vez de `http://images.example.com.s3.us-east-1.amazonaws.com/`. Qualquer bucket com um nome compatível com o DNS pode ser mencionado da seguinte forma: `http://BucketName.s3.Region.amazonaws.com/[Filename]`, por exemplo, `http://images.example.com.s3.us-east-1.amazonaws.com/mydog.jpg`. Ao usar o CNAME, mapeie `images.example.com` para um nome de host do Amazon S3, para que o URL anterior se torne `http://images.example.com/mydog.jpg`.

O nome do bucket deve ser o mesmo que o CNAME. Por exemplo, se você criar um CNAME para mapear `images.example.com` para `images.example.com.s3.us-east-1.amazonaws.com`, `http://images.example.com/filename` e `http://images.example.com.s3.us-east-1.amazonaws.com/filename` serão o mesmo.

O registro DNS do CNAME deve apelidar o nome do domínio para o nome de host apropriado do estilo de hospedagem virtual. Por exemplo, se o nome do bucket e o nome do domínio forem `images.example.com` e o bucket estiver na região Leste dos EUA (Norte da Virgínia), o alias do registro do CNAME deverá ser `images.example.com.s3.us-east-1.amazonaws.com`.

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

O Amazon S3 usa o nome de host para determinar o nome do bucket. O nome do CNAME e o nome do bucket devem ser os mesmos. Por exemplo, suponha que você tenha configurado `www.example.com` como um CNAME para `www.example.com.s3.us-east-1.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
```

O Amazon S3 vê apenas o nome do host original `www.example.com` e não tem conhecimento do mapeamento CNAME usado para resolver a solicitação.

Você pode usar qualquer endpoint do Amazon S3 em um alias CNAME. Por exemplo, `s3.ap-southeast-1.amazonaws.com` pode ser usado em aliases CNAME. Para obter mais informações sobre endpoints, consulte [Endpoints de solicitações \(p. 1213\)](#). Para criar um site estático usando um domínio

personalizado, consulte [Configurar um site estático usando um domínio personalizado registrado no Route 53 \(p. 105\)](#).

Important

Ao usar URLs personalizados com CNAMEs, você precisará garantir que exista um bucket correspondente para qualquer registro CNAME ou alias configurado. Por exemplo, se você criar entradas DNS para `www.example.com` e `login.example.com` com o objetivo de publicar conteúdo da Web usando o S3, será necessário criar os dois buckets, `www.example.com` e `login.example.com`.

Quando um registro CNAME ou alias é configurado apontando para um endpoint do S3 sem um bucket correspondente, qualquer usuário da AWS pode criar esse intervalo e publicar conteúdo com o alias configurado, mesmo que a propriedade não seja a mesma.

Pelo mesmo motivo, recomendamos que você altere ou remova o CNAME ou o alias correspondente ao excluir um bucket.

Como associar um nome de host a um bucket do Amazon S3

Como associar um nome de host a um bucket do Amazon S3 usando um alias CNAME

1. Selecione um nome de host que pertença a um domínio controlado por você.

Este exemplo usa o subdomínio `images` do domínio `example.com`.

2. Crie um bucket que corresponda ao nome de host.

Neste exemplo, os nomes de host e do bucket são `images.example.com`. O nome do bucket deve ser exatamente igual ao nome de host.

3. Crie um registro de DNS CNAME que defina o nome de host como um alias para o bucket do Amazon S3.

Por exemplo:

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

Important

Por motivos de encaminhamento de solicitações, o registro de DNS CNAME deve estar definido exatamente como mostrado no exemplo anterior. Do contrário, a operação pode parecer correta, mas acabará causando um comportamento imprevisível.

O procedimento para configurar registros de DNS CNAME 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

O suporte a SOAP via HTTP está obsoleto, mas o SOAP continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com SOAP. Em vez de SOAP, recomendamos usar a API REST ou os AWS SDKs.

Compatibilidade retroativa

As seções a seguir abordam vários aspectos da compatibilidade com versões anteriores do Amazon S3 relacionados a solicitações de URL no estilo de caminho e no estilo de hospedagem virtual.

Endpoints herdados

Algumas regiões oferecem suporte a endpoints legados. Você poderá ver esses endpoints nos logs de acesso ao servidor ou nos logs do AWS CloudTrail. Para obter mais informações, leia as informações

a seguir. Para obter uma lista completa de regiões e endpoints do Amazon S3, consulte [Amazon S3 endpoints and quotas](#) (Endpoints e cotas do Amazon S3) na Amazon Web Services General Reference (Referência geral da Amazon Web Services).

Important

Embora você possa ver endpoints legados nos seus logs, recomendamos que você sempre use a sintaxe do endpoint padrão para acessar os buckets.

Os URLs no estilo de hospedagem virtual do Amazon S3 usam o seguinte formato:

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

No Amazon S3, os URLs no estilo de caminho usam o seguinte formato:

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

s3-Region

Algumas regiões mais antigas do Amazon S3 são compatíveis com endpoints que contêm um traço (-) entre o s3 e a região (por exemplo, s3#us-west-2), em vez de um ponto (por exemplo, s3.us-west-2). Se o bucket estiver em uma dessas regiões, você poderá ver o seguinte formato de endpoint nos logs de acesso ao servidor ou nos logs do CloudTrail:

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

Neste exemplo, o nome do bucket é DOC-EXAMPLE-BUCKET1 e a região é Oeste dos EUA (Oregon):

```
https://DOC-EXAMPLE-BUCKET1.s3-us-west-2.amazonaws.com
```

Endpoint global herdado

Para algumas regiões, é possível usar o endpoint global herdado para criar solicitações que não explicitam um endpoint específico da região. A seguir, o endpoint global herdado:

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

Nos logs de acesso ao servidor ou nos logs do CloudTrail, você poderá ver solicitações que usam o endpoint global herdado. Neste exemplo, o nome do bucket é DOC-EXAMPLE-BUCKET1 e o endpoint global herdado é:

```
https://DOC-EXAMPLE-BUCKET1.s3.amazonaws.com
```

Solicitações no estilo de hospedagem virtual para Leste dos EUA (N. da Virgínia)

Por padrão, as solicitações feitas com o endpoint global herdado vão para a região Leste dos EUA (N. da Virgínia). Portanto, o endpoint global herdado às vezes é usado no lugar do endpoint regional para o Leste dos EUA (Norte da Virgínia). Se você criar um bucket no Leste dos EUA (Norte da Virgínia) e usar o endpoint global, o Amazon S3 roteará sua solicitação para essa região por padrão.

Solicitações no estilo de hospedagem virtual para outras regiões

O endpoint global herdado também é usado para solicitações no estilo de hospedagem virtual em outras regiões compatíveis. Se você criar um bucket em uma região lançada antes de 20 de março de 2019 e usar o endpoint global herdado, o Amazon S3 atualizará o registro DNS para encaminhar novamente a solicitação ao local correto, o que pode levar algum tempo. Enquanto isso, a regra padrão se aplica e sua solicitação no estilo de hospedagem virtual vai para a região Leste dos EUA (N. da Virgínia). Depois, o Amazon S3 a redireciona com um redirecionamento temporário HTTP 307 à região correta.

Para buckets do S3 em regiões lançadas após 20 de março de 2019, o servidor DNS não encaminha a solicitação diretamente à Região da AWS na qual o bucket reside. Em vez disso, ele retorna um erro de solicitação incorreta HTTP 400. Para obter mais informações, consulte [Fazer solicitações \(p. 1211\)](#).

Solicitações no estilo de caminho

Para a região Leste dos EUA (N. da Virgínia), você pode usar o endpoint global herdado para solicitações no estilo de caminho.

Para todas as outras regiões, a sintaxe de caminho exige usar o endpoint específico da região ao tentar acessar um bucket. Se tentar acessar um bucket com o endpoint global herdado ou outro endpoint diferente daquele para a região onde o bucket reside, você receberá um erro de redirecionamento temporário com o código 307 de resposta HTTP e uma mensagem indicando o URI correto para o seu recurso. Por exemplo, se você usar `https://s3.amazonaws.com/bucket-name` para um bucket criado na região Oeste dos EUA (Oregon), você receberá um erro de redirecionamento temporário HTTP 307.

Redirecionamento de solicitação e a API REST

Tópicos

- [Redirecionamentos e agentes de usuário de HTTP \(p. 1255\)](#)
- [Redirecionamentos e 100-Continue \(p. 1255\)](#)
- [Exemplo de redirecionamento \(p. 1256\)](#)

Esta seção descreve como processar redirecionamentos HTTP usando a API REST do Amazon S3. Para obter informações gerais sobre redirecionamentos do Amazon S3, consulte [Fazer solicitações \(p. 1211\)](#) na Referência da API do Amazon Simple Storage Service.

Redirecionamentos e agentes de usuário de HTTP

Os programas que usam a API REST do Amazon S3 devem processar os redirecionamentos na camada de aplicação 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
```

Desenvolvimento com o Amazon S3 usando a AWS CLI

Siga estas etapas para fazer download e configurar a AWS Command Line Interface (AWS CLI).

Para obter uma lista de comandos da AWS CLI do Amazon S3, consulte as seguintes páginas na Referência de comando da AWS CLI:

- [s3](#)
- [s3api](#)
- [s3control](#)

Note

Os serviços na AWS, como o Amazon S3, exigem o fornecimento de credenciais quando acessados. 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 para acessar a AWS CLI ou a API. No entanto, não recomendamos que você acesse a AWS usando as credenciais de sua Conta da AWS. Em vez disso, recomendamos usar o 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 o grupo de administradores](#) no Manual 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 Manual do usuário do AWS Command Line Interface:
 - [Começar a usar a AWS Command Line Interface](#)
 - [Configurar a AWS Command Line Interface](#)
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. Para obter mais informações, consulte [Perfis nomeados da AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

```
[adminuser]
aws_access_key_id = adminuser access key ID
aws_secret_access_key = adminuser secret access key
region = aws-region
```

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

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
```

- Execute um comando `s3` usando as credenciais `adminuser` que você acabou de criar. Para fazer isso, adicione o parâmetro `--profile` ao comando para especificar o nome do perfil. Neste exemplo, o comando `ls` lista os buckets em sua conta. A AWS CLI usa as credenciais `adminuser` para autenticar a solicitação.

```
aws s3 ls --profile adminuser
```

Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers

Você também pode usar os AWS SDKs ao desenvolver aplicações 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 JavaScript do AWS Amplify também estão disponíveis para a compilação de aplicações Web e aplicações para dispositivos móveis conectados usando a AWS.

Esta seção fornece uma visão geral dos AWS SDKs para desenvolver aplicações do Amazon S3. Esta seção também descreve como você pode testar os exemplos de código do AWS SDK fornecidos neste guia.

Tópicos

- [Usar este serviço com um AWS SDK \(p. 1259\)](#)
- [Especificar a versão da assinatura na autenticação de solicitações \(p. 1260\)](#)
- [Usar a AWS SDK for Java \(p. 1266\)](#)
- [Usar a AWS SDK for .NET \(p. 1267\)](#)
- [Usar o AWS SDK for PHP e executar exemplos do PHP \(p. 1268\)](#)
- [Usar o AWS SDK for Ruby - versão 3 \(p. 1269\)](#)
- [Usar a AWS SDK for Python \(Boto\) \(p. 1271\)](#)
- [Usar os AWS Mobile SDKs for iOS e Android \(p. 1271\)](#)
- [Uso da biblioteca JavaScript do AWS Amplify \(p. 1271\)](#)
- [Usar a AWS SDK for JavaScript \(p. 1271\)](#)

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 AWS Command Line Interface (AWS CLI) 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 for Eclipse. O AWS Explorer for Eclipse é um plugin de código aberto para o Eclipse para Java IDE que facilita o desenvolvimento, a depuração e a implantação de aplicações Java para os desenvolvedores que usam a AWS. A interface gráfica fácil de usar permite acessar e administrar a infraestrutura da AWS, incluindo o Amazon S3. É possível executar operações comuns, como gerenciar buckets e objetos, além de definir políticas do IAM ao mesmo tempo que desenvolve aplicações, tudo no contexto do IDE do Eclipse for Java. Para obter instruções de configuração, consulte [Configurar o toolkit](#). Para obter exemplos de como usar o Explorer, consulte [Como acessar o AWS Explorer](#).

AWS Toolkit for Visual Studio

AWS Explorer para Visual Studio é uma extensão do Microsoft Visual Studio que facilita o desenvolvimento, a depuração e a implantação de aplicações .NET para os desenvolvedores que usam a Amazon Web Services. A interface gráfica fácil de usar permite acessar e administrar a infraestrutura da AWS, incluindo o Amazon S3. É possível executar operações comuns, como gerenciar buckets e objetos ou definir políticas do IAM ao desenvolver aplicações, tudo no contexto do Visual Studio. Para obter instruções de configuração, visite [Configurar o AWS Toolkit for Visual Studio](#). Para obter exemplos de como usar o Amazon S3 com o Explorer, consulte [Uso do Amazon S3 com o AWS Explorer](#).

AWSSDKs da

Você pode fazer download somente dos SDKs. Para obter informações sobre como fazer 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](#).

Usar este serviço com um AWS SDK

Os kits de desenvolvimento de software (SDKs) da AWS estão disponíveis para muitas linguagens de programação populares. Cada SDK fornece uma API, exemplos de código e documentação que facilitam a criação de aplicações em seu idioma preferido pelos desenvolvedores.

Documentação do SDK	Exemplos de código
AWS SDK for C++	Exemplos de código AWS SDK for C++
AWS SDK for Go	Exemplos de código AWS SDK for Go
AWS SDK for Java	Exemplos de código AWS SDK for Java
AWS SDK for JavaScript	Exemplos de código AWS SDK for JavaScript
AWS SDK for Kotlin	Exemplos de código AWS SDK for Kotlin
AWS SDK for .NET	Exemplos de código AWS SDK for .NET
AWS SDK for PHP	Exemplos de código AWS SDK for PHP
AWS SDK for Python (Boto3)	Exemplos de código AWS SDK for Python (Boto3)
AWS SDK for Ruby	Exemplos de código AWS SDK for Ruby
AWS SDK for Rust	Exemplos de código AWS SDK for Rust
AWS SDK for Swift	Exemplos de código AWS SDK for Swift

Exemplo de disponibilidade

Você não consegue encontrar o que precisa? Solicite um código de exemplo no link Forneça comentários na parte inferior desta página.

Especificar a versão da assinatura na autenticação de solicitações

O Amazon S3 oferece suporte apenas ao AWS Signature Version 4 na maioria das Regiões da AWS. No entanto, em algumas das Regiões da AWS mais antigas, o Amazon S3 é compatível com o Signature Version 4 e o Signature Version 2. No entanto, o Signature versão 2 está sendo desativado (defasado). Para obter mais informações sobre o fim do suporte ao Signature versão 2, consulte [AWS Signature versão 2 desativado \(defasado\) para o Amazon S3 \(p. 1261\)](#).

Para obter uma lista de todas as regiões do Amazon S3 e das versões do Signature com as quais elas são compatíveis, consulte [Regiões e endpoints](#) na Referência geral da AWS.

Para todas as Regiões da AWS, por padrão, os AWS SDKs usam o Signature Version 4 para autenticar solicitações. Ao usar os AWS SDKs 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
AWS CLI	<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 para o SDK v2 do PHP:</p> <pre><?php \$client = S3Client::factory(['region' => 'YOUR-REGION', 'version' => 'latest', 'signature' => 'v4']);</pre>

SDK	Solicitar o Signature versão 4 para autenticação de solicitações
	<p>Ao usar o SDK v3 do PHP, defina o parâmetro <code>signature_version</code> como <code>v4</code> durante a criação do cliente de serviço do Amazon S3:</p> <pre><?php \$s3 = new Aws\S3\S3Client(['version' => '2006-03-01', 'region' => 'YOUR-REGION', 'signature_version' => 'v4']);</pre>
SDK do Python-Boto	Especifique o seguinte no arquivo de configuração boto padrão:
	<pre>[s3] use-sigv4 = True</pre>
SDK do Ruby	<p>SDK do Ruby - versão 1: defina o parâmetro <code>:s3_signature_version</code> como <code>:v4</code> ao criar o cliente:</p> <pre>s3 = AWS::S3::Client.new(:s3_signature_version => :v4)</pre> <p>SDK do Ruby - versão 3: defina o parâmetro <code>signature_version</code> como <code>v4</code> ao criar o cliente:</p> <pre>s3 = Aws::S3::Client.new(signature_version: 'v4')</pre>
SDK do .NET	<p>Adicione o código a seguir antes de criar o cliente do Amazon S3:</p> <pre>AWSConfig.S3.UseSignatureVersion4 = true;</pre> <p>Ou adicione o seguinte ao arquivo config:</p> <pre><appSettings> <add key="AWS.S3.UseSignatureVersion4" value="true" /> </appSettings></pre>

AWS Signature versão 2 desativado (defasado) para o Amazon S3

O Signature versão 2 está sendo desativado (defasado) no Amazon S3. 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. A assinatura de solicitações permite que o Amazon S3 identifique quem está enviando uma solicitação e a protege contra agentes mal intencionados.

Para obter mais informações sobre a assinatura de solicitações da AWS, consulte [Assinatura de solicitações de API da AWS](#) na Referência geral da AWS.

Qual atualização está sendo feita?

No momento, oferecemos suporte a solicitações da API do Amazon S3 assinadas usando os processos das versões 2 e 4 do Signature. Depois disso, o Amazon S3 só aceitará solicitações assinadas com o Signature versão 4.

Para obter mais informações sobre a assinatura de solicitações da AWS, consulte [Alterações no Signature Version 4](#) na Referência geral da AWS.

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 Version 4 é compatível com todas as Regiões da AWS. O Signature Version 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 AWS SDKs usam o Signature Version 4. Para software de terceiro, entre em contato com a equipe de suporte dele para confirmar a versão necessária. Para enviar chamadas REST diretamente para o Amazon S3, modifique a aplicação para usar o processo de assinatura do Signature versão 4.

Para obter informações sobre qual versão dos AWS SDKs deve ser usada depois da transição para o Signature Version 4, consulte [Migração do Signature versão 2 para o Signature versão 4 \(p. 1263\)](#).

Para obter informações sobre como usar o Signature Version 4 com a API REST do Amazon S3, consulte [Autenticação de solicitações \(AWS Signature Version 4\)](#) na Referência da API do Amazon Simple Storage Service.

O que acontecerá se eu não atualizar?

As solicitações assinadas com o Signature versão 2 feitas após essa data não autenticarão 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 Version 2 para assinar e autenticar URLs pré-assinados. 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](#).
- Visualize a lista de alterações entre o Signature versão 2 e o Signature versão 4 em [Alterações no Signature versão 4](#).
- Consulte a publicação [AWS Signature Version 4 como substituição ao AWS Signature Version 2 para assinar solicitações da API do Amazon S3](#) nos fóruns da AWS.
- Em caso de dúvidas ou preocupações, entre em contato conosco em [AWS Support](#).

Migração do Signature versão 2 para o Signature versão 4

Se estiver usando o Signature versão 2 para autenticar solicitações de API do Amazon S3, migre para o Signature versão 4. Não haverá mais suporte para o Signature versão 2, conforme descrito em [AWS Signature versão 2 desativado \(defasado\) para o Amazon S3 \(p. 1261\)](#).

Para obter informações sobre como usar o Signature Version 4 com a API REST do Amazon S3, consulte [Autenticação de solicitações \(AWS Signature Version 4\)](#) na Referência da API do Amazon Simple Storage Service.

A tabela a seguir contém os SDKs que exigem a utilização do Signature versão 4 (SigV4). Se você utiliza URLs pré-assinados 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 [Especificar a versão da assinatura na autenticação de solicitações \(p. 1260\)](#).

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	Atualize para Java 1.11.201+ ou v2 no 4º trimestre de 2018.	Sim	Especificar a versão da assinatura na autenticação de solicitações (p. 1260)
AWS SDK for Java v2 (pré-visualização)	Não é necessário atualizar os SDKs.	Não	AWS SDK for Java
AWS SDK for .NET v1	Atualize para o 3.1.10 ou versão superior.	Sim	AWS SDK for .NET
AWS SDK for .NET v2	Atualize para o 3.1.10 ou versão superior.	Não	AWS SDK for .NET v2
AWS SDK for .NET v3	Atualize para a versão 3.3.0.0 ou superior.	Sim	AWS SDK for .NET v3
AWS SDK for JavaScript v1	Atualize para o 2.68.0 ou versão superior.	Sim	AWS SDK for JavaScript
AWS SDK for JavaScript v2	Atualize para o 2.68.0 ou versão superior.	Sim	AWS SDK for JavaScript
AWS SDK for JavaScript v3	Nenhuma outra ação é	Não	AWS SDK for JavaScript

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 for PHP v1	Recomenda-se atualizar para a versão mais recente do PHP ou, ao menos, para a versão 2.7.4 com o parâmetro de assinatura definido como v4 na configuração do cliente do S3.	Sim	AWS SDK for PHP
AWS SDK for PHP v2	Recomenda-se atualizar para a versão mais recente do PHP ou, ao menos, para a versão 2.7.4 com o parâmetro de assinatura definido como v4 na configuração do cliente do S3.	Não	AWS SDK for PHP
AWS SDK for PHP v3	Não é necessário atualizar os SDKs.	Não	AWS SDK for PHP
Boto2	Atualize para Boto2 v2.49.0.	Sim	Atualização do Boto 2
Boto3	Atualize para 1.5.71 (BotoCore), 1.4.6 (Boto3).	Sim	Boto 3 - AWS SDK para Python

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 CLI	Atualize para 1.11.108.	Sim	AWS Command Line Interface
AWS CLI v2 (pré-visualização)	Não é necessário atualizar os SDKs.	Não	AWS Command Line Interface versão 2
AWS SDK for Ruby v1	Atualize para Ruby V3.	Sim	Ruby V3 for AWS
AWS SDK for Ruby v2	Atualize para Ruby V3.	Sim	Ruby V3 for AWS
AWS SDK for Ruby v3	Não é necessário atualizar os SDKs.	Não	Ruby V3 for AWS
Go	Não é necessário atualizar os SDKs.	Não	AWS SDK for Go
C++	Não é necessário atualizar os SDKs.	Não	AWS SDK for C++

AWS Tools for Windows PowerShell ou AWS Tools for PowerShell Core

Se você estiver usando versões de módulo anteriores à 3.3.0.0, deverá atualizar para a 3.3.0.0.

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 3.3.0.0, 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.

Usar a AWS SDK for Java

O AWS SDK for Java fornece uma API para as operações de buckets e objetos 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 [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).

Tópicos

- [A organização da API Java \(p. 1266\)](#)
- [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#)

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 REST subjacentes do Amazon S3, 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 streams 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 de dados da API de alto nível.

Essa API de alto nível é projetada para simplificar a transferência de objetos para o Amazon S3 e vice-versa. 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 a API do AWS SDK for Java, consulte a [Referência da API do AWS SDK for Java](#).

Testar exemplos de código Java no 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](#) no Guia do desenvolvedor do AWS SDK for Java.

Usar a AWS SDK for .NET

O AWS SDK for .NET fornece a API para as operações de buckets e objetos 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 [Carregar e copiar objetos usando multipart upload \(p. 180\)](#)).

Tópicos

- [A organização da API .NET \(p. 1268\)](#)
- [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#)

O AWS SDK for .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 REST subjacentes do Amazon S3 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 [Carregar e copiar objetos usando multipart upload \(p. 180\)](#)), 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 streams 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 aplicações do Amazon S3 usando o AWS SDK for .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 REST multipart upload do Amazon S3 (consulte [Uso dos REST API \(p. 200\)](#)).

- `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 o `AmazonS3EncryptionClient`.
- `Amazon.S3.Util`: fornece várias classes de utilitários, como `AmazonS3Util` e `BucketRegionDetector`.

Para obter mais informações sobre a API do AWS SDK for .NET, consulte a [Referência da API do AWS SDK for .NET Version 3](#).

Executar os exemplos de código do Amazon S3 .NET

Os exemplos de código .NET neste guia são compatíveis com o AWS SDK for .NET versão 3.0. Para obter informações sobre como configurar e executar exemplos de código, consulte [Conceitos básicos do AWS SDK for .NET](#) no Guia do desenvolvedor do AWS SDK for .NET.

Usar o AWS SDK for PHP e executar exemplos do PHP

O AWS SDK for PHP fornece acesso à API para operações de buckets e objetos 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 for PHP](#), que também tem instruções para instalar e começar a usar o SDK.

A configuração para usar o AWS SDK for PHP depende do ambiente e de como você deseja executar seu aplicativo. Para configurar o ambiente para executar os exemplos desta documentação, consulte o [Guia de conceitos básicos do AWS SDK for PHP](#).

Tópicos

- [AWS SDK for PHP Níveis do \(p. 1269\)](#)
- [Executar exemplos do PHP \(p. 1269\)](#)
- [Recursos relacionados \(p. 1269\)](#)

AWS SDK for PHP Níveis do

O AWS SDK for 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 REST subjacentes do Amazon S3 incluindo operações de criação, atualização e exclusão em buckets e objetos. As APIs de baixo nível fornecem maior controle sobre essas operações. Por exemplo, é possível colocar as 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 [Carregar e copiar objetos usando multipart upload \(p. 180\)](#).

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 for 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

Para configurar e usar os exemplos do Amazon S3 para a versão 3 do AWS SDK for PHP, consulte [Instalação](#) no Guia do desenvolvedor do AWS SDK for PHP.

Recursos relacionados

- [AWS SDK for PHP for Amazon S3](#)
- [Documentação do AWS SDK para PHP](#)
- [AWS SDK para PHP API for Amazon S3](#)
- [Exemplos de código do AWS SDK para PHP versão 3](#)

Usar o AWS SDK for Ruby - versão 3

O AWS SDK for Ruby fornece uma API para as operações de buckets e objetos 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 [Fazer upload de um objeto usando multipart upload \(p. 187\)](#)). 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 aplicações do Amazon S3 usando o AWS SDK for Ruby SDK for Ruby, instale o gem SDK for Ruby. Para obter mais informações, consulte [AWS SDK for Ruby Version 3](#). Depois de instalado, você pode acessar a API, incluindo as seguintes classes de chaves:

- Aws::S3::Resource: representa a interface para o 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 a API do AWS SDK for Ruby, acesse [AWS SDK for Ruby Version 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 for Ruby gem. Para obter informações sobre como instalar ou atualizar a gem mais recente, acesse [AWS SDK for Ruby Version 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 for 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 sua aplicação do SDK for Ruby. Para obter mais informações, consulte Configuração do AWS SDK for Ruby .
2	Crie um script do SDK para Ruby 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 require 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 AWS Management Console. Para obter mais informações sobre o AWS Management Console, acesse https://aws.amazon.com/console/ .

Exemplos do Ruby

Os links a seguir contêm exemplos para ajudar você a começar a usar o SDK para Ruby versão 3:

- [Criação de um bucket \(p. 129\)](#)

- [Fazer upload de objetos \(p. 171\)](#)

Usar a 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 for iOS e Android

Você pode usar os AWS Mobile SDKs for [Android](#) e [iOS](#) para integrar de maneira rápida e fácil back-ends de nuvem robustos aos suas aplicações 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

[Uso da biblioteca JavaScript do AWS Amplify \(p. 1271\)](#)

Uso da biblioteca JavaScript do AWS Amplify

O AWS Amplify é uma biblioteca de JavaScript de código aberto para desenvolvedores para web e dispositivos móveis que compilam aplicações compatíveis com a nuvem. AWS OAWS 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 .

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 do Amplify](#)

Para obter mais informações sobre o AWS Amplify, consulte [AWS Amplify no GitHub](#).

Mais informações

[Usar os AWS Mobile SDKs for iOS e Android \(p. 1271\)](#)

Usar a AWS SDK for JavaScript

O AWS SDK for JavaScript fornece uma API JavaScript para serviços da AWS. Você pode usar a API JavaScript para criar bibliotecas ou aplicativos para Node.js ou o navegador.

Para obter mais informações sobre como usar o AWS SDK for JavaScript para Amazon S3, consulte abaixo.

- [O que é o AWS SDK for JavaScript? \(v2\)](#)
- [AWS SDK for JavaScript: exemplos do Amazon S3 \(v2\)](#)
- [O que é o AWS SDK for JavaScript? \(v3\)](#)
- [AWS SDK for JavaScript: exemplos do Amazon S3 \(v3\)](#)

- [AWS SDK for JavaScript for Amazon S3](#)

Desenvolver com o Amazon S3 usando a API REST

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

No momento, o Amazon S3 fornece uma interface REST. Com a 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. A API REST é uma interface HTTP para o Amazon S3. Usando REST, você usa solicitações HTTP padrão 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 correspondesse ao estilo de uso padrão do HTTP.

Para obter mais informações sobre envio de e-mail usando a API REST, consulte [Fazer solicitações usando a API REST \(p. 1247\)](#). Sobre algumas considerações que você deve ter em mente ao usar a API REST, consulte os tópicos abaixo.

Tópicos

- [Roteamento de solicitação \(p. 1272\)](#)

Roteamento de solicitação

Programas que fazem solicitações em buckets criados usando a API [CreateBucket](#) e incluem uma [CreateBucketConfiguration](#) devem ser compatíveis com 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 aplicação 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 uma aplicação 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 [Métodos de acesso a um bucket \(p. 135\)](#). Para todas as regiões lançadas após 20 de março de 2019, se uma solicitação chega no local errado do Amazon S3, o Amazon S3 retorna um erro de solicitação incorreta HTTP 400.

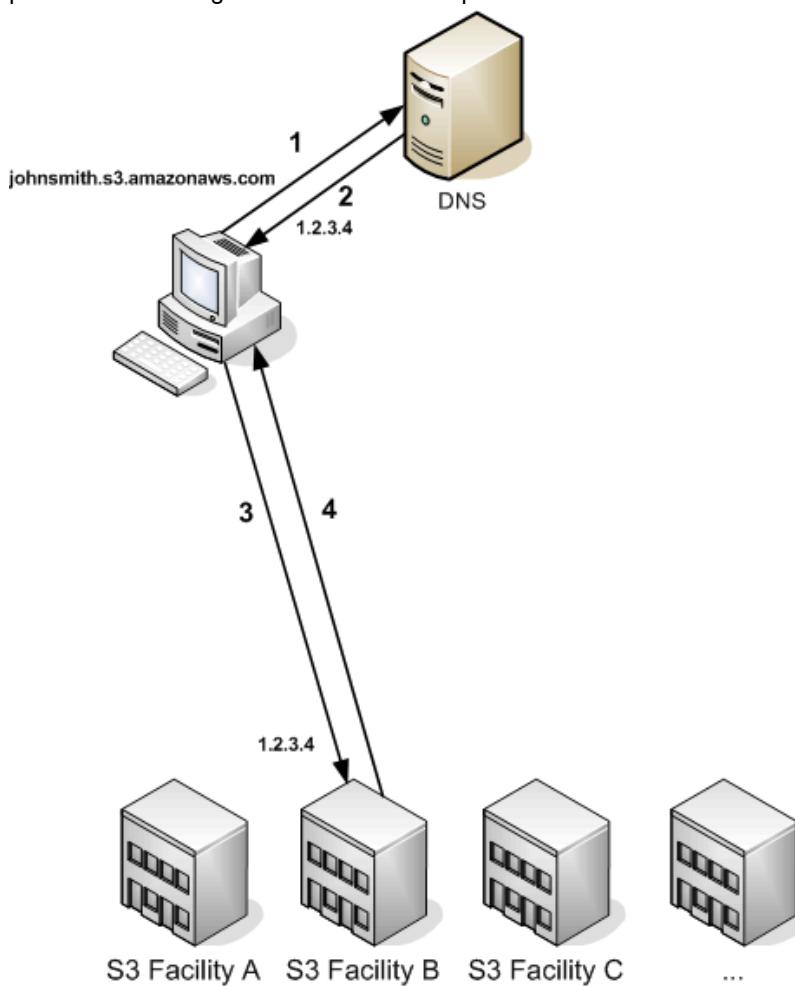
Para obter mais informações sobre como ativar ou desativar uma Região da AWS, consulte [Regiões da AWS e endpoints](#) na Referência geral da AWS.

Tópicos

- [Roteamento de DNS \(p. 1273\)](#)
- [Redirecionamento de solicitação temporário \(p. 1274\)](#)
- [Redirecionamento permanente de solicitação \(p. 1275\)](#)
- [Exemplos de redirecionamento de solicitação \(p. 1275\)](#)

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.

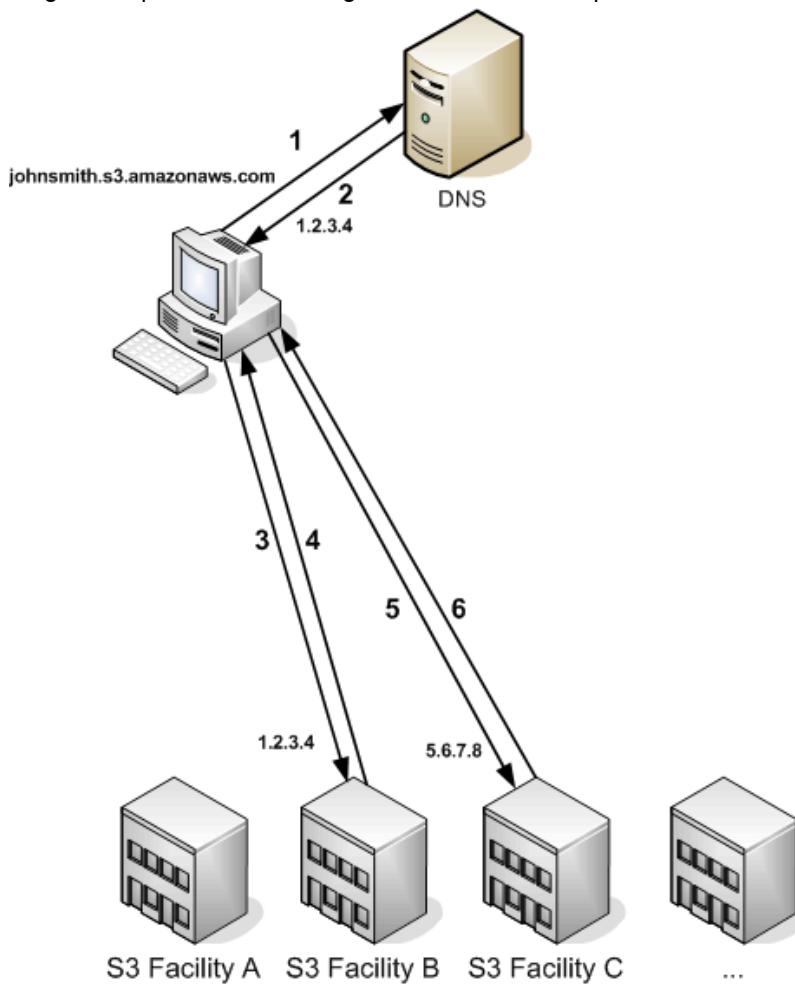
Por exemplo, se você criar um bucket novo e, em seguida, fizer uma solicitação a esse bucket, é possível que receba um redirecionamento temporário dependendo da restrição de localização do bucket. Se você criou o bucket na Região da AWS Leste dos EUA (Norte da Virgínia), não verá o redirecionamento, 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 [Métodos de acesso a um bucket \(p. 135\)](#).

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://awsexamplebucket1.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>awsexamplebucket1.s3-gztb4pa9sq.amazonaws.com</Endpoint>
</Error>
```

API SOAP de resposta de redirecionamento temporário

Note

O suporte a SOAP via HTTP está obsoleto, mas o SOAP continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com SOAP. Em vez de SOAP, recomendamos usar a API REST ou os AWS SDKs.

```
<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 de cumprirmos 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](#).

Tratar erros de REST e SOAP

Tópicos

- [A resposta de erro de REST \(p. 1276\)](#)
- [A resposta de erro de SOAP \(p. 1278\)](#)
- [Melhores práticas com relação a erros do Amazon S3 \(p. 1278\)](#)

Esta seção descreve os erros de REST e de SOAP e como tratá-los.

Note

O suporte a SOAP via HTTP está obsoleto, mas o SOAP continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com SOAP. Em vez de SOAP, recomendamos usar a API REST ou os AWS SDKs.

A resposta de erro de REST

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

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 a SOAP via HTTP está obsoleto, mas o SOAP continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com SOAP. Em vez de SOAP, recomendamos usar a API REST ou os AWS SDKs.

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 a SOAP via HTTP está obsoleto, mas o SOAP continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com SOAP. Em vez de SOAP, recomendamos usar a API REST ou os AWS SDKs.

A 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 a SOAP via HTTP está obsoleto, mas o SOAP continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com SOAP. Em vez de SOAP, recomendamos usar a API REST ou os AWS SDKs.

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 ao 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 com relação a erros do Amazon S3

Ao desenvolver uma aplicação 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.

Ajustar 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. No entanto, se você quiser obter mais informações ou

estiver enfrentando erros de desaceleração em grande quantidade ou inesperados, publique no nosso fórum de desenvolvedores do Amazon S3 ou cadastre-se no AWS Support <https://aws.amazon.com/premiumsupport/>.

Erros isolados

Note

O suporte a SOAP via HTTP está obsoleto, mas o SOAP continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com SOAP. Em vez de SOAP, recomendamos usar a API REST ou os AWS SDKs.

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 para 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.

Referência do desenvolvedor

Este apêndice inclui as seções a seguir.

Tópicos

- [Apêndice A: Usar a API SOAP \(p. 1279\)](#)
- [Apêndice B: Autenticação de solicitações \(AWS Signature Version 2\) \(p. 1282\)](#)

Apêndice A: Usar a API SOAP

Note

O suporte a SOAP via HTTP está obsoleto, mas o SOAP continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com SOAP. Em vez de SOAP, recomendamos usar a API REST ou os AWS SDKs.

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.

Tópicos

- [Elementos comuns da API SOAP \(p. 1280\)](#)

- Autenticar solicitações SOAP (p. 1280)
- Configurar políticas de acesso padrão com SOAP (p. 1281)

Elementos comuns da API SOAP

Note

O suporte a SOAP via HTTP está obsoleto, mas o SOAP continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com SOAP. Em vez de SOAP, recomendamos usar a API REST ou os AWS SDKs.

Você pode interagir com o Amazon S3 usando SOAP 1.1 via HTTP. O WSDL do Amazon S3, que descreve a API do Amazon S3 de maneira legível por máquina, está disponível em: <https://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl>. O esquema do Amazon S3 está disponível em <https://doc.s3.amazonaws.com/2006-03-01/AmazonS3.xsd>.

A maioria dos usuários vai interagir 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 do 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 da chave de acesso da AWS do solicitante
- **Timestamp**: A hora atual do seu sistema
- **Signature**: A assinatura da solicitação

Autenticar solicitações SOAP

Note

O suporte a SOAP via HTTP está obsoleto, mas o SOAP continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com SOAP. Em vez de SOAP, recomendamos usar a API REST ou os AWS SDKs.

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 da 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 \(p. 1211\)](#).

- **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. Haverá falha na autorização se esse time stamp tiver mais de 15 minutos de diferença do relógio nos servidores do Amazon S3.
- **Signature**: O resumo da RFC 2104 HMAC-SHA1 (acesse <http://www.ietf.org/rfc/rfc2104.txt>) da concatenação de "AmazonS3" + OPERAÇÃO + carimbo de data/hora, 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="https://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 datas e horas excessivamente específicas ao Amazon S3. Isso pode ser realizado criando manualmente objetos `DateTime` com precisão de apenas milissegundos.

Configurar políticas de acesso padrão com SOAP

Note

O suporte a SOAP via HTTP está obsoleto, mas o SOAP continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com SOAP. Em vez de SOAP, recomendamos usar a API REST ou os AWS SDKs.

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 [Identity and Access Management no Amazon S3 \(p. 419\)](#). 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="https://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>75cc57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</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="https://s3.amazonaws.com/doc/2006-03-01">
  <PutObjectInlineResponse>
    <ETag>"828ef3fd96f00ad9f27c383fc9ac7f"</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 (AWS Signature Version 2)

Important

Esta seção descreve como autenticar solicitações usando o AWS Signature Version 2. O Signature versão 2 está sendo desativado (defasado), o Amazon S3 aceitará somente solicitações de API assinadas usando o Signature versão 4. Para obter mais informações, consulte [AWS Signature versão 2 desativado \(defasado\) para o Amazon S3 \(p. 1261\)](#)

O Signature Version 4 é compatível com todas as Regiões da AWS. Essa é a única versão compatível com novas regiões. Para obter mais informações, consulte [Autenticação de solicitações \(AWS Signature Version 4\)](#) na Referência de APIs do Amazon Simple Storage Service.

O Amazon S3 oferece a possibilidade de identificar qual versão da assinatura de API foi usada para assinar uma solicitação. É importante identificar se algum dos fluxos de trabalho estão utilizando as assinaturas do Signature versão 2 e atualizá-los para que usem o Signature versão 4 para evitar que seus negócios sejam impactados.

- Se você estiver usando os logs de eventos do CloudTrail (opção recomendada), consulte [Identificar solicitações do Amazon S3 Signature versão 2 usando o CloudTrail \(p. 1037\)](#) sobre como consultar e identificar essas solicitações.

- Se você estiver usando os logs de acesso ao servidor do Amazon S3, consulte [Identificar solicitações do Signature versão 2 usando logs de acesso do Amazon S3 \(p. 1066\)](#)

Tópicos

- [Autenticar solicitações usando a API REST \(p. 1284\)](#)
- [Assinar e autenticar as solicitações REST \(p. 1286\)](#)
- [Uploads baseados no navegador usando POST \(AWS Signature Version 2\) \(p. 1296\)](#)

Autenticar solicitações usando a API REST

Ao acessar o Amazon S3 usando REST, é necessário fornecer os seguintes itens na solicitação para que ela seja autenticada:

Elementos da solicitação

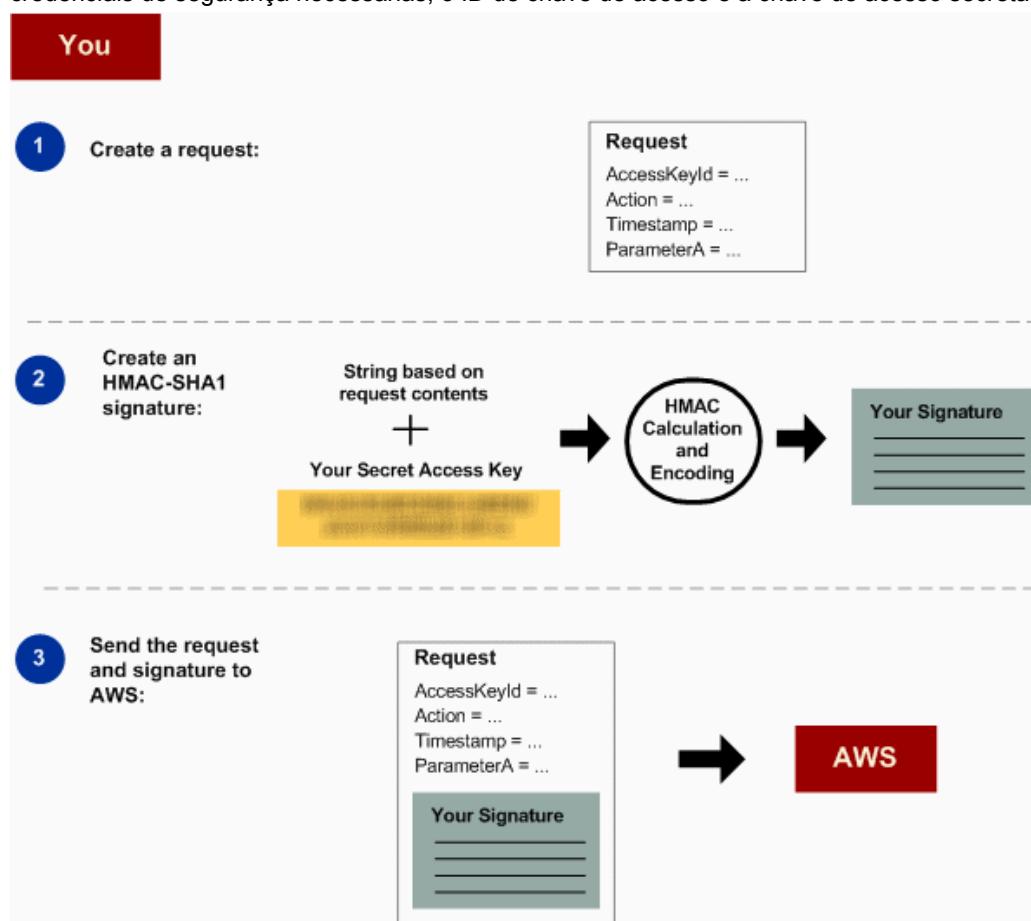
- ID da chave de acesso da AWS: cada solicitação deve conter o ID da chave de acesso da identidade que você estiver usando para enviar a solicitação.
- Assinatura: cada solicitação deve conter uma assinatura de solicitação válida. Caso contrário, a solicitação será rejeitada.

A assinatura de uma solicitação é calculada com a chave de acesso secreta, um segredo compartilhado conhecido apenas por você e

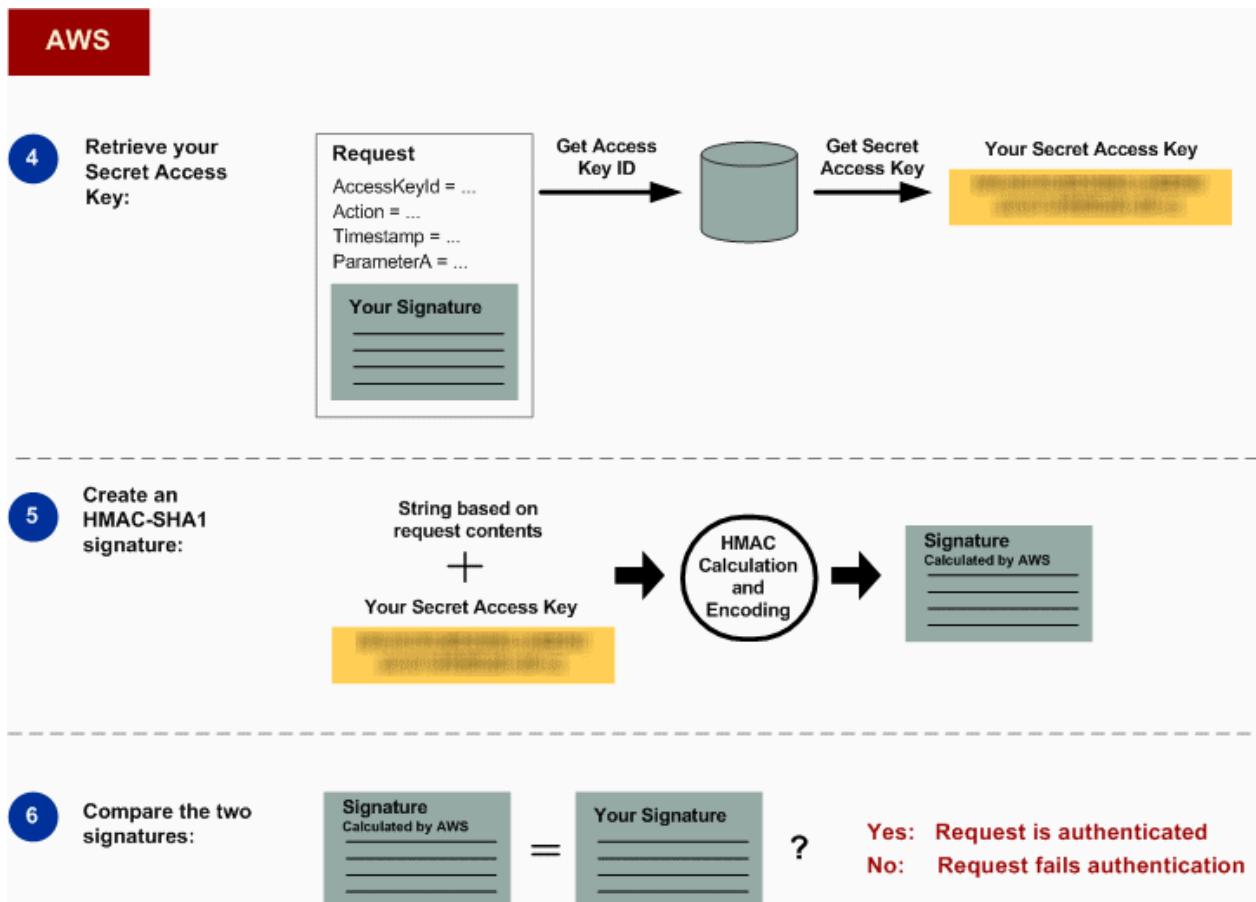
- Time stamp: cada solicitação deve conter a data e a hora de criação da solicitação, representada como uma string em UTC.
- Data: cada solicitação deve conter o time stamp da solicitação.

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 ao 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
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 executa as próximas três etapas.



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 será considerada autêntica. Se a houver falha na comparação, 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 \(p. 1286\)](#).

Assinar e autenticar as solicitações REST

Tópicos

- [Uso de credenciais de segurança temporárias \(p. 1287\)](#)
- [Cabeçalho de autenticação \(p. 1287\)](#)
- [Canonização de solicitação para assinatura \(p. 1288\)](#)
- [Criar o elemento CanonicalizedResource \(p. 1288\)](#)
- [Criar o elemento CanonicalizedAmzHeaders \(p. 1289\)](#)
- [Elementos StringToSign de cabeçalho HTTP posicionais versus nomeados \(p. 1289\)](#)
- [Requisito de time stamp \(p. 1290\)](#)
- [Exemplos de autenticação \(p. 1290\)](#)
- [Problemas de assinatura de solicitação REST \(p. 1294\)](#)
- [Alternativa de autenticação de solicitação por string de consulta \(p. 1294\)](#)

Note

Este tópico explica solicitações de autenticação usando o Signature versão 2. O Amazon S3 agora é compatível com o mais recente Signature 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, acesse [Autenticação de solicitações \(AWS Signature versão 4\)](#) na Referência de APIs do Amazon Simple Storage Service.

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 \(AWS Signature Version 2\) \(p. 1296\)](#).

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 há correspondência entre as duas assinaturas, o sistema concluirá 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 autenticada do Amazon S3

```
GET /photos/puppy.jpg HTTP/1.1
```

```
Host: awsexamplebucket1.us-west-1.s3.amazonaws.com
Date: Tue, 27 Mar 2007 19:36:42 +0000

Authorization: AWS AKIAIOSFODNN7EXAMPLE:
qgk2+6Sv9/oM7G3qLEjTH1a11ig=
```

Uso de credenciais de segurança temporárias

Se você assinar sua solicitação usando credenciais de segurança temporárias (consulte [Fazer solicitações \(p. 1211\)](#)), 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. Forneça o valor do token de sessão no cabeçalho `x-amz-security-token` ao enviar solicitações ao Amazon S3. Para obter informações sobre a API do AWS Security Token Service fornecida pelo IAM, acesse [Ação](#) no Guia de referência da API do AWS Security Token Service.

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 da 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 elemento `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( UTF-8-Encoding-Of(YourSecretAccessKey), 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", or "?logging"];

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 (`YourSecretAccessKey`) como a chave e a codificação UTF-8 do elemento `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, é necessário 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 Hospedagem virtual de buckets (p. 1249).</p> <p>Para uma solicitação no estilo de hospedagem virtual "https://awsexamplebucket1.s3.us-west-1.amazonaws.com/photos/puppy.jpg", o <code>CanonicalizedResource</code> é "/awsexamplebucket1".</p> <p>Para uma solicitação em estilo de caminho, "https://s3.us-west-1.amazonaws.com/awsexamplebucket1/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 no estilo de hospedagem virtual "https://awsexamplebucket1.s3.us-west-1.amazonaws.com/photos/puppy.jpg", o <code>CanonicalizedResource</code> é "/awsexamplebucket1/photos/puppy.jpg".</p> <p>Para uma solicitação no estilo de hospedagem virtual "https://s3.us-west-1.amazonaws.com/awsexamplebucket1/photos/puppy.jpg", o <code>CanonicalizedResource</code> é "/awsexamplebucket1/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 GET Service, adicione "/".</p>
4	<p>Se a solicitação endereça um sub-recurso, como <code>?versioning</code>, <code>?location</code>, <code>?acl</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 '&', por exemplo, <code>?acl&versionId=value</code>.</p> <p>Os sub-recursos que devem ser incluídos ao criar o elemento <code>CanonicalizedResource</code> são <code>acl</code>, <code>lifecycle</code>, <code>location</code>, <code>logging</code>, <code>notification</code>, <code>partNumber</code>, <code>policy</code>, <code>requestPayment</code>, <code>uploadId</code>, <code>uploads</code>, <code>versionId</code>, <code>versioning</code>, <code>versions</code> e <code>website</code>.</p>

Se a solicitação especificar os parâmetros de query string que cancelam os valores de cabeçalho da 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 \(p. 1249\)](#).

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 do cabeçalho com o mesmo nome em um par "header-name:comma-separated-value-list" conforme descrito em 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 em RFC 2616, seção 4.2) substituindo o espaço de dobramento (incluindo a nova linha) por um único espaço.
5	Remova todos os espaços 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>U+000A</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 positional. `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 string de consulta \(p. 1294\)](#).

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 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 awsexamplebucket1.

Solicitação	StringToSign
<pre>GET /photos/puppy.jpg HTTP/1.1 Host: awsexamplebucket1.us- west-1.s3.amazonaws.com Date: Tue, 27 Mar 2007 19:36:42 +0000 Authorization: AWS AKIAIOSFODNN7EXAMPLE: ggk2+6Sv9/oM7G3qLEjTH1allig=</pre>	<pre>GET\n \n \n Tue, 27 Mar 2007 19:36:42 +0000\n /awsexamplebucket1/photos/puppy.jpg</pre>

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.)

Note

O script Python a seguir calcula a assinatura precedente, usando os parâmetros fornecidos. É possível usar esse script para criar suas próprias assinaturas, substituindo as chaves e StringToSign conforme apropriado.

```
import base64
import hmac
from hashlib import sha1

access_key = 'AKIAIOSFODNN7EXAMPLE'.encode("UTF-8")
secret_key = 'wJalrXUtnFEMI/K7MDENG/bPxRfCYEXAMPLEKEY'.encode("UTF-8")

string_to_sign = 'GET\n\n\nTue, 27 Mar 2007 19:36:42 +0000\n/awsexamplebucket1/\nphotos/puppy.jpg'.encode("UTF-8")
signature = base64.encodestring(
    hmac.new(
        secret_key, string_to_sign, sha1
    ).digest()
).strip()

print(f"AWS {access_key.decode()}:{signature.decode()}")
```

PUT objeto

Este exemplo coloca um objeto no bucket awsexamplebucket1 bucket.

Solicitação	StringToSign
PUT /photos/puppy.jpg HTTP/1.1 Content-Type: image/jpeg Content-Length: 94328 Host: awsexamplebucket1.s3.us-west-1.amazonaws.com Date: Tue, 27 Mar 2007 21:15:45 +0000 <i>Authorization: AWS AKIAIOSFODNN7EXAMPLE: iqRzw+ileNPu1fhspnRs8nOjjIA=</i>	PUT\n\nimage/jpeg\nTue, 27 Mar 2007 21:15:45 +0000\n/awsexamplebucket1/photos/puppy.jpg

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.

List

Este exemplo lista o conteúdo do bucket awsexamplebucket1.

Solicitação	StringToSign
GET /?prefix=photos&max-keys=50&marker=puppy HTTP/1.1 User-Agent: Mozilla/5.0 Host: awsexamplebucket1.s3.us-west-1.amazonaws.com Date: Tue, 27 Mar 2007 19:42:41 +0000 <i>Authorization: AWS AKIAIOSFODNN7EXAMPLE:</i>	GET\n\n\nTue, 27 Mar 2007 19:42:41 +0000\n

Solicitação	StringToSign
<code>m0WP8eCtspQl5Ahe6L1SozdX9YA=</code>	/awsexamplebucket1/

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 “awsexamplebucket1” bucket.

Solicitação	StringToSign
<pre>GET /?acl HTTP/1.1 Host: awsexamplebucket1.s3.us-west-1.amazonaws.com Date: Tue, 27 Mar 2007 19:44:46 +0000 Authorization: AWS AKIAIOSFODNN7EXAMPLE: 82ZHiFIjc+WbcwFKGUVEQspPn+0=</pre>	<pre>GET\n \n \n Tue, 27 Mar 2007 19:44:46 +0000\n /awsexamplebucket1/?acl</pre>

Observe como o parâmetro de query string do sub-recurso está incluído no CanonicalizedResource.

Excluir

Este exemplo exclui um objeto do bucket “awsexamplebucket1” usando o estilo de caminho e a alternativa de data.

Solicitação	StringToSign
<pre>DELETE /awsexamplebucket1/photos/puppy.jpg HTTP/1.1 User-Agent: dotnet Host: s3.us-west-1.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:XbyTlbQdu9Xw5o8P4iMwPktxQd8=</pre>	<pre>DELETE\n \n \n Tue, 27 Mar 2007 21:20:26 +0000\n /awsexamplebucket1/photos/puppy.jpg</pre>

Observe como usamos o método alternativo 'x-amz-data' de especificar a data (porque a biblioteca de cliente nos impediu 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.example.com:8080</pre>	<pre>PUT\n 4gJE4saaMU4BqNR0kLY+lw==\n application/x-download\n</pre>

Solicitação	StringToSign
<pre>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@example.com X-Amz-Meta-ReviewedBy: jane@example.com 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: dKZcB+bz2EPXgSdXZp9ozGeOM4I=</pre>	<pre>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@example.com,jane@example.com\n /static.example.com/db-backup.dat.gz</pre>

Observe como os cabeçalhos 'x-amz-' são classificados, os espaços são removidos e os cabeçalhos são convertidos em 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.)

Listar todos os meus buckets

Solicitação	StringToSign
<pre>GET / HTTP/1.1 Host: s3.us-west-1.amazonaws.com Date: Wed, 28 Mar 2007 01:29:59 +0000 Authorization: AWS AKIAIOSFODNN7EXAMPLE:qGzdERIC03wnaRNKh6OqZehG9s=</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%a9re HTTP/1.1 Host: s3.us-west-1.amazonaws.com Date: Wed, 28 Mar 2007 01:49:49 +0000 Authorization: AWS AKIAIOSFODNN7EXAMPLE:DNEZGsoieTZ92F3bUfSPQcbGm%2C3%a8re</pre>	<pre>GET\n \n \n Wed, 28 Mar 2007 01:49:49 +0000\n /dictionary/fran%C3%A7ais/pr%c3%a9re</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 tcptrace.

Alternativa de autenticação de solicitação por string de consulta

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.

Para obter mais informações sobre como usar parâmetros de consulta para autenticar solicitações, consulte [Autenticação de solicitações: uso de parâmetros de consulta \(AWS Signature Version 4\)](#) na Referência de APIs do Amazon Simple Storage Service. Para obter exemplos de uso de AWS SDKs para gerar URLs pré-assinados, consulte [Compartilhar objetos usando URLs pré-assinados \(p. 275\)](#).

Criar uma assinatura

Veja a seguir um exemplo de solicitação REST do Amazon S3 autenticada por query string.

```
GET /photos/puppy.jpg
?AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Expires=1141889120&Signature=vjbyPxybdZaNmGa
%2ByT272YEAiv4%3D HTTP/1.1
Host: awsexamplebucket1.s3.us-west-1.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 string de consulta	Valor de exemplo	Descrição
<code>AWSAccessKeyId</code>	<code>AKIAIOSFODNN7EXAMPLE</code>	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 do desenvolvedor que fez a solicitação.
<code>Expires</code>	<code>1141889120</code>	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.

Nome de parâmetro de string de consulta	Valor de exemplo	Descrição
Signature	vjbyPxybdZaNmGa %2ByT272YEAv4%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( YourSecretAccessKey, UTF-8-Encoding-Of( StringToSign ) ) ) );

StringToSign = HTTP-VERB + "\n" +
    Content-MD5 + "\n" +
    Content-Type + "\n" +
    Expires + "\n" +
    CanonicalizedAmzHeaders +
    CanonicalizedResource;
```

YourSecretAccessKey é o ID da chave de acesso secreta da AWS que a Amazon atribui a você ao se cadastrar 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 string de consulta

Solicitação	StringToSign
<pre>GET /photos/puppy.jpg? AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE& Signature=NpgCjnDzrM %2BWpzENXmpNDUsSn8%3D& Expires=1175139620 HTTP/1.1 Host: awsexamplebucket1.s3.us- west-1.amazonaws.com</pre>	<pre>GET\n \n \n 1175139620\n /awsexamplebucket1/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 [Exemplos de autenticação \(p. 1290\)](#) do Amazon S3.

Uploads baseados no navegador usando POST (AWS Signature Version 2)

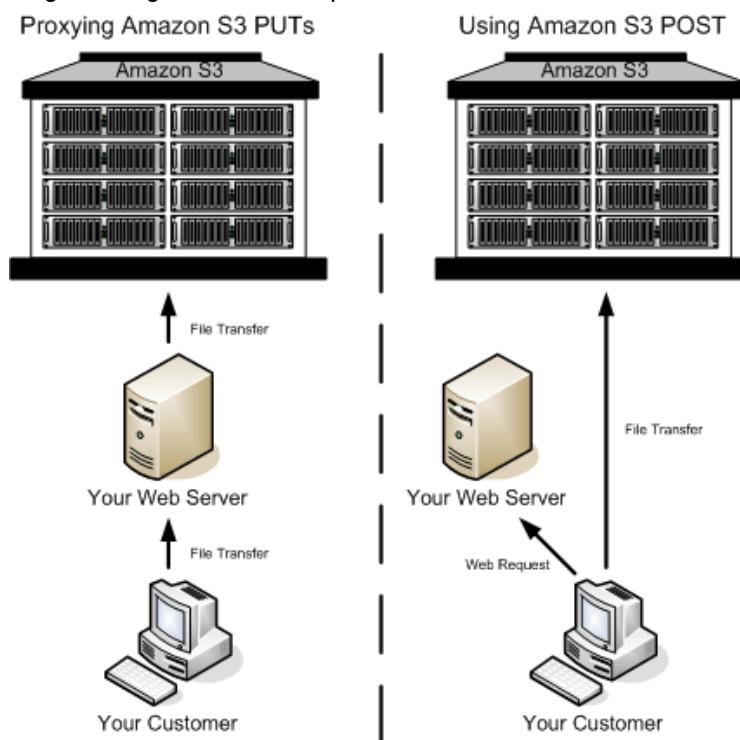
O Amazon S3 é compatível com POST, o que permite aos usuários fazer upload de conteúdo diretamente para o Amazon S3. POST foi projetada para simplificar uploads, reduzir a latência do upload e economizar seu dinheiro nas aplicações em que os usuários fazem upload de dados para armazenamento para o Amazon S3.

Note

A autenticação de solicitação discutida nesta seção é baseada no AWS Signature Version 2, um protocolo para autenticar solicitações de entrada da API para serviços da AWS.

Agora, o Amazon S3 é compatível com o Signature Version 4, um protocolo para autenticação de solicitações de API de entrada para produtos 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 Version 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 [\(Autenticação de solicitações em uploads baseados em navegador usando POST \(AWS Signature Version 4\)\)](#) na Referência de APIs do Amazon Simple Storage Service.

A figura a seguir mostra um upload usando 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 tem um formulário HTTP que contém toda as informações necessárias para que o usuário faça upload de conteúdo no Amazon S3.
3	O usuário faz uploads do conteúdo diretamente para o Amazon S3.

Note

Não há suporte para autenticação por query string para POST.

Formulários HTML (AWS Signature Version 2)

Tópicos

- [Codificação do formulário HTML \(p. 1297\)](#)
- [Declaração do formulário HTML \(p. 1298\)](#)
- [Campos do formulário HTML \(p. 1298\)](#)
- [Criação de política \(p. 1301\)](#)
- [Criar uma assinatura \(p. 1304\)](#)
- [Redirecionamento \(p. 1305\)](#)

Ao se comunicar com o Amazon S3, você normalmente usa as APIs REST ou SOAP para executar as operações “put”, “get”, “delete”, entre outras. Com POST, os usuários fazem upload dos dados diretamente para o Amazon S3 por meios dos navegadores, que não são capazes de processar a API SOAP ou criar uma solicitação PUT de REST.

Note

O suporte a SOAP via HTTP está obsoleto, mas o SOAP continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com SOAP. Em vez de SOAP, recomendamos usar a API REST ou os AWS SDKs.

Para permitir que os usuários façam upload de conteúdo para o 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 do 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 seu bucket for `awsexamplebucket1` e a região for Oeste dos EUA (Norte da Califórnia), o URL será `https://awsexamplebucket1.s3.us-west-1.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 "awsexamplebucket1".

```
<form action="https://awsexamplebucket1.s3.us-west-1.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
<code>AWSAccessKeyId</code>	O ID da 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
<code>acl</code>	Uma lista de controle de acesso (ACL) do Amazon S3. Se uma lista de controle de acesso inválida for especificada, um erro será	Não

Nome do campo	Descrição	Obrigatório
	<p>gerado. Para obter mais informações sobre as ACLs, consulte Listas de controle de acesso (ACLs) (p. 6).</p> <p>Type: string</p> <p>Padrão: privado</p> <p>Valores válidos: private public-read public-read-write aws-exec-read authenticated-read bucket-owner-read bucket-owner-full-control</p>	
<code>Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires</code>	Cabeçalhos específicos para REST. Para obter mais informações, consulte Objeto PUT .	Não
<code>key</code>	<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 Trabalhar com metadados de objeto (p. 166).</p>	Sim
<code>policy</code>	Política de segurança que descreve o que é permitido na solicitação. As solicitações sem uma política de segurança são consideradas anônimas e terão sucesso apenas em buckets com gravação pública.	Não

Nome do campo	Descrição	Obrigatório
<code>success_action_redirect</code> , <code>redirect</code>	<p>O URL para o qual o cliente é redirecionado após um upload bem-sucedido. O Amazon S3 anexa os valores de bucket, chave e etag como parâmetros de query string ao 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 for capaz de interpretar o URL, o campo será ignorado.</p> <p>Se houver falha no upload, o Amazon S3 exibirá um erro e não redirecionará o usuário para um URL.</p> <p>Para obter mais informações, consulte Redirecionamento (p. 1305).</p> <p>Note</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 estiver definido como 200 ou 204, o Amazon S3 retornará um documento vazio com um código de status 200 ou 204.</p> <p>Se o valor estiver definido como 201, o Amazon S3 retornará um documento XML com um código de status 201. Para obter informações sobre o conteúdo do documento XML, consulte Objeto POST.</p> <p>Se o valor não estiver definido ou for um valor inválido, o Amazon S3 retornará um documento vazio com um código de status 204.</p> <p>Note</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
<code>signature</code>	<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 Identity and Access Management no Amazon S3 (p. 419).</p>	Condisional
<code>x-amz-security-token</code>	<p>Um token de segurança usado por credenciais de sessão</p> <p>Se a solicitação estiver usando o Amazon DevPay, serão necessários dois campos do formulário <code>x-amz-security-token</code>: 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 <code>x-amz-security-token</code>. Para obter mais informações, consulte Credenciais de segurança temporárias no Manual do usuário do IAM.</p>	Não
Outros nomes de campos com o prefixo <code>x-amz-meta-</code>	<p>Metadados especificados pelo usuário.</p> <p>O Amazon S3 não valida ou usa esses dados.</p> <p>Para obter mais informações, consulte Objeto PUT.</p>	Não
<code>file</code>	<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 \(p. 1302\)](#)
- [Condições \(p. 1302\)](#)
- [Correspondência de condição \(p. 1303\)](#)
- [Caracteres de escape \(p. 1304\)](#)

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": "awsexamplebucket1"},  
    ["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 \(p. 1303\)](#).

A tabela a seguir descreve as condições do documento de política.

Nome do elemento	Descrição
acl	Especifica as condições que a ACL deve satisfazer. Oferece suporte à correspondência exata e a starts-with.
content-length-range	Especifica os tamanhos mínimo e máximo permitidos para o conteúdo carregado. Oferece suporte à correspondência por intervalo.

Nome do elemento	Descrição
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	Cabeçalhos específicos para REST. Oferece suporte à correspondência exata e a starts-with.
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 do 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>

Condição	Descrição
	["eq", "\$acl", "public-read"]
Inicia com	<p>Se o valor deve iniciar com um certo valor, use starts-with. Este exemplo indica que a chave deve iniciar com user/betty:</p> <pre>["starts-with", "\$key", "user/betty/"]</pre>
Corresponder qualquer conteúdo	<p>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:</p> <pre>["starts-with", "\$success_action_redirect", ""]</pre>
Especificando intervalos	<p>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:</p> <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.

Etapa	Descrição
4	Codifique a assinatura SHA-1 usando Base64.

Para obter informações gerais sobre a autenticação, consulte [Identity and Access Management no Amazon S3 \(p. 419\)](#).

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 o 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 houver falha na solicitação POST, 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 Version 2)

Tópicos

- [Upload de arquivo \(p. 1305\)](#)
- [Upload de área de texto \(p. 1308\)](#)

Note

A autenticação de solicitação discutida nesta seção é baseada no AWS Signature Version 2, um protocolo para autenticar solicitações de entrada da API para serviços da AWS.

Agora, o Amazon S3 é compatível com o Signature Version 4, um protocolo para autenticação de solicitações de API de entrada para produtos 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 Version 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 em navegador usando HTTP POST \(usando o AWS Signature Version 4\)](#) na Referência de APIs do Amazon Simple Storage Service.

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 é compatível com uploads para o bucket awsexamplebucket1 do Amazon S3.

```
{ "expiration": "2007-12-01T12:00:00.000Z", }
```

```
"conditions": [
    {"bucket": "awsexamplebucket1"},
    ["starts-with", "$key", "user/eric/"],
    {"acl": "public-read"},
    {"success_action_redirect": "https://awsexamplebucket1.s3.us-west-1.amazonaws.com/
successful_upload.html"},
    ["starts-with", "$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 upload do conteúdo deve ser feito para o bucket awsexamplebucket1.
- A chave deve começar com "user/eric/".
- A ACL está definida para leitura pública.
- O success_action_redirect está definido como https://awsexamplebucket1.s3.us-
west-1.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.

```
eyAizXhwaXJhdGlvbIi6ICiYMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFOiLAogICJjb25kaXRpb25zIjogWwogICAgeyJidWNrZXQioiA
```

Crie uma assinatura usando suas credenciais. Por exemplo 0RavWzkygo6QX9caELEqKi9kDbU= é a assinatura para o documento de política anterior.

O formulário a seguir é compatível com uma solicitação POST para o bucket DOC-EXAMPLE-BUCKET que usa essa política.

```
<html>
<head>
    ...
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    ...
</head>
<body>
    ...
<form action="https://DOC-EXAMPLE-BUCKET.s3.us-west-1.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="https://
awsexamplebucket1.s3.us-west-1.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>
```

```
</form>
...
</html>
```

Exemplo de solicitação

Essa solicitação pressupõe que a imagem carregada tem 117.108 bytes; os dados da imagem não estão inclusos.

```
POST / HTTP/1.1
Host: awsexamplebucket1.s3.us-west-1.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"

https://awsexamplebucket1.s3.us-west-1.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"

eyAiZXhwXJhdGlvbiI6IClEyMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFoilAogICJjb25kaXRpb25zIjogWwogICAgeyJidWNrZXQiOiA
--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--
```

Exemplo de resposta

```
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: https://awsexamplebucket1.s3.us-west-1.amazonaws.com/
successful_upload.html?bucket=awsexamplebucket1&key=user/eric/
MyPicture.jpg&etag="39d459dfbc0faabb5e179358dfb94c3";
Server: AmazonS3
```

Upload de área de texto

Tópicos

- [Criação de política e formulário \(p. 1308\)](#)
- [Exemplo de solicitação \(p. 1309\)](#)
- [Exemplo de resposta \(p. 1310\)](#)

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 é compatível com uploads de área de texto para o bucket awsexamplebucket1 do Amazon S3.

```
{
  "expiration": "2007-12-01T12:00:00.000Z",
  "conditions": [
    {"bucket": "awsexamplebucket1"},
    ["starts-with", "$key", "user/eric/"],
    {"acl": "public-read"},
    {"success_action_redirect": "https://awsexamplebucket1.s3.us-west-1.amazonaws.com/
new_post.html"},
    ["eq", "$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 upload do conteúdo deve ser feito para o bucket awsexamplebucket1.
- A chave deve começar com "user/eric/".
- A ACL está definida para leitura pública.
- O success_action_redirect está definido como https://awsexamplebucket1.s3.us-west-1.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.

```
eyAizXhwaXJhdGlvbiI6ICiyMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFOiLAogICJjb25kaXRpb25zIjogWwogICAgeyJidWNrZXQiOiAiam9obnNtaXRoIn0sCiAgICBbInN0YXJ0cy13aXRoIiwgIIrrZXkiLCACaidxNlci9lcmljLyLAogICAgeyJhY2wiOiAicHVibGljLXJlYWQifSwKICAgIHsic3VjY2Vzc19hY3Rpb25fcvmVkaXJlY3QjOiaHR0cDovL2pvaG5zbWLc5zM5hbWF6b25hd3MuY29tL25ld19wb3N0Lmh0bWwifSwKICAgIFsizXEiLCACiJENvbnRlbnQtVHlwZSISICJ0ZXh0L2h0bWwiXSwsKCAgIHSieC1hbXotbWV0YS11dwlkIjogIjE0MzY1MTIzNjUxMjc0In0sCiAgICBbInN0YXJ0cy13aXRoIiwgIIr4LWFtei1tzXrhLXRhIsICIixQogIF0KfQo=
```

Crie uma assinatura usando suas credenciais. Por exemplo, qA7FWXKq6VvU68lI9KdveT1cWgF= é a assinatura para o documento de política anterior.

O formulário a seguir é compatível com uma solicitação POST para o bucket DOC-EXAMPLE-BUCKET que usa essa política.

```
<html>
  <head>
    ...
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    ...
  </head>
  <body>
    ...
    <form action="https://DOC-EXAMPLE-BUCKET.s3.us-west-1.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="https://awsexamplebucket1.s3.us-west-1.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>
```

Exemplo de solicitação

Essa solicitação pressupõe que a imagem carregada tem 117.108 bytes; os dados da imagem não estão inclusos.

```
POST / HTTP/1.1
Host: awsexamplebucket1.s3.us-west-1.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"

https://awsexamplebucket1.s3.us-west-1.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"
eyAIZXhwaXJhdGlvbiI6IClEyMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFOiLAogICJjb25kaXRpb25zIjogWwogICAgeyJidWNrZXQioiA

--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--
```

Exemplo de resposta

```
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: https://awsexamplebucket1.s3.us-west-1.amazonaws.com/new_post.html?
bucket=awsexamplebucket1&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 no 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 aplicações do Adobe Flash que fazem upload para o 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 a POST para retornar uma resposta sem corpo vazio, defina `success_action_status` para 201. O Amazon S3 retornará um documento XML com um 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 \(p. 1298\)](#).

Padrões de design de melhores práticas: otimizar a performance do Amazon S3

As aplicações podem executar facilmente milhares de transações por segundo em performance de solicitação ao fazer upload e recuperar armazenamento do Amazon S3. O Amazon S3 escala automaticamente para taxas de solicitações elevadas. Por exemplo, sua aplicação pode atingir pelo menos 3,5 mil solicitações PUT/COPY/POST/DELETE ou 5,5 mil solicitações GET/HEAD por segundo por [prefixo](#) particionado. Não há limite para o número de prefixos em um bucket. Você pode melhorar a performance de leitura ou gravação usando paralelização. Por exemplo, se você criar 10 prefixos em um bucket do Amazon S3 para paralelizar leituras, poderá escalar a performance de leitura para 55.000 solicitações de leitura por segundo. Da mesma forma, você pode dimensionar as operações de gravação gravando em vários prefixos. Para obter mais informações sobre como criar e usar prefixos, consulte [Organizar objetos usando prefixos \(p. 258\)](#).

Por exemplo, algumas aplicações de data lake no Amazon S3 verificam milhões ou bilhões de objetos para consultas que são executadas em petabytes de dados. Essas aplicações de data lake atingem taxas de transferência de instância única que maximizam o uso da interface de rede para a instância do [Amazon EC2](#), que podem ser de até 100 Gb/s em uma única instância. Esses aplicativos então agregam a taxa de transferência em várias instâncias para obter vários terabits por segundo.

Outros aplicativos são sensíveis à latência, como aplicativos de mensagem de mídias sociais. Essas aplicações podem atingir latências consistentes de objetos pequenos (e latências de saída de primeiro byte para objetos maiores) de aproximadamente 100 a 200 milissegundos.

Outros serviços da AWS também podem ajudar a acelerar a performance para diferentes arquiteturas de aplicação. Por exemplo, para obter taxas de transferência mais altas em uma única conexão HTTP ou latências de um dígito de milissegundos, use o [Amazon CloudFront](#) ou o [Amazon ElastiCache](#) para armazenar em cache com o Amazon S3.

Além disso, se você quiser um transporte de dados rápido em longas distâncias entre um cliente e um bucket do S3, use [Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration \(p. 147\)](#). O Transfer Acceleration usa os pontos de presença distribuídos globalmente no CloudFront para acelerar o transporte de dados em distâncias geográficas. Se a workload do Amazon S3 usa criptografia no lado do servidor com o AWS Key Management Service (SSE-KMS), consulte [Limites do AWS KMS](#) no Guia do desenvolvedor do AWS Key Management Service para obter informações sobre as taxas de solicitações compatíveis com seu caso de uso.

Os tópicos a seguir descrevem as diretrizes de melhores práticas e os padrões de design para otimizar a performance de aplicações que usam o Amazon S3. Consulte [Diretrizes de performance do Amazon S3 \(p. 1313\)](#) e [Padrões de design de performance do Amazon S3 \(p. 1315\)](#) para obter as informações mais recentes sobre a otimização de performance do Amazon S3.

Tópicos

- [Diretrizes de performance do Amazon S3 \(p. 1313\)](#)
- [Padrões de design de performance do Amazon S3 \(p. 1315\)](#)

Diretrizes de performance do Amazon S3

Ao criar aplicações que fazem upload e recuperam objetos do Amazon S3, siga as diretrizes de melhores práticas para otimizar a performance. Também oferecemos mais detalhado [Padrões de design de performance \(p. 1315\)](#).

Para obter a melhor performance para a aplicação no Amazon S3, recomendamos as seguintes diretrizes.

Tópicos

- [Avaliar a performance \(p. 1313\)](#)
- [Dimensionar conexões de armazenamento na horizontal \(p. 1313\)](#)
- [Usar consulta na escala de bytes \(p. 1313\)](#)
- [Solicitações de repetição para aplicativos sensíveis à latência \(p. 1314\)](#)
- [Combinar o Amazon S3 \(armazenamento\) e o Amazon EC2 \(computação\) na mesma Região da AWS \(p. 1314\)](#)
- [Usar o Amazon S3 Transfer Acceleration para minimizar a latência causada pela distância \(p. 1314\)](#)
- [Usar a versão mais recente dos AWS SDKs \(p. 1314\)](#)

Avaliar a performance

Ao otimizar a performance, observe os requisitos de throughput de rede, CPU e DRAM. Dependendo da combinação de demandas desses recursos diferentes, convém avaliar os diferentes tipos de instância do [Amazon EC2](#). Para obter mais informações sobre tipos de instância, consulte [Tipos de instância](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Também é útil observar o tempo de pesquisa de DNS, a latência e a velocidade de transferência dos dados usando ferramentas de análise de HTTP ao avaliar a performance.

Dimensionar conexões de armazenamento na horizontal

A distribuição de solicitações em muitas conexões é um padrão de design comum para dimensionamento horizontal da performance. Ao criar aplicações de alta performance, pense no Amazon S3 como um sistema distribuído muito grande, não como um único endpoint de rede de um servidor de armazenamento tradicional. Para atingir a melhor performance, emita várias solicitações simultâneas para o Amazon S3. Espalhe essas solicitações em conexões separadas para maximizar a largura de banda acessível do Amazon S3. O Amazon S3 não tem limites para o número de conexões feitas ao bucket.

Usar consulta na escala de bytes

Usando o cabeçalho HTTP `Range` em uma solicitação de [Objeto GET](#), é possível consultar um objeto em escala de bytes, transferindo somente a parte especificada. É possível usar conexões simultâneas ao Amazon S3 para buscar diferentes escalas de bytes no mesmo objeto. Isso ajuda a atingir um throughput agregado maior em comparação com uma única solicitação de objeto inteiro. A consulta de escalas menores de um objeto grande também permite que o aplicativo melhore os tempos de repetição quando as solicitações são interrompidas. Para obter mais informações, consulte [Fazer download de um objeto \(p. 223\)](#).

Os tamanhos típicos das solicitações de escala de bytes são 8 MB ou 16 MB. Se os objetos usarem a solicitação PUT com um upload de várias partes, é recomendado usar a solicitação GET nos mesmos tamanhos de parte (ou pelo menos de acordo com os limites de parte) para obter a melhor performance. As solicitações GET podem abordar partes individuais diretamente; por exemplo, `GET ?partNumber=N`.

Solicitações de repetição para aplicativos sensíveis à latência

Repetições e tempos limite agressivos ajudam a obter uma latência consistente. Devido à grande escala do Amazon S3, se a primeira solicitação for lenta, uma solicitação repetida provavelmente seguirá um caminho diferente e será bem-sucedida rapidamente. Os AWS SDKs têm valores configuráveis de tempo limite e repetição que podem ser ajustados de acordo com as tolerâncias da aplicação específica.

Combinar o Amazon S3 (armazenamento) e o Amazon EC2 (computação) na mesma Região da AWS

Embora os nomes de buckets do S3 sejam [globalmente exclusivos](#), cada bucket é armazenado em uma região selecionada ao criar o bucket. Para otimizar a performance, recomendamos acessar o bucket nas instâncias do Amazon EC2 na mesma Região da AWS quando possível. Isso ajuda a reduzir os custos de latência de rede e transferência de dados.

Para obter mais informações sobre custos de transferência de dados, consulte [Definição de preço do Amazon S3](#).

Usar o Amazon S3 Transfer Acceleration para minimizar a latência causada pela distância

[Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration \(p. 147\)](#) O gerencia transferências de arquivos rápidas, fáceis e seguras em longas distâncias entre o cliente e um bucket do S3. O Transfer Acceleration tira proveito dos pontos de presença distribuídos globalmente no [Amazon CloudFront](#). Conforme os dados chegam a um ponto de presença, eles são roteados para o Amazon S3 por um caminho de rede otimizado. O Transfer Acceleration é ideal para transferir gigabytes a terabytes de dados regularmente entre os continentes. Ele também é útil para clientes que fazem upload em um bucket centralizado do mundo todo.

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

Usar a versão mais recente dos AWS SDKs

Os AWS SDKs fornecem suporte integrado a muitas das diretrizes recomendadas para otimizar a performance do Amazon S3. Os SDKs fornecem uma API mais simples para aproveitar o Amazon S3 em uma aplicação e são atualizados regularmente para seguir as práticas recomendadas mais recentes. Por exemplo, os SDKs incluem uma lógica para executar automaticamente solicitações de repetição em erros HTTP 503 e estão investindo em código para responder e se adaptar a conexões lentas.

Os SDKs também fornecem o [Gerenciador de transferências](#), que automatiza conexões de dimensionamento horizontal para atingir milhares de solicitações por segundo, usando solicitações na escala de bytes quando apropriado. É importante usar a última versão dos AWS SDKs para obter os recursos mais recentes de otimização de performance.

Também é possível otimizar a performance ao usar solicitações de API REST HTTP. Ao usar a API REST, siga as mesmas práticas recomendadas que fazem parte dos SDKs. Permita tempo limite e repetição em solicitações lentas e que várias conexões consultem dados de objeto em paralelo. Para obter informações sobre como usar a API REST, consulte a [Referência da API do Amazon Simple Storage Service](#).

Padrões de design de performance do Amazon S3

Ao projetar aplicações para fazer upload e recuperar objetos do Amazon S3, use nossas melhores práticas e padrões de design para atingir a melhor performance da aplicação. Também oferecemos [Diretrizes de performance \(p. 1313\)](#) a ser considerado ao planejar a arquitetura da aplicação.

Para otimizar a performance, você pode usar os padrões de design a seguir.

Tópicos

- [Usar o cache para conteúdo acessado com frequência \(p. 1315\)](#)
- [Tempo limite e repetição para aplicativos sensíveis à latência \(p. 1315\)](#)
- [Dimensionamento horizontal e paralelização de solicitações para alto throughput \(p. 1316\)](#)
- [Usar o Amazon S3 Transfer Acceleration para acelerar transferências de dados geograficamente dispersas \(p. 1317\)](#)

Usar o cache para conteúdo acessado com frequência

Muitas aplicações que armazenam dados no Amazon S3 fornecem um “conjunto de trabalhos” de dados que são solicitados várias vezes pelos usuários. Se uma workload estiver enviando solicitações GET repetidas para um conjunto comum de objetos, você poderá usar um cache, como o [Amazon CloudFront](#), o [Amazon ElastiCache](#) ou o [AWS Elemental MediaStore](#) para optimizar a performance. A adoção bem-sucedida do cache pode resultar em baixar latência e altas taxas de transferência de dados. As aplicações que usam o armazenamento em cache também enviam menos solicitações diretas ao Amazon S3, o que também pode ajudar a reduzir os custos de solicitações.

O Amazon CloudFront é uma rede de entrega de conteúdo (CDN) rápida que armazena os dados do Amazon S3 em cache com transparência em um grande conjunto de pontos de presença (PoPs) distribuídos geograficamente. Quando os objetos podem ser acessados em várias regiões ou pela Internet, o CloudFront permite que os dados sejam armazenados em cache perto dos usuários que estão acessando os objetos. Isso pode resultar na entrega de alta performance de conteúdo popular do Amazon S3. Para obter mais informações sobre o CloudFront, consulte o [Guia do desenvolvedor do Amazon CloudFront](#).

O Amazon ElastiCache é um cache de memória gerenciado. Com o ElastiCache, é possível provisionar instâncias do Amazon EC2 que armazenam objetos em cache na memória. Esse armazenamento em cache resulta em pedidos de redução de magnitude da latência de GET e aumentos significativos no throughput de download. Para usar o ElastiCache, modifique a lógica da aplicação para preencher o cache com objetos dinâmicos e verifique se esses objetos estão presentes no cache antes de solicitá-los do Amazon S3. Para obter exemplos de como usar o ElastiCache para melhorar a performance de GET do Amazon S3, consulte a publicação do blog [Turbocharge Amazon S3 with Amazon ElastiCache for Redis](#).

O AWS Elemental MediaStore é um sistema de armazenamento em cache e de distribuição de conteúdo criado especificamente para fluxos de trabalho de vídeo e entrega de mídia do Amazon S3. O MediaStore fornece APIs de armazenamento completas especificamente para vídeo e é recomendado para workloads de vídeo sensíveis à performance. Para obter informações sobre o MediaStore, consulte o [Guia do usuário do AWS Elemental MediaStore](#).

Tempo limite e repetição para aplicativos sensíveis à latência

Há determinadas situações em que uma aplicação recebe uma resposta do Amazon S3 indicando que uma nova tentativa é necessária. O Amazon S3 mapeia nomes de bucket e de objetos para os dados do

objeto associados a eles. Se um aplicativo gerar altas taxas de solicitação (normalmente taxas constantes de mais de 5.000 solicitações por segundo para um pequeno número de objetos), ele poderá receber respostas HTTP 503 de lentidão. Se esses erros ocorrerem, cada AWS SDK implementará uma lógica de repetição automática usando o recuo exponencial. Se você não estiver usando um AWS SDK, implemente a lógica de repetição ao receber o erro HTTP 503. Para obter informações sobre técnicas de recuo, consulte [Novas tentativas após erros e recuo exponencial na AWS](#) na Referência geral da Amazon Web Services.

O Amazon S3 é dimensionado automaticamente em resposta a novas taxas constantes de solicitação, otimizando a performance dinamicamente. Enquanto o Amazon S3 estiver sendo otimizado internamente para uma nova taxa de solicitação, você receberá respostas de solicitação HTTP 503 temporariamente até a otimização terminar. Depois que o Amazon S3 otimiza a performance internamente para a nova taxa de solicitação, todas as solicitações serão executadas de forma geral sem repetições.

Para aplicações sensíveis à latência, o Amazon S3 recomenda rastrear e repetir agressivamente as operações mais lentas. Ao repetir uma solicitação, recomendamos usar uma nova conexão ao Amazon S3 e executar uma nova pesquisa de DNS.

Ao fazer solicitações de tamanhos variavelmente grandes (por exemplo, mais de 128 MB), recomendamos rastrear o throughput atingido e repetir os 5% mais lentos das solicitações. Ao fazer solicitações menores (por exemplo, menos de 512 KB), onde latências medianas geralmente estão na faixa de dezenas de milissegundos, é recomendado repetir uma operação GET ou PUT depois de 2 segundos. Se outras repetições forem necessárias, é recomendado recuar. Por exemplo, recomendamos emitir uma repetição depois de 2 segundos e uma segunda repetição depois de mais 4 segundos.

Se a aplicação fizer solicitações de tamanho fixo para o Amazon S3, espere tempos de resposta mais consistentes para cada uma dessas solicitações. Nesse caso, uma estratégia simples é identificar o 1% mais lento de solicitações e repeti-las. Uma única repetição consegue reduzir a latência.

Se estiver usando o AWS Key Management Service (AWS KMS) para criptografia no lado do servidor, consulte [Limites](#) no Guia do desenvolvedor do AWS Key Management Service para obter informações sobre as taxas de solicitações compatíveis com seu caso de uso.

Dimensionamento horizontal e paralelização de solicitações para alto throughput

O Amazon S3 é um sistema distribuído muito grande. Para ajudar a aproveitar essa escala, recomendamos dimensionar horizontalmente as solicitações paralelas para os endpoints do serviço Amazon S3. Além de distribuir as solicitações no Amazon S3, esse tipo de abordagem de dimensionamento ajuda a distribuir a carga em vários caminhos na rede.

Para obter altas taxas de transferência, o Amazon S3 recomenda usar aplicações com várias conexões para executar solicitações GET ou PUT paralelas de dados. Por exemplo, isso é compatível com o [Amazon S3 Transfer Manager](#) no AWS SDK for Java, e a maioria dos outros AWS SDKs fornece construções semelhantes. Para alguns aplicativos, você pode atingir conexões paralelas iniciando várias solicitações ao mesmo tempo em diferentes threads de aplicativo ou em diferentes instâncias de aplicativo. A melhor abordagem depende do aplicativo e da estrutura dos objetos acessados.

Você pode usar os AWS SDKs para emitir solicitações GET e PUT diretamente em vez de empregar o gerenciamento de transferências no AWS SDK. Essa abordagem permite ajustar a workload mais diretamente, além de ainda aproveitar o suporte do SDK para repetições e o processamento das eventuais respostas HTTP 503. Como regra geral, ao fazer download de grandes objetos em uma região do Amazon S3 para o [Amazon EC2](#), sugerimos fazer solicitações simultâneas em escalas de bytes de um objeto na granularidade de 8 a 16 MB. Faça uma solicitação simultânea para cada 85 a 90 MB/s da taxa de transferência de rede desejada. Para saturar uma placa de interface de rede (NIC) de 10 Gb/s, você pode usar cerca de 15 solicitações simultâneas em conexões separadas. É possível dimensionar as solicitações simultâneas em mais conexões para saturar NICs mais rápidas, como NICs de 25 Gb/s ou 100 Gb/s.

A avaliação da performance é importante ao ajustar o número de solicitações a serem emitidas ao mesmo tempo. Recomendamos começar com uma única solicitação de cada vez. Meça a largura de banda de rede atingida e o uso de outros recursos que o aplicativo usa no processamento dos dados. Você pode identificar o recurso de gargalo (isto é, o recurso com maior utilização) e, assim, o número de solicitações que provavelmente serão úteis. Por exemplo, se processar uma solicitação por vez usa 25% da CPU, é recomendado acomodar até quatro solicitações simultâneas. A medição é essencial e vale a pena confirmar o uso do recurso conforme a taxa de solicitação aumenta.

Se a aplicação emitir solicitações diretamente para o Amazon S3 usando a API REST, recomendamos usar um grupo de conexões HTTP e reutilizar cada conexão para uma série de solicitações. Evitar a configuração de conexão por solicitação elimina a necessidade de realizar handshakes Secure Sockets Layer (SSL) e TCP de inicialização lenta em cada solicitação. Para obter informações sobre como usar a API REST, consulte a [Referência da API do Amazon Simple Storage Service](#).

Finalmente, vale prestar atenção ao DNS e verificar novamente se as solicitações estão sendo distribuídas em um grande grupo de endereços IP do Amazon S3. As consultas de DNS para o Amazon S3 percorrem uma grande lista de endpoints IP. No entanto, o armazenamento em cache de provedores ou código do aplicativo que reutiliza um único endereço IP não aproveita a diversidade de endereços e o balanceamento de carga associados. Ferramentas de utilitário de rede, como a ferramenta de linha de comando `netstat`, podem mostrar os endereços IP usados para comunicação com o Amazon S3, e nós fornecemos diretrizes para as configurações de DNS que devem ser usadas. Para obter mais informações sobre essas diretrizes, consulte [Fazer solicitações \(p. 1211\)](#).

Usar o Amazon S3 Transfer Acceleration para acelerar transferências de dados geograficamente dispersas

[Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration \(p. 147\)](#)

O é útil para minimizar ou eliminar a latência causada pela distância geográfica entre clientes distribuídos globalmente e uma aplicação regional que usa o Amazon S3. O Transfer Acceleration usa os pontos de presença distribuídos globalmente no CloudFront para transporte de dados. A rede de borda da AWS tem pontos de presença em mais de 50 locais. Atualmente, ela é usada para distribuir conteúdo por meio do CloudFront e fornecer respostas rápidas para consultas de DNS feitas para o [Amazon Route 53](#).

A rede de borda também ajuda a acelerar transferências de dados enviadas e recebidas do Amazon S3. Ela é ideal para aplicativos que transferem dados em ou entre continentes, possuem uma rápida conexão com a Internet, usam objetos grandes ou possuem muito conteúdo para upload. Conforme os dados chegam em um ponto de presença, eles são roteados para o Amazon S3 por um caminho de rede otimizado. Em geral, quanto mais distante você está de uma região do Amazon S3, maior a melhoria de velocidade que pode esperar do uso do Transfer Acceleration.

Você pode configurar o Transfer Acceleration em buckets novos ou existentes. Use um endpoint separado do Amazon S3 Transfer Acceleration para usar os locais de borda da AWS. A melhor maneira de testar se o Transfer Acceleration ajuda a performance da solicitação do cliente é usar a [Ferramenta de comparação de velocidade do Amazon S3 Transfer Acceleration](#). As configurações e condições de rede variam periodicamente e de um lugar para outro. Portanto, você só é cobrado para transferências em que o Amazon S3 Transfer Acceleration pode melhorar a performance do upload. Para obter informações sobre o uso do Transfer Acceleration com diferentes AWS SDKs, consulte [Habilitar e usar o S3 Transfer Acceleration \(p. 150\)](#).

O que é o Amazon S3 on Outposts?

O AWS Outposts é um serviço totalmente gerenciado que oferece a mesma infraestrutura da AWS, APIs, ferramentas e serviços da AWS, para praticamente qualquer datacenter, espaço de colocalização ou instalação on-premises para uma experiência híbrida verdadeiramente consistente. O AWS Outposts é ideal para workloads que exigem acesso de baixa latência a sistemas on-premises, processamento de dados local, residência de dados e migração de aplicações com interdependências do sistema local. Para obter mais informações, consulte [O que é o AWS Outposts?](#) no Guia do usuário do AWS Outposts.

Com o Amazon S3 on Outposts, você pode criar buckets do S3 em seu AWS Outposts, além de armazenar e recuperar objetos facilmente on-premises. O S3 on Outposts fornece uma nova classe de armazenamento, OUTPOSTS, que usa as APIs do Amazon S3 e é projetada para armazenar dados de forma duradoura e redundante em vários dispositivos e servidores em seu AWS Outposts. Você se comunica com o bucket do Outposts usando um ponto de acesso e uma conexão de endpoint em uma Virtual Private Cloud (VPC). É possível usar as mesmas APIs e recursos nos buckets do Outposts da mesma maneira que no Amazon S3, como políticas de acesso, criptografia e marcação. Só é possível usar o S3 on Outposts por meio do AWS Management Console, da AWS Command Line Interface (AWS CLI), de AWS SDKs ou da API REST.

- [Como funciona o S3 on Outposts \(p. 1318\)](#)
- [Recursos do S3 on Outposts \(p. 1321\)](#)
- [Serviços relacionados da \(p. 1322\)](#)
- [Acessar o S3 no Outposts \(p. 1322\)](#)
- [Pagar pelo S3 on Outposts \(p. 1323\)](#)
- [Next steps \(Próximas etapas\) \(p. 1323\)](#)

Como funciona o S3 on Outposts

O S3 on Outposts é um serviço de armazenamento de objetos que armazena dados como objetos em buckets em seu Outpost. Objeto é um arquivo de dados e quaisquer metadados que descrevam o arquivo. Um bucket é um contêiner de objetos.

Para armazenar seus dados no S3 on Outposts, primeiro crie um bucket. Ao criar o bucket, especifique um nome para ele e o Outpost que o conterá. Para acessar o bucket do S3 on Outposts e executar operações de objeto, em seguida você vai criar e configurar um ponto de acesso. Você também deve criar um endpoint para encaminhar solicitações para seu ponto de acesso.

Os pontos de acesso simplificam o acesso a dados para qualquer serviço da AWS ou aplicação de clientes que armazene dados no S3. Os pontos de acesso são endpoints de rede nomeados que são anexados a buckets e podem ser usados para executar operações de objeto, como `GetObject` e `PutObject`. Cada ponto de acesso tem permissões e controles de rede distintos.

Você pode criar e gerenciar seus buckets, pontos de acesso e endpoints do S3 on Outposts usando o AWS Management Console, a AWS CLI, AWS SDKs ou a API REST. Para carregar e gerenciar objetos no bucket do S3 on Outposts, é possível usar a AWS CLI, AWS SDKs ou a API REST.

Regiões

Durante o provisionamento AWS Outposts, você ou a AWS cria uma conexão de link de serviço que conecta seu Outpost de volta à Região da AWS escolhida ou à região inicial do Outposts para operações

de bucket e telemetria. Um Outpost depende da conectividade com a Região da AWS principal. O rack Outposts não foi projetado para operações ou ambientes desconectados com pouca ou nenhuma conectividade. Para obter mais informações, consulte [Conectividade do Outpost com as Regiões da AWS](#) no Manual do usuário do AWS Outposts.

Buckets

Bucket é um contêiner para objetos armazenados no S3 on Outposts. Você pode armazenar qualquer número de objetos em um bucket e ter até 100 buckets por conta e Outpost.

Ao criar um bucket, você insere um nome para ele e escolhe o Outpost onde ele residirá. Assim que você cria um bucket, não pode mais alterar o respectivo nome nem o mover para outro Outpost. Os nomes de buckets devem seguir as [regras de nomenclatura de buckets do Amazon S3](#). No S3 on Outposts, os nomes de bucket são exclusivos de um Outpost e de uma Conta da AWS. Os buckets do S3 on Outposts exigem o `outpost-id`, o `account-id` e o nome do bucket para identificá-los. O exemplo a seguir mostra o formato do nome do recurso da Amazon (ARN) para buckets do S3 no Outposts. O ARN é composto da região na qual o Outpost está hospedado, de sua conta do Outpost, do ID do Outpost e do nome do bucket.

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/bucket/bucket-name
```

Cada objeto está contido em um bucket. É necessário usar pontos de acesso para acessar qualquer objeto em um bucket do Outposts. Ao especificar o bucket para operações de objeto, você usa o ARN do ponto de acesso, que inclui o `outpost-id`, o `account-id` e o nome do ponto de acesso. O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 on Outposts em operações de objeto:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obter mais informações sobre buckets, consulte [Trabalhar com buckets do S3 on Outposts \(p. 1339\)](#).

Objetos

Os objetos são as entidades fundamentais armazenadas no S3 on Outposts. Os objetos consistem em metadados e dados de objeto. Os metadados são um conjunto de pares de nome e valor que descrevem o objeto. Esses pares incluem alguns metadados padrão, como a data da última modificação, e metadados HTTP padrão, como o `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 \(ou nome\) \(p. 5\)](#).

Com o Amazon S3 on Outposts, os dados do objeto são sempre armazenados no Outpost. Quando a AWS instala um rack do Outpost, seus dados permanecem no local do Outpost para atender aos requisitos de residência de dados. Seus objetos nunca saem do Outpost e não estão em uma Região da AWS. Como o AWS Management Console está hospedado na região, você não pode usá-lo para fazer upload de objetos no Outpost nem os gerenciar. No entanto, você pode usar a API REST, a AWS Command Line Interface (AWS CLI) e os SDKs para fazer upload de objetos e gerenciá-los por meio de seus pontos de acesso.

Chaves

Uma chave de objeto (ou nome da chave) é um identificador exclusivo de um objeto em um bucket. Cada objeto em um bucket tem exatamente uma chave. A combinação de um bucket e uma chave de objeto identifica exclusivamente cada objeto.

O exemplo a seguir mostra o formato do ARN para objetos do S3 on Outposts, que inclui o código Região da AWS para a região em que o Outpost está hospedado, o ID da Conta da AWS, o ID do Outpost, o nome do bucket e a chave de objeto:

```
arn:aws:s3-outposts:us-west-2:123456789012:outpost/op-01ac5d28a6a232904/bucket/DOC-EXAMPLE-BUCKET1/object/myobject
```

Para obter mais informações sobre chaves de objeto, consulte [Trabalhar com objetos usando o S3 on Outposts \(p. 1366\)](#).

Classe de armazenamento e criptografia

O S3 no Outposts fornece uma nova classe de armazenamento, o S3 Outposts (**OUTPOSTS**). A classe de armazenamento do S3 Outposts só está disponível para objetos armazenados em buckets no AWS Outposts. Se você tentar usar outras classes de armazenamento do S3 com o S3 on Outposts, o S3 on Outposts retornará o erro `InvalidStorageClass`.

Por padrão, os objetos armazenados na classe de armazenamento S3 Outposts (**OUTPOSTS**) são sempre criptografados usando criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 (SSE-S3). Para obter mais informações, consulte [Criptografia de dados no S3 on Outposts \(p. 1391\)](#).

Política de bucket

Uma política de bucket é baseada em recursos do AWS Identity and Access Management (IAM) que você pode usar para conceder permissões de acesso ao bucket e aos objetos nele contidos. Só o proprietário do bucket pode associar uma política a um bucket. As permissões anexadas ao bucket se aplicam a todos os objetos do bucket que pertencem ao proprietário do bucket. As políticas de bucket são limitadas a 20 KB.

As políticas de bucket usam uma linguagem de política do IAM baseadas em JSON que é padrão na AWS. Você pode usar políticas de bucket para adicionar ou negar permissões para os objetos em um bucket. As políticas de bucket permitem ou negam solicitações com base nos elementos da política. Esses elementos podem incluir o solicitante, ações do S3 on Outposts, recursos e aspectos ou condições da solicitação (por exemplo, o endereço IP usado para fazer a solicitação). Por exemplo, você pode criar uma política de bucket que conceda permissões entre contas para carregar objetos em um bucket do S3 on Outposts e garantir que o proprietário do bucket tenha controle total sobre os objetos carregados. Para obter mais informações, consulte [Exemplos de políticas de bucket \(p. 519\)](#).

Na política de bucket, você pode usar caracteres curinga (*) nos ARNs e outros valores para conceder permissões a um subconjunto de objetos. Por exemplo, você pode controlar o acesso a grupos de objetos que começam com um [prefixo](#) ou termine com uma determinada extensão, como .html.

Pontos de acesso do S3 on Outposts

Os pontos de acesso do Amazon S3 são endpoints de rede nomeados com políticas de acesso dedicadas que descrevem como os dados podem ser acessados usando esse endpoint. Os pontos de acesso simplificam o gerenciamento de acesso a dados em escala para conjuntos de dados compartilhados no S3 on Outposts. Os pontos de acesso são anexados a buckets que você pode usar para executar operações de objeto do S3, como `GetObject` e `PutObject`.

Cada ponto de acesso tem permissões distintas e controles de rede que o S3 on Outposts aplica para qualquer solicitação feita por meio desse ponto de acesso. Cada ponto de acesso impõe uma política de ponto de acesso personalizada que funciona em conjunto com a política de bucket anexada ao bucket subjacente.

Para obter mais informações, consulte [Acessar buckets e objetos do S3 on Outposts \(p. 1339\)](#).

Recursos do S3 on Outposts

Gerenciamento de acesso

O S3 on Outposts fornece recursos para auditoria e gerenciamento de acesso a seus buckets e objetos. Por padrão, os buckets do S3 on Outposts e os objetos que eles contêm são privados. Você tem acesso somente aos recursos do S3 on Outposts criados por você.

Para conceder permissões de recursos detalhadas que sejam compatíveis com seu caso de uso específico ou para auditar as permissões de seus recursos do S3 on Outposts, você pode usar os recursos a seguir.

- [Bloqueio de acesso público do S3](#): bloqueie o acesso público a buckets e objetos. Para buckets no Outposts, a opção Block Public Acces (Bloquear acesso público) está sempre habilitada por padrão.
- [AWS Identity and Access Management \(IAM\)](#): crie usuários do IAM para sua Conta da AWS para gerenciar o acesso aos recursos do S3 on Outposts. Por exemplo, é possível usar o IAM com o S3 on Outposts para controlar o tipo de acesso que um usuário ou um grupo de usuários tem a um bucket.
- [Pontos de acesso do S3 on Outposts](#): gerencie o acesso a dados para conjuntos de dados compartilhados no S3 on Outposts. Os pontos de acesso são endpoints de rede nomeados com políticas de acesso dedicadas. Os pontos de acesso são anexados a buckets e podem ser usados para executar operações de objeto, como `GetObject` e `PutObject`.
- [Políticas de buckets](#): use a linguagem de política baseada em IAM para configurar permissões baseadas em recursos para os buckets do S3 e os objetos neles contidos.
- [AWS Resource Access Manager\(AWS RAM\)](#): compartilhe com segurança a capacidade do S3 on Outposts com Contas da AWS, dentro de sua organização ou unidades organizacionais (OUs) no AWS Organizations.

Registro e monitoramento do armazenamento

O S3 on Outposts fornece ferramentas de registro em log e monitoramento que você pode usar para monitorar e controlar como seus recursos do S3 on Outposts estão sendo usados. Para obter mais informações, consulte [Ferramentas de monitoramento](#).

- [Métricas do Amazon CloudWatch para o S3 on Outposts](#): monitore a integridade operacional de seus recursos e conheça sua disponibilidade de capacidade.
- [Eventos do Amazon CloudWatch Events para o S3 on Outposts](#): crie uma regra para qualquer evento da API do S3 on Outposts para receber notificações por meio de todos os destinos compatíveis do CloudWatch Events, incluindo o Amazon Simple Queue Service (Amazon SQS), o Amazon Simple Notification Service (Amazon SNS) e o AWS Lambda.
- [Logs do AWS CloudTrail para o S3 on Outposts](#): registre ações executadas por um usuário, uma função ou um AWS service (Serviço da AWS) no S3 on Outposts. Os logs do CloudTrail fornecem rastreamento detalhado de API para operações no nível de bucket e objeto do S3.

Consistência forte

O S3 on Outposts oferece uma sólida consistência de leitura após gravação para solicitações PUT e DELETE de objetos no bucket do S3 on Outposts em todas as Regiões da AWS. Esse comportamento se aplica a ambas as gravações em novos objetos, bem como a solicitações PUT que substituem objetos existentes e a solicitações DELETE. Além disso, as etiquetas de objeto e os metadados de objeto do S3 on Outposts (por exemplo, objeto HEAD) são fortemente consistentes. Para obter mais informações, consulte [Modelo de consistência de dados do Amazon S3 \(p. 7\)](#).

Serviços relacionados da

Depois de carregar os dados no S3 on Outposts, você poderá usá-los com outros Serviços da AWS. Os serviços a seguir podem ser usados com mais frequência:

- [Amazon Elastic Compute Cloud \(Amazon EC2\)](#): oferece capacidade de computação escalável na Nuvem AWS. O uso do Amazon EC2 reduz a necessidade de investimento antecipado em hardware. Por isso, você pode desenvolver e implantar aplicações com maior rapidez. É possível usar o Amazon EC2 para executar quantos servidores virtuais forem necessários, configurar a segurança e as redes e gerenciar o armazenamento.
- [Amazon Elastic Block Store \(Amazon EBS\) on Outposts](#): use snapshots locais do Amazon EBS no Outposts para armazenar snapshots de volumes localmente em um Outpost no S3 on Outposts.
- [Amazon Relational Database Service \(Amazon RDS\) on Outposts](#): use backups locais do Amazon RDS para armazenar seus backups do Amazon RDS localmente em seu Outpost.
- [AWS DataSync](#): automatize a transferência de dados entre os Outposts e as Regiões da AWS, escolhendo o que transferir, quando transferir e a quantidade de largura de banda de rede a ser usada. O S3 no Outposts é integrado ao AWS DataSync. Para aplicativos locais que exigem processamento local de alto rendimento, o S3 on Outposts fornece armazenamento de objetos no local para minimizar transferências de dados e buffer de variações de rede, ao mesmo tempo que oferece a você a capacidade de transferir dados facilmente entre Outposts e Regiões da AWS.

Acessar o S3 no Outposts

Você pode trabalhar com o S3 on Outposts de uma das seguintes formas:

AWS Management Console

O console é uma interface de usuário baseada na Web para gerenciar o S3 on Outposts e os recursos da AWS. Se você se cadastrou em uma Conta da AWS, pode acessar o S3 on Outposts fazendo login no AWS Management Console e escolhendo S3 na página inicial do AWS Management Console. Depois, escolha Outposts buckets (Buckets do Outposts) no painel de navegação à esquerda.

AWS Command Line Interface

Você pode usar as ferramentas de linha de comando da AWS para emitir comandos ou criar scripts na linha de comando de seu sistema e executar tarefas da AWS (incluindo o S3).

A [AWS Command Line Interface \(AWS CLI\)](#) fornece comandos para um amplo conjunto de Serviços da AWS. A AWS CLI é compatível com Windows, macOS e Linux. Para começar a usar, consulte o [AWS Command Line Interface Guia do usuário](#). Para obter mais informações sobre os comandos que você pode usar com o S3 on Outposts, consulte [s3api](#), [s3control](#) e [s3outposts](#) na Referência de comando da AWS CLI.

AWS SDKs

A AWS fornece SDKs (kits de desenvolvimento de software) que consistem em bibliotecas e códigos de exemplo para várias linguagens de programação e plataformas (Java, Python, Ruby, .NET, iOS, Android etc.). Os AWS SDKs são uma forma conveniente de criar acesso programático para o S3 on Outposts e a AWS. Como o S3 on Outposts usa os mesmos SDKs do Amazon S3, o S3 on Outposts oferece uma experiência consistente usando a mesma automação e as mesmas APIs e ferramentas do S3.

O S3 on Outposts é um serviço REST. Você pode enviar solicitações para o S3 on Outposts usando bibliotecas do AWS SDK que envolvem a API REST subjacente e simplificam as tarefas de programação.

Por exemplo, os SDKs processam tarefas como calcular assinaturas, assinar solicitações de forma criptográfica, gerenciar erros e novas tentativas automáticas de solicitações. Para obter informações sobre os AWS SDKs, inclusive sobre como baixá-los e instalá-los, consulte [Aprenda a criar na AWS](#).

Pagar pelo S3 on Outposts

Você pode comprar uma variedade de configurações de rack do AWS Outposts, que oferecem uma combinação de tipos de instância do Amazon EC2, volumes de unidade de estado sólido (SSD) de uso geral (SSD) do Amazon EBS (gp2) e o S3 on Outposts. O preço inclui entrega, instalação e manutenção do serviço de infraestrutura, bem como patches e atualizações de software.

Para obter mais informações, consulte [Preços de rack AWS Outposts](#).

Next steps (Próximas etapas)

Para obter mais informações sobre como trabalhar com o S3 on Outposts, consulte os seguintes tópicos:

- [Configurar seu Outpost \(p. 1323\)](#)
- [O Amazon S3 on Outposts é diferente do Amazon S3? \(p. 1324\)](#)
- [Conceitos básicos do Amazon S3 on Outposts \(p. 1326\)](#)
- [Redes para S3 on Outposts \(p. 1338\)](#)
- [Trabalhar com buckets do S3 on Outposts \(p. 1339\)](#)
- [Trabalhar com objetos usando o S3 on Outposts \(p. 1366\)](#)
- [Segurança no S3 on Outposts \(p. 1390\)](#)
- [Gerenciar o armazenamento do S3 on Outposts \(p. 1398\)](#)
- [Desenvolver com o Amazon S3 on Outposts \(p. 1404\)](#)

Configurar seu Outpost

Para começar a usar o Amazon S3 on Outposts, você precisa de um Outpost com capacidade do Amazon S3 implantada em suas instalações. Para obter informações sobre opções para solicitar um Outpost e capacidade do S3, consulte [AWS Outposts](#). Para obter especificações e ver como o S3 on Outposts é diferente do Amazon S3, consulte [O Amazon S3 on Outposts é diferente do Amazon S3? \(p. 1324\)](#).

Para obter mais informações, consulte os tópicos a seguir.

Tópicos

- [Pedir um novo Outpost \(p. 1323\)](#)
- [Adicionar armazenamento do Amazon S3 a um Outpost existente \(p. 1324\)](#)

Pedir um novo Outpost

Se precisar solicitar um novo Outpost com capacidade do S3, consulte [Preços de racks do AWS Outposts](#) para entender a opção de capacidade do Amazon Elastic Compute Cloud (Amazon EC2), do Amazon Elastic Block Store (Amazon EBS) e do Amazon S3.

Depois de selecionar a configuração, siga as etapas em [Create an Outpost and order Outpost capacity \(Criar um Outpost e solicitar capacidade do Outpost\)](#) no Guia do usuário do AWS Outposts.

Adicionar armazenamento do Amazon S3 a um Outpost existente

Se o AWS Outposts já estiver implantado no local, dependendo da configuração atual do Outpost e da capacidade de armazenamento, você poderá adicionar o armazenamento do Amazon S3 a um Outpost existente ou talvez seja necessário trabalhar com a equipe da Conta da AWS para incluir hardware adicional a fim de oferecer compatibilidade com o Amazon S3 on Outposts.

Como adicionar capacidade do Amazon S3 a um Outpost existente

1. Na conta de proprietário do AWS Outposts, faça login no AWS Management Console e abra o console do AWS Outposts em <https://console.aws.amazon.com/outposts/>.
2. Escolha o ID do Outpost ao qual você deseja adicionar capacidade.
3. Escolha Actions (Ações) e, em seguida, Increase capacidade (Aumentar a capacidade).
4. Escolha Request increase (Solicitar aumento).

Note

Um representante da AWS entrará em contato com você em até três dias úteis. O aumento de capacidade pode exigir que a AWS instale equipamentos adicionais em seu local.

O Amazon S3 on Outposts é diferente do Amazon S3?

O Amazon S3 on Outposts fornece armazenamento de objetos ao seu ambiente AWS Outposts on-premises e ajuda você a atender a necessidades de processamento local, residência de dados e alto nível de performance, mantendo os dados próximos às aplicações on-premises. Usando APIs e recursos do Amazon S3, o S3 on Outposts facilita o armazenamento, a proteção, a etiquetagem, a geração de relatórios e o controle de acesso aos dados em seus Outposts, bem como a extensão da infraestrutura da AWS para suas instalações on-premises a fim de oferecer uma experiência híbrida consistente.

Para obter mais informações sobre os aspectos que diferenciam o S3 on Outposts, consulte os tópicos a seguir.

Tópicos

- [Especificações do Amazon S3 no Outposts \(p. 1324\)](#)
- [Operações de API compatíveis com o Amazon S3 no Outposts \(p. 1325\)](#)
- [Recursos do Amazon S3 não compatíveis com o S3 no Outposts. \(p. 1325\)](#)
- [Requisitos de rede do S3 on Outposts \(p. 1326\)](#)

Especificações do Amazon S3 no Outposts

- O tamanho máximo do bucket do Outposts é de 50 TB.
- O número máximo de buckets de Outposts é 100 por Conta da AWS.
- Os buckets do Outposts só podem ser acessados usando pontos de acesso e endpoints.
- O número máximo de pontos de acesso por bucket de Outposts é dez.
- As políticas de ponto de acesso estão limitadas a 20 KB.

- O proprietário do Outpost pode gerenciar o acesso dentro de sua organização no AWS Organizations usando o AWS Resource Access Manager. Todas as contas que precisam de acesso ao Outpost devem estar dentro da mesma organização que a conta de proprietário no AWS Organizations.
- A conta de proprietário do bucket do S3 on Outposts é sempre o proprietário de todos os objetos no bucket.
- Somente a conta de proprietário do bucket do S3 no Outposts pode executar operações no bucket.
- As limitações de tamanho do objeto são consistentes com o Amazon S3.
- Todos os objetos armazenados no S3 no Outposts são armazenados na classe de armazenamento do OUTPOSTS.
- Por padrão, todos os objetos armazenados na classe de armazenamento OUTPOSTS são armazenados usando criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 (SSE-S3). Você também pode optar explicitamente por armazenar objetos usando a criptografia no lado do servidor com chaves de criptografia fornecidas pelo cliente (SSE-C).
- Se não houver espaço suficiente para armazenar um objeto em seu Outpost, a API retornará uma exceção de capacidade insuficiente (ICE).

Operações de API compatíveis com o Amazon S3 no Outposts

Para obter uma lista das operações de API compatíveis com o S3 on Outposts, consulte [Operações de API do Amazon S3 on Outposts \(p. 1404\)](#).

Recursos do Amazon S3 não compatíveis com o S3 no Outposts.

Os recursos do Amazon S3 a seguir não são compatíveis com o Amazon S3 on Outposts. Todas as tentativas de usá-los são rejeitadas.

- Listas de controle de acesso (ACLs)
- Nomes de alias de ponto de acesso
- Compartilhamento de recursos de origem cruzada (CORS)
- Operações em lote do S3
- Relatórios de inventário do S3
- Alterar a criptografia de bucket padrão
- Buckets públicos
- Exclusão de autenticação multifator (MFA)
- Transições do ciclo de vida do S3 (além da exclusão de objetos e da interrupção de carregamentos fracionados incompletos)
- Retenção legal do Bloqueio de objetos do S3
- Retenção do Bloqueio de objetos
- Versionamento do S3
- Criptografia no lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS)
- Replicação do S3
- Controle do tempo de replicação do S3 (S3 RTC)
- Métricas de solicitação do Amazon CloudWatch
- Configuração de métricas
- Transfer Acceleration

- Notificações de eventos do S3
- Buckets de pagamento pelo solicitante
- S3 Select
- Eventos do AWS Lambda
- Server access logging (Registro em log de acesso ao servidor)
- Solicitações HTTP POST
- SOAP
- Acesso ao site

Requisitos de rede do S3 on Outposts

- Para rotear solicitações para um ponto de acesso do S3 no Outposts, você deve criar e configurar um endpoint do S3 no Outposts. Os seguintes limites se aplicam aos endpoints do S3 no Outposts:
 - Cada nuvem privada virtual (VPC) em um Outpost pode ter um endpoint associado e é possível ter até 100 endpoints por Outpost.
 - É possível mapear vários pontos de acesso para o mesmo endpoint.
 - Você só pode adicionar endpoints a VPCs com blocos CIDR nos subespaços dos seguintes intervalos CIDR:
 - 10.0.0.0/8
 - 172.16.0.0/12
 - 192.168.0.0/16
- Você só pode criar endpoints para um Outpost de VPCs que tenham blocos CIDR não sobrepostos.
- Só é possível criar um endpoint de sua própria sub-rede de Outposts.
- A sub-rede utilizada para criar um endpoint deve conter quatro endereços IP para uso do S3 on Outposts.
- Se você especificar o grupo de endereços IP de propriedade do cliente (grupo de CoIP), ele deverá conter quatro endereços IP para uso do S3 on Outposts.
- Só é possível criar um endpoint por Outpost por VPC.

Conceitos básicos do Amazon S3 on Outposts

Com o Amazon S3 on Outposts, é possível criar buckets do S3 no AWS Outposts, além de armazenar e recuperar facilmente objetos no local para aplicações que exigem acesso a dados locais, processamento de dados local e residência de dados. O S3 on Outposts fornece uma nova classe de armazenamento, o S3 Outposts (OUTPOSTS), que usa as APIs do Amazon S3 e é projetado para armazenar dados de forma duradoura e redundante em vários dispositivos e servidores em seu AWS Outposts. Você se comunica com o bucket do Outposts usando um ponto de acesso e uma conexão de endpoint em uma nuvem privada virtual (VPC). É possível usar os mesmos recursos e APIs nos buckets do Outposts da mesma maneira que em buckets do Amazon S3, incluindo políticas de acesso, criptografia e marcação. Só é possível usar o S3 on Outposts por meio do AWS Management Console, da AWS Command Line Interface (AWS CLI), de AWS SDKs ou da API REST.

Com o Amazon S3 on Outposts, você pode usar as APIs e recursos do Amazon S3, como armazenamento de objetos, políticas de acesso, criptografia e marcação, no AWS Outposts da mesma forma que no Amazon S3. Para obter informações sobre o S3 on Outposts, consulte [O que é o Amazon S3 on Outposts? \(p. 1318\)](#).

Tópicos

- [Configurar o IAM com o S3 on Outposts \(p. 1327\)](#)

- [Conceitos básicos do uso do AWS Management Console \(p. 1331\)](#)
- [Como começar a usar a AWS CLI e o SDK para Java \(p. 1333\)](#)

Configurar o IAM com o S3 on Outposts

O AWS Identity and Access Management (IAM) é um serviço da AWS service (Serviço da AWS) que ajuda a controlar o acesso aos recursos da AWS de forma segura. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (ter permissões) a usar os recursos da AWS. O IAM permite que você crie usuários e grupos em sua Conta da AWS. Você controla as permissões que os usuários têm para realizar tarefas usando recursos da AWS. Você pode usar o IAM sem custo adicional.

Por padrão, os usuários do IAM não têm permissões para recursos e operações do Amazon S3 on Outposts. Para permitir que os usuários do IAM gerenciem recursos do S3 on Outposts, faça o seguinte:

- Crie uma política do IAM que conceda permissões a usuários ou grupos do IAM de forma explícita.
- Anexe a política aos usuários ou grupos do IAM que precisam dessas permissões.

Além das políticas do IAM, o S3 on Outposts é compatível com políticas de bucket e de ponto de acesso. As políticas de bucket e políticas de ponto de acesso são [políticas baseadas em recursos](#) que são anexadas ao recurso S3 no Outposts.

- Uma política de bucket é anexada ao bucket e permite ou nega solicitações ao bucket e aos objetos dele com base nos elementos da política.
- Por outro lado, uma política de ponto de acesso é anexada ao ponto de acesso e permite ou nega solicitações ao ponto de acesso.

A política de ponto de acesso funciona com a política de bucket anexada ao bucket do S3 no Outposts. Para que uma aplicação ou um usuário acesse objetos em um bucket do S3 on Outposts por meio de um ponto de acesso do S3 on Outposts, tanto a política de ponto de acesso quanto a política de bucket devem permitir a solicitação.

As restrições que você incluir em uma política de ponto de acesso se aplicam somente a solicitações feitas por meio desse ponto de acesso. Por exemplo, se um ponto de acesso estiver anexado a um bucket, você não poderá usar a política de ponto de acesso para permitir ou negar solicitações feitas diretamente ao bucket. No entanto, as restrições que você aplicar a uma política de bucket podem permitir ou negar solicitações feitas diretamente ao bucket ou por meio do ponto de acesso.

Em uma política do IAM ou em uma política baseada em recursos, você define quais ações do S3 on Outposts serão permitidas ou negadas. As ações do S3 no Outposts correspondem a operações de API específicas do S3 no Outposts. As ações do S3 on Outposts usam o prefixo de namespace s3-outposts:. As solicitações feitas à API de controle do S3 on Outposts em uma Região da AWS e as solicitações feitas a endpoints da API de objeto no Outpost são autenticadas usando o IAM e autorizadas de acordo com o prefixo de namespace s3-outposts:. Para trabalhar com o S3 on Outposts, configure os usuários do IAM e autorize-os no namespace s3-outposts: do IAM.

Para obter mais informações, consulte [Ações, recursos e chaves de condição do Amazon S3 no Outposts](#) na Referência de autorização do serviço.

Note

- As listas de controle de acesso (ACLs) não são compatíveis com o S3 on Outposts.
- Para ajudar a garantir que o proprietário de um bucket não possa ser impedido de acessar ou excluir objetos, o S3 no Outposts usa o proprietário do bucket como proprietário do objeto por padrão.
- O S3 em Outposts sempre tem o Bloqueio de acesso público do S3 habilitado para ajudar a garantir que os objetos nunca tenham acesso público.

Para obter mais informações sobre como configurar o IAM para o S3 on Outposts, consulte os tópicos a seguir.

Tópicos

- [Entidades principais para S3 no Outposts \(p. 1328\)](#)
- [ARNs de recurso para S3 no Outposts \(p. 1328\)](#)
- [Exemplo de políticas para S3 no Outposts \(p. 1329\)](#)
- [Permissões para os endpoints do S3 on Outposts \(p. 1330\)](#)

Entidades principais para S3 no Outposts

Ao criar uma política baseada em recurso para conceder acesso ao seu bucket do S3 no Outposts, você deve usar o elemento `Principal` para especificar a pessoa ou o aplicativo que pode fazer uma solicitação de uma ação ou operação no respectivo recurso. Para políticas do S3 no Outposts, você pode usar uma das seguintes entidades principais:

- Uma Conta da AWS
- Um usuário do IAM
- Um perfil do IAM
- Todas as entidades principais, por meio da especificação de um caractere curinga (*) em uma política que use um elemento `Condition` para limitar o acesso a um intervalo específico de IPs

Important

Não é possível gravar uma política para um bucket do S3 on Outposts que use um caractere curinga (*) no elemento `Principal`, a menos que a política também inclua uma `Condition` que limite o acesso a um intervalo específico de endereços IPs. Essa restrição ajuda a garantir que não haja acesso público ao bucket do S3 on Outposts. Para ver um exemplo, consulte [Exemplo de políticas para S3 no Outposts \(p. 1329\)](#).

Para obter mais informações sobre o elemento `Principal`, consulte [Elementos de política JSON da AWS: entidade principal](#) no Guia do usuário do IAM.

ARNs de recurso para S3 no Outposts

Os nomes de recurso da Amazon (ARNs) para o S3 on Outposts contêm o ID do Outpost, bem como a Região da AWS na qual se encontra o Outpost, o ID da Conta da AWS e o nome do recurso. Para acessar e executar ações em buckets e objetos do Outposts, é necessário usar um dos formatos de ARN mostrados na tabela a seguir.

O valor `partition` no ARN refere-se a um grupo de Regiões da AWS. O escopo de cada Conta da AWS é uma partição. Estas são as partições compatíveis:

- `aws` – Regiões da AWS
- `aws-us-gov`: regiões AWS GovCloud (US)

Formatos de ARN do S3 on Outposts

ARN do Amazon S3 no Outposts	Formato ARN	Exemplo
ARN do bucket	<code>arn:<i>partition</i>:s3-outposts:<i>region</i>:</code>	<code>arn:aws:s3-outposts:us-west-2:123456789012:</code>

ARN do Amazon S3 no Outposts	Formato ARN	Exemplo
	<code>account_id: outpost/outpost_id/ bucket/bucket_name</code>	<code>outpost/ op-01ac5d28a6a232904/ bucket/DOC-EXAMPLE- BUCKET1</code>
ARN do ponto de acesso	<code>arn:partition:s3- outposts:region: account_id: outpost/outpost_id/ accesspoint/ accesspoint_name</code>	<code>arn:aws:s3-outposts:us- west-2:123456789012: outpost/ op-01ac5d28a6a232904/ accesspoint/access- point-name</code>
ARN do objeto	<code>arn:partition:s3- outposts:region: account_id: outpost/outpost_id/ bucket/bucket_name/ object/object_key</code>	<code>arn:aws:s3-outposts:us- west-2:123456789012: outpost/ op-01ac5d28a6a232904/ bucket/DOC-EXAMPLE- BUCKET1/object/myobject</code>
ARN do objeto de ponto de acesso do S3 on Outposts (usado nas políticas)	<code>arn:partition:s3- outposts:region: account_id: outpost/outpost_id/ accesspoint/ accesspoint_name/ object/object_key</code>	<code>arn:aws:s3-outposts:us- west-2:123456789012: outpost/ op-01ac5d28a6a232904/ accesspoint/access- point-name/object/ myobject</code>
ARN do S3 no Outposts	<code>arn:partition:s3- outposts:region: account_id: outpost/outpost_id</code>	<code>arn:aws:s3-outposts:us- west-2:123456789012: outpost/ op-01ac5d28a6a232904</code>

Exemplo de políticas para S3 no Outposts

Example : política de bucket do S3 on Outposts com uma entidade principal da Conta da AWS

A política de bucket a seguir usa uma entidade principal da Conta da AWS para conceder acesso a um bucket do S3 no Outposts. Para usar essa política de bucket, substitua os *user input placeholders* por suas próprias informações.

```
{
    "Version": "2012-10-17",
    "Id": "ExampleBucketPolicy1",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "123456789012"
            },
            "Action": "s3-outposts:*",
            "Resource": "arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/  
bucket/example-outpost-bucket"
        }
    ]
}
```

Example : política de bucket do S3 on Outposts com uma entidade principal de caractere curinga (*) e chave de condição para limitar o acesso a um intervalo específico de endereços IPs

A política de bucket a seguir usa uma entidade principal de caractere curinga (*) com a condição aws:SourceIp para limitar o acesso a um intervalo específico de endereços IPs. Para usar essa política de bucket, substitua os *user input placeholders* por suas próprias informações.

```
{
    "Version": "2012-10-17",
    "Id": "ExampleBucketPolicy2",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": { "AWS" : "*" },
            "Action": "s3-outposts:*",
            "Resource": "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket",
            "Condition" : {
                "IpAddress" : {
                    "aws:SourceIp": "192.0.2.0/24"
                },
                "NotIpAddress" : {
                    "aws:SourceIp": "198.51.100.0/24"
                }
            }
        }
    ]
}
```

Permissões para os endpoints do S3 on Outposts

O S3 on Outposts requer suas próprias permissões no IAM para gerenciar ações de endpoint do S3 on Outposts.

Note

- Para endpoints que usam o tipo de acesso de grupo de endereços IP de propriedade do cliente (grupo de CoIP), você também precisa ter permissões para trabalhar com endereços IP de seu grupo de CoIP, conforme descrito na tabela a seguir.
- Os usuários em contas compartilhadas que acessam o S3 on Outposts usando o AWS Resource Access Manager não podem criar seus próprios endpoints em uma sub-rede compartilhada. Se o usuário de uma conta compartilhada quiser gerenciar seus próprios endpoints, a conta compartilhada deverá criar sua própria sub-rede no Outpost. Para obter mais informações, consulte [the section called “Compartilhar o S3 on Outposts” \(p. 1401\)](#).

Permissões do IAM relacionadas a endpoints do S3 on Outposts

Ação	Permissões do IAM
CreateEndpoint	s3-outposts:CreateEndpoint ec2:CreateNetworkInterface ec2:DescribeNetworkInterfaces ec2:DescribeVpcs ec2:DescribeSecurityGroups

Ação	Permissões do IAM
	<p><code>ec2:DescribeSubnets</code></p> <p><code>ec2:CreateTags</code></p> <p>Para endpoints que estão usando o tipo de acesso ao grupo de endereços IP de propriedade do cliente (grupo do CoIP) on-premises, são necessárias as seguintes permissões adicionais:</p> <p><code>s3-outposts>CreateEndpoint</code></p> <p><code>ec2:DescribeCoipPools</code></p> <p><code>ec2:GetCoipPoolUsage</code></p> <p><code>ec2:AllocateAddress</code></p> <p><code>ec2:AssociateAddress</code></p> <p><code>ec2:DescribeAddresses</code></p> <p><code>ec2:DescribeLocalGatewayRouteTableVpcAssociations</code></p>
<code>DeleteEndpoint</code>	<p><code>s3-outposts>DeleteEndpoint</code></p> <p><code>ec2>DeleteNetworkInterface</code></p> <p><code>ec2:DescribeNetworkInterfaces</code></p> <p>Para endpoints que estão usando o tipo de acesso ao grupo de endereços IP de propriedade do cliente (grupo do CoIP) on-premises, são necessárias as seguintes permissões adicionais:</p> <p><code>s3-outposts>DeleteEndpoint</code></p> <p><code>ec2:DisassociateAddress</code></p> <p><code>ec2:DescribeAddresses</code></p> <p><code>ec2:ReleaseAddress</code></p>
<code>ListEndpoints</code>	<code>s3-outposts>ListEndpoints</code>

Note

Você pode usar tags de recurso em uma política do IAM para gerenciar permissões.

Conceitos básicos do uso do AWS Management Console

Com o Amazon S3 on Outposts, é possível criar buckets do S3 no AWS Outposts, além de armazenar e recuperar facilmente objetos no local para aplicações que exigem acesso a dados locais, processamento de dados local e residência de dados. O S3 on Outposts fornece uma nova classe de armazenamento, o S3 Outposts (OUTPOSTS), que usa as APIs do Amazon S3 e é projetado para armazenar dados de forma duradoura e redundante em vários dispositivos e servidores em seu AWS Outposts. Você se comunica

com o bucket do Outposts usando um ponto de acesso e uma conexão de endpoint em uma nuvem privada virtual (VPC). É possível usar os mesmos recursos e APIs nos buckets do Outposts da mesma maneira que em buckets do Amazon S3, incluindo políticas de acesso, criptografia e marcação. Só é possível usar o S3 on Outposts por meio do AWS Management Console, da AWS Command Line Interface (AWS CLI), de AWS SDKs ou da API REST. Para obter mais informações, consulte [O que é o Amazon S3 on Outposts? \(p. 1318\)](#)

Para começar a usar o S3 on Outposts utilizando o console, consulte os tópicos a seguir. Para começar a usar a AWS CLI ou o AWS SDK for Java, consulte.

Tópicos

- [Criar um bucket, um ponto de acesso e um endpoint \(p. 1332\)](#)
- [Next steps \(Próximas etapas\) \(p. 1333\)](#)

Criar um bucket, um ponto de acesso e um endpoint

O procedimento a seguir mostra como criar seu primeiro bucket no S3 on Outposts. Ao criar um bucket pela primeira vez usando o console, você também cria um ponto de acesso e um endpoint associados ao bucket para que possa começar a armazenar objetos no bucket imediatamente.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts buckets (Buckets do Outposts).
3. Escolha Create Outposts bucket (Criar bucket do Outposts).
4. Em Bucket name (Nome do bucket), insira um nome compatível com o Sistema de Nome de Domínio (DNS) para seu bucket.

O nome do bucket deve:

- Ser único dentro da Conta da AWS, do Outpost e da Região da AWS na qual o Outpost está alojado.
- Ter entre 3 e 63 caracteres.
- Não contém caracteres maiúsculos.
- Começar com uma letra minúscula ou um número.

Depois de criado o bucket, você não pode mudar seu nome. Para obter informações sobre nomeação de buckets, consulte [Regras de nomeação de bucket \(p. 127\)](#).

Important

Evite incluir informações confidenciais, como números de conta, no nome do bucket. O nome do bucket é visível nos URLs que apontam para os objetos no bucket.

5. Em Outpost, escolha o Outpost onde você quer que o bucket resida.
6. (Opcional) Adicione as optional tags (etiquetas opcionais) que você gostaria de associar ao bucket do Outposts. Você pode usar etiquetas para monitorar os critérios para projetos específicos ou grupos de projetos ou para rotular seus buckets usando etiquetas de alocação de custo.

Por padrão, todos os objetos armazenados no bucket do Outposts são armazenados usando criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 (SSE-S3). Você também pode optar explicitamente por armazenar objetos usando a criptografia no lado do servidor com chaves de criptografia fornecidas pelo cliente (SSE-C). Para alterar o tipo de criptografia, você deve usar a API REST, a AWS Command Line Interface (AWS CLI) ou os AWS SDKs.

7. Na seção Outposts access point settings (Configurações do ponto de acesso do Outposts), insira o nome do ponto de acesso.

Os pontos de acesso do S3 on Outposts simplificam o gerenciamento de acesso a dados em escala para conjuntos de dados compartilhados no S3 on Outposts. Os pontos de acesso são endpoints de rede nomeados que são anexados a buckets do Outposts que você pode usar para executar operações de objeto S3. Para obter mais informações, consulte [Pontos de acesso \(p. 1340\)](#).

Os nomes de pontos de acesso devem ser exclusivos dentro da conta para esta região e Outpost, além de cumprir com o [Restrições e limitações de pontos de acesso \(p. 337\)](#).

8. Escolha a VPC para este ponto de acesso do Amazon S3 no Outposts.

Se você não tiver uma VPC, escolha Create VPC (Criar VPC). Para obter mais informações, consulte [Criar pontos de acesso restritos a uma nuvem privada virtual \(p. 327\)](#).

A Virtual Private Cloud (VPC) permite iniciar recursos da AWS em uma rede virtual definida por você. Essa rede virtual se assemelha a uma rede tradicional que você operaria no seu próprio datacenter, com os benefícios de usar a infraestrutura escalável da AWS.

9. (Opcional para uma VPC existente) Escolha uma Sub-rede de endpoint para o endpoint.

Uma sub-rede é uma gama de endereços IP na VPC. Se você não tiver a sub-rede desejada, escolha Create subnet (Criar sub-rede). Para obter mais informações, consulte [Redes para S3 on Outposts \(p. 1338\)](#).

10. (Opcional para uma VPC existente) Escolha um Grupo de segurança de endpoint para o endpoint.

Um [grupo de segurança](#) atua como um firewall virtual para controlar o tráfego de entrada e saída.

11. (Opcional para uma VPC existente) Escolha o Tipo de acesso ao endpoint:

- Private (Privado): a ser usado com a VPC.
- Customer owned IP (IP de propriedade do cliente): a ser usado com um grupo de endereços IP (grupo CoIP) de propriedade do cliente em sua rede on-premises.

12. (Opcional) Especifique a política de ponto de acesso do Outpost. O console exibe automaticamente o nome do recurso da Amazon (ARN) do ponto de acesso, que você pode usar na política.

13. Escolha Create Outposts bucket (Criar bucket do Outposts).

Note

Pode levar até cinco minutos para que seu endpoint do Outposts seja criado e seu bucket esteja pronto para uso. Para definir configurações de bucket adicionais, escolha [View details](#) (Visualizar detalhes).

Next steps (Próximas etapas)

Com o Amazon S3 on Outposts, os dados do objeto são sempre armazenados no Outpost. Quando a AWS instala um rack do Outpost, seus dados permanecem no local do Outpost para atender aos requisitos de residência de dados. Seus objetos nunca saem do Outpost e não estão em uma Região da AWS. Como o AWS Management Console está hospedado na região, você não pode usá-lo para fazer upload de objetos no Outpost nem os gerenciar. No entanto, você pode usar a API REST, a AWS Command Line Interface (AWS CLI) e os SDKs para fazer upload de objetos e gerenciá-los por meio de seus pontos de acesso.

Depois de criar um bucket no S3 on Outposts, um ponto de acesso e um endpoint, você pode usar a AWS CLI ou o SDK para Java para carregar um objeto no bucket. Para obter mais informações, consulte [Etapa 4: Fazer upload de um objeto em um bucket do S3 on Outposts \(p. 1336\)](#).

Como começar a usar a AWS CLI e o SDK para Java

Com o Amazon S3 on Outposts, é possível criar buckets do S3 no AWS Outposts, além de armazenar e recuperar facilmente objetos no local para aplicações que exigem acesso a dados locais, processamento

de dados local e residência de dados. O S3 on Outposts fornece uma nova classe de armazenamento, o S3 Outposts (OUTPOSTS), que usa as APIs do Amazon S3 e é projetado para armazenar dados de forma duradoura e redundante em vários dispositivos e servidores em seu AWS Outposts. Você se comunica com o bucket do Outposts usando um ponto de acesso e uma conexão de endpoint em uma nuvem privada virtual (VPC). É possível usar os mesmos recursos e APIs nos buckets do Outposts da mesma maneira que em buckets do Amazon S3, incluindo políticas de acesso, criptografia e marcação. Só é possível usar o S3 on Outposts por meio do AWS Management Console, da AWS Command Line Interface (AWS CLI), de AWS SDKs ou da API REST. Para obter mais informações, consulte [O que é o Amazon S3 on Outposts? \(p. 1318\)](#)

Para começar a usar o S3 on Outposts, você deve criar um bucket, um ponto de acesso e um endpoint. Depois, você pode carregar objetos no bucket. Os exemplos a seguir mostram como começar a usar o S3 on Outposts com a AWS CLI e o SDK para Java. Para começar a usar o console, consulte [Conceitos básicos do uso do AWS Management Console \(p. 1331\)](#).

Tópicos

- [Etapa 1: Criar um bucket \(p. 1334\)](#)
- [Etapa 2: Criar um ponto de acesso \(p. 1335\)](#)
- [Etapa 3: Criar um endpoint \(p. 1335\)](#)
- [Etapa 4: Fazer upload de um objeto em um bucket do S3 on Outposts \(p. 1336\)](#)

Etapa 1: Criar um bucket

Os exemplos de AWS CLI e SDK para Java a seguir mostram como criar um bucket do S3 on Outposts.

AWS CLI

Example

O exemplo a seguir cria um bucket do S3 on Outposts (`s3-outposts:CreateBucket`) usando a AWS CLI. Para executar esse comando, substitua os `user input placeholders` por suas próprias informações.

```
aws s3control create-bucket --bucket example-outpost-bucket --outpost-id op-01ac5d28a6a232904
```

SDK for Java

Example

O exemplo a seguir cria um bucket do S3 on Outposts (`s3-outposts:CreateBucket`) usando o SDK para Java.

```
import com.amazonaws.services.s3control.model.*;

public String createBucket(String bucketName) {

    CreateBucketRequest reqCreateBucket = new CreateBucketRequest()
        .withBucket(bucketName)
        .withOutpostId(OutpostId)
        .withCreateBucketConfiguration(new CreateBucketConfiguration());

    CreateBucketResult respCreateBucket =
        s3ControlClient.createBucket(reqCreateBucket);
    System.out.printf("CreateBucket Response: %s%n", respCreateBucket.toString());

    return respCreateBucket.getBucketArn();
}
```

}

Etapa 2: Criar um ponto de acesso

Para acessar o bucket do Amazon S3 on Outposts, você deve criar e configurar um ponto de acesso. Esses exemplos mostram como criar um ponto de acesso usando a AWS CLI e o SDK para Java.

Os pontos de acesso simplificam o gerenciamento do acesso a dados em escala para conjuntos de dados compartilhados no Amazon S3. Os pontos de acesso são endpoints de rede nomeados e anexados a buckets que você pode usar para realizar operações de objeto do Amazon S3, como `GetObject` e `PutObject`. Com o S3 on Outposts, você deve usar pontos de acesso para acessar qualquer objeto em um bucket do Outposts. Os pontos de acesso são compatíveis apenas com o endereçamento em estilo de host virtual.

AWS CLI

Example

O exemplo da AWS CLI a seguir cria um ponto de acesso para um bucket do Outposts. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control create-access-point --account-id 123456789012
--name example-outposts-access-point --bucket "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-
bucket" --vpc-configuration VpcId=example-vpc-12345
```

SDK for Java

Example

O exemplo do SDK para Java a seguir cria um ponto de acesso para um bucket do Outposts. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
import com.amazonaws.services.s3control.model.*;
public String createAccessPoint(String bucketArn, String accessPointName) {
    CreateAccessPointRequest reqCreateAP = new CreateAccessPointRequest()
        .withAccountId(AccountId)
        .withBucket(bucketArn)
        .withName(accessPointName)
        .withVpcConfiguration(new VpcConfiguration().withVpcId("vpc-12345"));

    CreateAccessPointResult respCreateAP =
        s3ControlClient.createAccessPoint(reqCreateAP);
    System.out.printf("CreateAccessPoint Response: %s%n", respCreateAP.toString());

    return respCreateAP.getAccessPointArn();
}
```

Etapa 3: Criar um endpoint

Para rotear solicitações para um ponto de acesso do Amazon S3 on Outposts, você deve criar e configurar um endpoint do S3 on Outposts. Cada nuvem privada virtual (VPC) em seu Outpost pode ter um endpoint associado. Para obter mais informações sobre cotas de endpoints, consulte [Requisitos de rede do S3 on Outposts \(p. 1326\)](#). Você deve criar esses endpoints para poder acessar seus buckets do Outposts e executar operações de objeto. Para obter mais informações, consulte [Endpoints \(p. 1340\)](#).

Esses exemplos mostram como criar um endpoint usando a AWS CLI e o SDK para Java. Para obter mais informações sobre as permissões necessárias para criar e gerenciar endpoints, consulte [Permissões para os endpoints do S3 on Outposts \(p. 1330\)](#).

AWS CLI

Example

O seguinte exemplo da AWS CLI cria um endpoint para um Outpost usando o tipo de acesso a recursos da VPC. A VPC é derivada da sub-rede. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
subnet-8c7a57c5 --security-group-id sg-ab19e0d1
```

O exemplo a seguir da AWS CLI cria um endpoint para um Outpost usando o tipo de acesso do grupo de endereços IP de propriedade do cliente (grupo do CoIP). Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
subnet-8c7a57c5 --security-group-id sg-ab19e0d1 --access-type CustomerOwnedIp --
customer-owned-ipv4-pool ipv4pool-coip-12345678901234567
```

SDK for Java

Example

O exemplo do SDK para Java a seguir cria um endpoint para um Outpost. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.CreateEndpointRequest;
import com.amazonaws.services.s3outposts.model.CreateEndpointResult;

public void createEndpoint() {
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
        .standard().build();

    CreateEndpointRequest createEndpointRequest = new CreateEndpointRequest()
        .withOutpostId("op-0d79779cef3c30a40")
        .withSubnetId("subnet-8c7a57c5")
        .withSecurityGroupId("sg-ab19e0d1")
        .withAccessType("CustomerOwnedIp")
        .withCustomerOwnedIpv4Pool("ipv4pool-coip-12345678901234567");
    // Use .withAccessType and .withCustomerOwnedIpv4Pool only when the access type is
    // customer-owned IP address pool (CoIP pool)
    CreateEndpointResult createEndpointResult =
    s3OutpostsClient.createEndpoint(createEndpointRequest);
    System.out.println("Endpoint is created and its ARN is " +
    createEndpointResult.getEndpointArn());
}
```

Etapa 4: Fazer upload de um objeto em um bucket do S3 on Outposts

Objetos são as entidades fundamentais armazenadas no Amazon S3 on Outposts. Cada objeto está contido em um bucket. É necessário usar pontos de acesso para acessar qualquer objeto em um bucket do

Outpost. Ao especificar o bucket para operações de objeto, você usa o nome do recurso da Amazon (ARN) do ponto de acesso, que inclui o código Região da AWS da região em que o Outpost está hospedado, o ID da Conta da AWS, o ID do Outpost e o nome do ponto de acesso. O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 on Outposts em operações de objeto:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obter mais informações sobre o S3 on Outposts, consulte [ARNs de recurso para S3 no Outposts \(p. 1328\)](#).

Com o Amazon S3 on Outposts, os dados do objeto são sempre armazenados no Outpost. Quando a AWS instala um rack do Outpost, seus dados permanecem no local do Outpost para atender aos requisitos de residência de dados. Seus objetos nunca saem do Outpost e não estão em uma Região da AWS. Como o AWS Management Console está hospedado na região, você não pode usá-lo para fazer upload de objetos no Outpost nem os gerenciar. No entanto, você pode usar a API REST, a AWS Command Line Interface (AWS CLI) e os SDKs para fazer upload de objetos e gerenciá-los por meio de seus pontos de acesso.

Os exemplos de AWS CLI e AWS SDK for Java a seguir mostram como fazer upload de um objeto em um bucket do S3 on Outposts usando um ponto de acesso.

AWS CLI

Example

O exemplo a seguir coloca um objeto chamado `sample-object.xml` em um bucket do S3 on Outposts (`s3-outposts:PutObject`) usando a AWS CLI. Para usar esse comando, substitua cada `user input placeholder` por suas próprias informações. Para obter mais informações sobre esse comando, consulte [put-object](#) na Referência da AWS CLI.

```
aws s3api put-object --bucket arn:aws:s3-outposts:Region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point --key sample-object.xml --body sample-object.xml
```

SDK for Java

Example

O exemplo a seguir coloca um objeto em um bucket do S3 on Outposts usando o SDK para Java. Para usar esse exemplo, substitua cada `user input placeholder` por suas próprias informações. Para obter mais informações, consulte [Fazer upload de objetos \(p. 171\)](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectMetadata;
import com.amazonaws.services.s3.model.PutObjectRequest;

import java.io.File;

public class PutObject {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String stringObjKeyName = "*** String object key name ***";
        String fileObjKeyName = "*** File object key name ***";
        String fileName = "*** Path to file to upload ***";

        try {
```

```
// This code expects that you have AWS credentials set up per:  
// https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-  
credentials.html  
AmazonS3 s3Client = AmazonS3ClientBuilder.standard()  
.enableUseArnRegion()  
.build();  
  
// Upload a text string as a new object.  
s3Client.putObject(accessPointArn, stringObjKeyName, "Uploaded String  
Object");  
  
// Upload a file as a new object with ContentType and title specified.  
PutObjectRequest request = new PutObjectRequest(accessPointArn,  
fileObjKeyName, new File(fileName));  
ObjectMetadata metadata = new ObjectMetadata();  
metadata.setContentType("plain/text");  
metadata.addUserMetadata("title", "someTitle");  
request.setMetadata(metadata);  
s3Client.putObject(request);  
} catch (AmazonServiceException e) {  
// The call was transmitted successfully, but Amazon S3 couldn't process  
// it, so it returned an error response.  
e.printStackTrace();  
} catch (SdkClientException e) {  
// Amazon S3 couldn't be contacted for a response, or the client  
// couldn't parse the response from Amazon S3.  
e.printStackTrace();  
}  
}  
}  
}
```

Redes para S3 on Outposts

Você pode usar o Amazon S3 on Outposts para armazenar e recuperar objetos on-premises para aplicações que exijam acesso a dados locais, processamento de dados e residência de dados. Esta seção descreve os requisitos de rede para acessar o S3 on Outposts.

Tópicos

- [Escolher seu tipo de acesso às redes \(p. 1338\)](#)
- [Acessar buckets e objetos do S3 on Outposts \(p. 1339\)](#)
- [Interfaces de rede elástica entre contas \(p. 1339\)](#)

Escolher seu tipo de acesso às redes

É possível acessar o S3 on Outposts de dentro de uma VPC ou de sua rede on-premises. Você se comunica com o bucket do Outposts usando um ponto de acesso e uma conexão de endpoint. Isso mantém o tráfego entre a VPC e os buckets do S3 on Outposts na rede da AWS. Ao criar um endpoint, é necessário especificar o tipo de acesso ele como `Private` (para roteamento de VPC) ou `CustomerOwnedIp` [para um grupo de endereços IP do cliente (CoIP)].

- `Private` (para roteamento de VPC): se você não especificar o tipo de acesso, o S3 on Outposts usará `Private` por padrão. Com o tipo de acesso `Private`, as instâncias em sua VPC não exigem que endereços IP públicos se comuniquem com recursos no Outpost. É possível trabalhar com o S3 on Outposts de dentro de uma VPC. Esse tipo de endpoint não é acessível de sua rede on-premises.
- `CustomerOwnedIp` (para grupo de CoIP): se você não usar o padrão para o tipo de acesso `Private` e escolher `CustomerOwnedIp`, especifique um intervalo de endereços IP. Você pode usar esse tipo de

acesso para trabalhar com o S3 on Outposts de sua rede on-premises e em uma VPC. Ao acessar o S3 on Outposts em uma VPC, seu tráfego é limitado à largura de banda do gateway local.

Acessar buckets e objetos do S3 on Outposts

Para acessar buckets e objetos do S3 on Outposts, é preciso ter o seguinte:

- Um ponto de acesso para a VPC.
- Um endpoint para a mesma VPC.
- Uma conexão ativa entre seu Outpost e sua Região da AWS. Para obter mais informações sobre como conectar seu Outpost a uma região, consulte [Conectividade do Outpost para regiões da AWS](#) no Guia do usuário do AWS Outposts.

Para obter mais informações sobre como acessar buckets e objetos no S3 on Outposts, consulte [Trabalhar com buckets do S3 on Outposts \(p. 1339\)](#) e [Trabalhar com objetos usando o S3 on Outposts \(p. 1366\)](#).

Interfaces de rede elástica entre contas

Os endpoints do S3 on Outposts são recursos que recebem nomes de recurso da Amazon (ARNs). Quando esses endpoints são criados, o AWS Outposts configura múltiplas interfaces de rede elástica entre contas. As interfaces de rede elástica entre contas do S3 on Outposts são semelhantes a outras interfaces de rede, com uma exceção: o S3 on Outposts associa as interfaces de rede elástica entre contas a instâncias do Amazon EC2.

O Sistema de Nomes de Domínio (DNS) do S3 on Outposts balanceia a carga de suas solicitações na interface de rede elástica entre contas. O S3 on Outposts cria a interface de rede elástica entre contas em sua conta da AWS que é visível no painel Network interfaces (Interfaces de rede) do console do Amazon EC2.

Para endpoints que usam o tipo de acesso ao grupo de ColIP, o S3 on Outposts aloca e associa endereços IP à interface de rede elástica entre contas do grupo de ColIP configurado.

Trabalhar com buckets do S3 on Outposts

Com o Amazon S3 on Outposts, é possível criar buckets do S3 no AWS Outposts, além de armazenar e recuperar facilmente objetos on-premises para aplicações que exigem acesso a dados locais, processamento de dados local e residência de dados. O S3 on Outposts fornece uma nova classe de armazenamento, o S3 Outposts (OUTPOSTS), que usa as APIs do Amazon S3 e é projetado para armazenar dados de forma duradoura e redundante em vários dispositivos e servidores em seu AWS Outposts. É possível usar os mesmos recursos e APIs nos buckets do Outposts da mesma maneira que no Amazon S3, incluindo políticas de acesso, criptografia e marcação. Para obter mais informações, consulte [O que é o Amazon S3 on Outposts? \(p. 1318\)](#)

Você se comunica com os buckets do Outpost usando um ponto de acesso e uma conexão de endpoint em uma nuvem privada virtual (VPC). Para acessar buckets e objetos do S3 on Outposts, é preciso ter um ponto de acesso para a VPC e um endpoint para a mesma VPC. Para obter mais informações, consulte [Redes para S3 on Outposts \(p. 1338\)](#).

Buckets

No S3 on Outposts, os nomes de bucket são exclusivos de um Outpost e exigem o código Região da AWS para a região em que o Outpost está hospedado, o ID da Conta da AWS, o ID do Outpost e o nome do bucket para identificá-los.

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/bucket/bucket-name
```

Para obter mais informações, consulte [ARNs de recurso para S3 no Outposts \(p. 1328\)](#).

Pontos de acesso

O Amazon S3 on Outposts oferece suporte a pontos de acesso somente de nuvem privada virtual (VPC) como o único meio de acessar os buckets do Outposts.

Os pontos de acesso simplificam o gerenciamento do acesso a dados em escala para conjuntos de dados compartilhados no Amazon S3. Os pontos de acesso são endpoints de rede nomeados e anexados a buckets que você pode usar para realizar operações de objeto do Amazon S3, como `GetObject` e `PutObject`. Com o S3 on Outposts, você deve usar pontos de acesso para acessar qualquer objeto em um bucket do Outposts. Os pontos de acesso são compatíveis apenas com o endereçamento em estilo de host virtual.

O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 on Outposts. O ARN do ponto de acesso inclui o código Região da AWS para a região em que o Outpost está hospedado, o ID da Conta da AWS, o ID do Outpost e o nome do ponto de acesso.

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Endpoints

Para rotear solicitações para um ponto de acesso do S3 no Outposts, você deve criar e configurar um endpoint do S3 no Outposts. Com endpoints do S3 on Outposts, você pode conectar sua VPC de forma privada ao bucket do Outposts. Os endpoints do S3 on Outposts são identificadores de recursos uniformes (URIs) virtuais do ponto de entrada para o bucket do S3 on Outposts. Eles são componentes de VPC escalados horizontalmente, redundantes e altamente disponíveis.

Cada nuvem privada virtual (VPC) em seu Outpost pode ter um endpoint associado, e você pode ter até 100 endpoints por Outpost. Você deve criar esses endpoints para poder acessar seus buckets do Outposts e executar operações de objeto. Criar esses endpoints também possibilita que o modelo e os comportamentos da API sejam os mesmos ao permitir que as mesmas operações funcionem no S3 e no S3 on Outposts.

Operações de API no S3 on Outposts

Para gerenciar operações de API no bucket do Outposts, o S3 on Outposts hospeda um endpoint separado que é distinto do endpoint do Amazon S3. Este endpoint é `s3-outposts.region.amazonaws.com`.

Para usar as operações de API do Amazon S3, é necessário assinar o bucket e os objetos usando o formato de ARN correto. Você deve passar ARNs para operações de API para que o Amazon S3 possa determinar se a solicitação é para o Amazon S3 (`s3-control.region.amazonaws.com`) ou para o S3 on Outposts (`s3-outposts.region.amazonaws.com`). Com base no formato ARN, o S3 pode então assinar e rotear a solicitação adequadamente.

Sempre que uma solicitação é enviada para o plano de controle do Amazon S3, o SDK extrai os componentes do ARN e inclui o cabeçalho adicional `x-amz-outpost-id` com o valor `outpost-id` extraído do ARN. O nome do serviço do ARN é usado para assinar a solicitação antes de ser roteada para o endpoint do S3 on Outposts. Esse comportamento se aplica a todas as operações de API manipuladas pelo cliente `s3control`.

A tabela a seguir lista as operações de API estendidas para o Amazon S3 on Outposts e suas alterações em relação ao Amazon S3.

API	Valor do parâmetro do S3 no Outposts
CreateBucket	Nome do bucket como ARN, ID do Outpost
ListRegionalBuckets	ID do Outpost
DeleteBucket	Nome do bucket como ARN
DeleteBucketLifecycleConfiguration	Nome do bucket como ARN
GetBucketLifecycleConfiguration	Nome do bucket como ARN
PutBucketLifecycleConfiguration	Nome do bucket como ARN
GetBucketPolicy	Nome do bucket como ARN
PutBucketPolicy	Nome do bucket como ARN
DeleteBucketPolicy	Nome do bucket como ARN
GetBucketTagging	Nome do bucket como ARN
PutBucketTagging	Nome do bucket como ARN
DeleteBucketTagging	Nome do bucket como ARN
CreateAccessPoint	Nome do ponto de acesso como ARN
DeleteAccessPoint	Nome do ponto de acesso como ARN
GetAccessPoint	Nome do ponto de acesso como ARN
GetAccessPoint	Nome do ponto de acesso como ARN
ListAccessPoints	Nome do ponto de acesso como ARN
PutAccessPointPolicy	Nome do ponto de acesso como ARN
GetAccessPointPolicy	Nome do ponto de acesso como ARN
DeleteAccessPointPolicy	Nome do ponto de acesso como ARN

Criar e gerenciar o bucket do S3 no Outposts

Para obter mais informações sobre como criar e gerenciar buckets do S3 on Outposts, consulte os tópicos a seguir.

Tópicos

- [Criar um bucket do S3 on Outposts \(p. 1342\)](#)
- [Adicionar etiquetas aos buckets do S3 on Outposts \(p. 1344\)](#)
- [Criar e gerenciar uma configuração de ciclo de vida para um bucket do Amazon S3 on Outposts \(p. 1345\)](#)
- [Gerenciar o acesso a um bucket do Amazon S3 on Outposts usando uma política de bucket \(p. 1350\)](#)
- [Listar buckets do Amazon S3 on Outposts \(p. 1353\)](#)

- [Obter um bucket do S3 on Outposts usando a AWS CLI e o SDK para Java \(p. 1354\)](#)
- [Excluir seu bucket do S3 on Outposts \(p. 1355\)](#)
- [Trabalhar com pontos de acesso do Amazon S3 on Outposts \(p. 1356\)](#)
- [Trabalhar com endpoints do Amazon S3 on Outposts \(p. 1362\)](#)

Criar um bucket do S3 on Outposts

Com o Amazon S3 on Outposts, é possível criar buckets do S3 no AWS Outposts, além de armazenar e recuperar facilmente objetos no local para aplicações que exigem acesso a dados locais, processamento de dados local e residência de dados. O S3 on Outposts fornece uma nova classe de armazenamento, o S3 Outposts (OUTPOSTS), que usa as APIs do Amazon S3 e é projetado para armazenar dados de forma duradoura e redundante em vários dispositivos e servidores em seu AWS Outposts. Você se comunica com o bucket do Outposts usando um ponto de acesso e uma conexão de endpoint em uma nuvem privada virtual (VPC). É possível usar os mesmos recursos e APIs nos buckets do Outposts da mesma maneira que em buckets do Amazon S3, incluindo políticas de acesso, criptografia e marcação. Só é possível usar o S3 on Outposts por meio do AWS Management Console, da AWS Command Line Interface (AWS CLI), de AWS SDKs ou da API REST. Para obter mais informações, consulte [O que é o Amazon S3 on Outposts? \(p. 1318\)](#)

Note

A Conta da AWS que cria o bucket é proprietária dele e é a única que pode confirmar suas ações. Os buckets têm propriedades de configuração como Outpost, etiquetas, criptografia padrão e configurações de ponto de acesso. As configurações de ponto de acesso incluem a nuvem privada virtual (VPC), a política do ponto de acesso para acessar os objetos no bucket e outros metadados. Para obter mais informações, consulte [Especificações do Amazon S3 no Outposts \(p. 1324\)](#).

Os exemplos a seguir mostram como criar um bucket do S3 on Outposts usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) e o AWS SDK for Java.

Uso do console do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts buckets (Buckets do Outposts).
3. Escolha Create Outposts bucket (Criar bucket do Outposts).
4. Em Bucket name (Nome do bucket), insira um nome compatível com o Sistema de Nome de Domínio (DNS) para seu bucket.

O nome do bucket deve:

- Ser único dentro da Conta da AWS, do Outpost e da Região da AWS na qual o Outpost está alojado.
- Ter entre 3 e 63 caracteres.
- Não contém caracteres maiúsculos.
- Começar com uma letra minúscula ou um número.

Depois de criado o bucket, você não pode mudar seu nome. Para obter informações sobre nomeação de buckets, consulte [Regras de nomeação de bucket \(p. 127\)](#).

Important

Evite incluir informações confidenciais, como números de conta, no nome do bucket. O nome do bucket é visível nos URLs que apontam para os objetos no bucket.

5. Em Outpost, escolha o Outpost onde você quer que o bucket resida.
6. (Opcional) Adicione as optional tags (etiquetas opcionais) que você gostaria de associar ao bucket do Outposts. Você pode usar etiquetas para monitorar os critérios para projetos específicos ou grupos de projetos ou para rotular seus buckets usando etiquetas de alocação de custo.

Por padrão, todos os objetos armazenados no bucket do Outposts são armazenados usando criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 (SSE-S3). Você também pode optar explicitamente por armazenar objetos usando a criptografia no lado do servidor com chaves de criptografia fornecidas pelo cliente (SSE-C). Para alterar o tipo de criptografia, você deve usar a API REST, a AWS Command Line Interface (AWS CLI) ou os AWS SDKs.

7. Na seção Outposts access point settings (Configurações do ponto de acesso do Outposts), insira o nome do ponto de acesso.

Os pontos de acesso do S3 on Outposts simplificam o gerenciamento de acesso a dados em escala para conjuntos de dados compartilhados no S3 on Outposts. Os pontos de acesso são endpoints de rede nomeados que são anexados a buckets do Outposts que você pode usar para executar operações de objeto S3. Para obter mais informações, consulte [Pontos de acesso \(p. 1340\)](#).

Os nomes de pontos de acesso devem ser exclusivos dentro da conta para esta região e Outpost, além de cumprir com o [Restrições e limitações de pontos de acesso \(p. 337\)](#).

8. Escolha a VPC para este ponto de acesso do Amazon S3 no Outposts.

Se você não tiver uma VPC, escolha Create VPC (Criar VPC). Para obter mais informações, consulte [Criar pontos de acesso restritos a uma nuvem privada virtual \(p. 327\)](#).

A Virtual Private Cloud (VPC) permite iniciar recursos da AWS em uma rede virtual definida por você. Essa rede virtual se assemelha a uma rede tradicional que você operaria no seu próprio datacenter, com os benefícios de usar a infraestrutura escalável da AWS.

9. (Opcional para uma VPC existente) Escolha uma Sub-rede de endpoint para o endpoint.

Uma sub-rede é uma gama de endereços IP na VPC. Se você não tiver a sub-rede desejada, escolha Create subnet (Criar sub-rede). Para obter mais informações, consulte [Redes para S3 on Outposts \(p. 1338\)](#).

10. (Opcional para uma VPC existente) Escolha um Grupo de segurança de endpoint para o endpoint.

Um [grupo de segurança](#) atua como um firewall virtual para controlar o tráfego de entrada e saída.

11. (Opcional para uma VPC existente) Escolha o Tipo de acesso ao endpoint:

- Private (Privado): a ser usado com a VPC.
- Customer owned IP (IP de propriedade do cliente): a ser usado com um grupo de endereços IP (grupo CoIP) de propriedade do cliente em sua rede on-premises.

12. (Opcional) Especifique a política de ponto de acesso do Outpost. O console exibe automaticamente o nome do recurso da Amazon (ARN) do ponto de acesso, que você pode usar na política.

13. Escolha Create Outposts bucket (Criar bucket do Outposts).

Note

Pode levar até cinco minutos para que seu endpoint do Outposts seja criado e seu bucket esteja pronto para uso. Para definir configurações de bucket adicionais, escolha [View details \(Visualizar detalhes\)](#).

Uso do AWS CLI

Example

O exemplo a seguir cria um bucket do S3 on Outposts (`s3-outposts:CreateBucket`) usando a AWS CLI. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control create-bucket --bucket example-outpost-bucket --outpost-id op-01ac5d28a6a232904
```

Usar o AWS SDK for Java

Example

O exemplo a seguir cria um bucket do S3 on Outposts (`s3-outposts:CreateBucket`) usando o SDK para Java.

```
import com.amazonaws.services.s3control.model.*;  
  
public String createBucket(String bucketName) {  
  
    CreateBucketRequest reqCreateBucket = new CreateBucketRequest()  
        .withBucket(bucketName)  
        .withOutpostId(OutpostId)  
        .withCreateBucketConfiguration(new CreateBucketConfiguration());  
  
    CreateBucketResult respCreateBucket = s3ControlClient.createBucket(reqCreateBucket);  
    System.out.printf("CreateBucket Response: %s%n", respCreateBucket.toString());  
  
    return respCreateBucket.getBucketArn();  
}
```

Adicionar etiquetas aos buckets do S3 on Outposts

Você pode adicionar etiquetas para seus buckets do Amazon S3 on Outpost para monitorar o custo de armazenamento e outros critérios para projetos individuais ou grupos de projetos.

Note

A Conta da AWS que cria o bucket é proprietária dele e é a única que pode alterar suas tags.

Uso do console do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts buckets (Buckets do Outposts).
3. Escolha o bucket do Outposts com as etiquetas que você deseja editar.
4. Escolha a guia Properties (Propriedades).
5. Em Tags, escolha Edit (Editar).
6. Escolha Add new tag (Adicionar nova etiqueta) e insira a Key (Chave) e um Value (Valor) opcional.

Adicione todas as etiquetas que você gostaria de associar ao bucket do Outposts para monitorar outros critérios referentes a projetos individuais ou grupos de projetos.

7. Selecione Save changes.

Uso do AWS CLI

O exemplo de AWS CLI a seguir aplica uma configuração de marcação a um bucket do S3 on Outposts usando um documento JSON na pasta atual que especifica as etiquetas ([tagging.json](#)). Para usar esse exemplo, substitua cada `user input placeholder` por suas próprias informações.

```
aws s3control put-bucket-tagging --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket --tagging file://tagging.json

tagging.json

{
  "TagSet": [
    {
      "Key": "organization",
      "Value": "marketing"
    }
  ]
}
```

O exemplo de AWS CLI a seguir aplica uma configuração de marcação a um bucket do S3 on Outposts diretamente da linha de comando.

```
aws s3control put-bucket-tagging --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket --tagging 'TagSet=[{Key=organization,Value=marketing}]'
```

Para obter mais informações sobre esse comando, consulte [put-bucket-tagging](#) na Referência da AWS CLI.

Criar e gerenciar uma configuração de ciclo de vida para um bucket do Amazon S3 on Outposts

As regras de ciclo de vida dos buckets do Amazon S3 on Outposts são limitadas à exclusão de objetos. Você pode usar regras de ciclo de vida para definir quando iniciar a exclusão de objetos com base na idade ou na data. Você pode criar, habilitar, desabilitar e excluir uma regra de ciclo de vida.

Note

A Conta da AWS que cria o bucket tem propriedade sobre ele e é a única que pode criar, habilitar, desabilitar e excluir uma regra de ciclo de vida.

Para criar e gerenciar a configuração do ciclo de vida do bucket do S3 on Outposts, consulte os tópicos a seguir.

Tópicos

- [Criar e gerenciar uma regra de ciclo de vida usando o AWS Management Console \(p. 1345\)](#)
- [Criar e gerenciar uma configuração de ciclo de vida usando a AWS CLI e o SDK para Java \(p. 1347\)](#)

Criar e gerenciar uma regra de ciclo de vida usando o AWS Management Console

As regras de ciclo de vida dos buckets do Amazon S3 on Outposts são limitadas à exclusão de objetos. Você pode usar regras de ciclo de vida para definir quando iniciar a exclusão de objetos com base na idade ou na data. Você pode criar, habilitar, desabilitar e excluir uma regra de ciclo de vida.

Note

A Conta da AWS que cria o bucket tem propriedade sobre ele e é a única que pode criar, habilitar, desabilitar e excluir uma regra de ciclo de vida.

Para criar e gerenciar uma regra de ciclo de vida para um bucket do S3 on Outposts usando o AWS Management Console, consulte os tópicos a seguir.

Tópicos

- [Criar uma regra de ciclo de vida \(p. 1346\)](#)
- [Habilitar uma regra de ciclo de vida \(p. 1346\)](#)
- [Editar uma regra de ciclo de vida \(p. 1346\)](#)
- [Excluir uma regra de ciclo de vida \(p. 1347\)](#)

Criar uma regra de ciclo de vida

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts buckets (Buckets do Outposts).
3. Escolha o bucket do Outposts para o qual você deseja criar uma regra de ciclo de vida.
4. Escolha a guia Management (Gerenciamento) e escolha Create Lifecycle rule (Criar regra de ciclo de vida).
5. Na seção Lifecycle rule configuration (Configuração da regra de ciclo de vida):
 - a. Insira o Lifecycle rule name (Nome da regra de ciclo de vida).
 - b. Escolha Rule scope (Escopo da regra).

Important

Se você quiser que a regra se aplique a objetos específicos, você deve usar um filtro para identificar esses objetos. Escolha Limit the scope to specific objects or tags (Limitar o escopo a etiquetas ou objetos específicos). Para limitar o escopo, faça o seguinte:

- Adicione um filtro de prefixo para limitar o escopo dessa regra a um único prefixo.
 - Adicione etiquetas para limitar o escopo dessa regra aos pares de chave-valor adicionados.
6. Na seção Lifecycle rule trigger (Acionador de regra de ciclo de vida), escolha o rule trigger (acionador de regra) com base em uma data específica ou na idade do objeto.

Habilitar uma regra de ciclo de vida

Para habilitar e desabilitar uma regra de ciclo de vida de bucket

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts buckets (Buckets do Outposts).
3. Escolha o bucket do Outposts para o qual deseja habilitar e desabilitar uma regra de ciclo de vida.
4. Escolha a guia Management (Gerenciamento) e escolha a Lifecycle rule (Regra de ciclo de vida) que você deseja habilitar ou desabilitar.
5. Em Action (Ação), escolha Enable or disable rule (Habilitar e desabilitar regra).

Editar uma regra de ciclo de vida

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.

2. No painel de navegação à esquerda, escolha Outposts buckets (Buckets do Outposts).
3. Escolha o bucket do Outposts para o qual você deseja editar uma regra de ciclo de vida.
4. Escolha a guia Management (Gerenciamento) e escolha a Lifecycle rule (Regra de ciclo de vida) que você deseja editar.
5. Na seção de Lifecycle rule configuration (Configuração da regra do ciclo de vida), faça o seguinte:
 - a. Atualize o Lifecycle rule name (Nome da regra do ciclo de vida).
 - b. Atualize o Rule scope (Escopo da regra).

Important

Se você quiser que a regra se aplique a objetos específicos, você deve usar um filtro para identificar esses objetos. Escolha Limit the scope to specific objects or tags (Limitar o escopo a etiquetas ou objetos específicos). Para limitar o escopo, faça o seguinte:

- Adicione um filtro de prefixo para limitar o escopo dessa regra a um único prefixo.
 - Adicione etiquetas para limitar o escopo dessa regra aos pares de chave-valor adicionados.
6. Na seção Lifecycle rule trigger (Acionador de regra de ciclo de vida), atualize o rule trigger (acionador de regra) com base em uma data específica ou na idade do objeto.

Excluir uma regra de ciclo de vida

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts buckets (Buckets do Outposts).
3. Escolha o bucket do Outposts para o qual você deseja excluir uma regra de ciclo de vida.
4. Escolha a guia Management (Gerenciamento) e escolha a Lifecycle rule (Regra de ciclo de vida) que você deseja excluir.
5. Escolha Delete (Excluir).

Criar e gerenciar uma configuração de ciclo de vida usando a AWS CLI e o SDK para Java

As regras de ciclo de vida dos buckets do Amazon S3 on Outposts são limitadas à exclusão de objetos. Você pode usar regras de ciclo de vida para definir quando iniciar a exclusão de objetos com base na idade ou na data. Você pode criar, habilitar, desabilitar e excluir uma regra de ciclo de vida.

Note

A Conta da AWS que cria o bucket tem propriedade sobre ele e é a única que pode criar, habilitar, desabilitar e excluir uma regra de ciclo de vida.

Para criar e gerenciar uma configuração de ciclo de vida de um bucket do S3 on Outposts usando a AWS Command Line Interface (AWS CLI) e o AWS SDK for Java, consulte os exemplos a seguir.

Tópicos

- [Executar PUT em uma configuração de ciclo de vida \(p. 1348\)](#)
- [Executar GET na configuração de ciclo de vida em um bucket do S3 on Outposts \(p. 1349\)](#)

Executar PUT em uma configuração de ciclo de vida

AWS CLI

O exemplo a seguir da AWS CLI coloca uma política de configuração de ciclo de vida em um bucket do Outposts. Essa política especifica que todos os objetos que têm o prefixo sinalizado (*myprefix*) e tags expiram após dez dias. Para usar esse exemplo, substitua cada *user input placeholder* por suas próprias informações.

1. Salve a política da configuração do ciclo de vida em um arquivo JSON. Neste exemplo, o nome do arquivo é `lifecycle1.json`.

```
{  
    "Rules": [  
        {  
            "ID": "id-1",  
            "Filter": {  
                "And": [  
                    {"Prefix": "myprefix",  
                     "Tags": [  
                        {"Value": "mytagvalue1",  
                         "Key": "mytagkey1"},  
                        {"Value": "mytagvalue2",  
                         "Key": "mytagkey2"}  
                    ]  
                ]  
            },  
            "Status": "Enabled",  
            "Expiration": {  
                "Days": 10  
            }  
        }  
    ]  
}S3OutpostsPutBucketLifecycleConfigurationCLI
```

2. Envie o arquivo JSON como parte do comando `put-bucket-lifecycle-configuration` da CLI. Para usar esse comando, substitua cada *user input placeholder* por suas próprias informações. Para obter mais informações sobre esse comando, consulte `put-bucket-lifecycle-configuration` na Referência da AWS CLI.

```
aws s3control put-bucket-lifecycle-configuration --account-id 123456789012 --  
bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/  
bucket/example-outpost-bucket --lifecycle-configuration file://lifecycle1.json
```

SDK for Java

O exemplo a seguir do SDK para Java coloca uma configuração de ciclo de vida em um bucket do Outposts. Essa configuração de ciclo de vida especifica que todos os objetos que têm o prefixo sinalizado (*myprefix*) e tags expiram após dez dias. Para usar esse exemplo, substitua cada *user input placeholder* por suas próprias informações. Para obter mais informações, consulte `PutBucketLifecycleConfiguration` na Referência da API do Amazon Simple Storage Service.

```
import com.amazonaws.services.s3control.model.*;  
  
public void putBucketLifecycleConfiguration(String bucketArn) {
```

```
S3Tag tag1 = new S3Tag().withKey("mytagkey1").withValue("mytagvalue1");
S3Tag tag2 = new S3Tag().withKey("mytagkey2").withValue("mytagvalue2");

LifecycleRuleFilter lifecycleRuleFilter = new LifecycleRuleFilter()
    .withAnd(new LifecycleRuleAndOperator()
        .withPrefix("myprefix")
        .withTags(tag1, tag2));

LifecycleExpiration lifecycleExpiration = new LifecycleExpiration()
    .withExpiredObjectDeleteMarker(false)
    .withDays(10);

LifecycleRule lifecycleRule = new LifecycleRule()
    .withStatus("Enabled")
    .withFilter(lifecycleRuleFilter)
    .withExpiration(lifecycleExpiration)
    .withID("id-1");

LifecycleConfiguration lifecycleConfiguration = new LifecycleConfiguration()
    .withRules(lifecycleRule);

PutBucketLifecycleConfigurationRequest reqPutBucketLifecycle = new
PutBucketLifecycleConfigurationRequest()
    .withAccountId(AccountId)
    .withBucket(bucketArn)
    .withLifecycleConfiguration(lifecycleConfiguration);

PutBucketLifecycleConfigurationResult respPutBucketLifecycle =
s3ControlClient.putBucketLifecycleConfiguration(reqPutBucketLifecycle);
System.out.printf("PutBucketLifecycleConfiguration Response: %s%n",
respPutBucketLifecycle.toString());

}
```

Executar GET na configuração de ciclo de vida em um bucket do S3 on Outposts

AWS CLI

O seguinte exemplo da AWS CLI obtém uma configuração de ciclo de vida em um bucket do Outposts. Para usar esse comando, substitua cada `user input placeholder` por suas próprias informações. Para obter mais informações sobre esse comando, consulte [get-bucket-lifecycle-configuration](#) na Referência da AWS CLI.

```
aws s3control get-bucket-lifecycle-configuration --account-id 123456789012 --bucket
arn:aws:s3-outposts:<your-region>:123456789012:outpost/op-01ac5d28a6a232904/
bucket/example-outpost-bucket
```

SDK for Java

O exemplo a seguir do SDK para Java obtém uma configuração de ciclo de vida de um bucket do Outposts. Para obter mais informações, consulte [GetBucketLifecycleConfiguration](#) na Referência da API do Amazon Simple Storage Service.

```
import com.amazonaws.services.s3control.model.*;

public void getBucketLifecycleConfiguration(String bucketArn) {

    GetBucketLifecycleConfigurationRequest reqGetBucketLifecycle = new
    GetBucketLifecycleConfigurationRequest()
```

```
        .withAccountId(AccountId)
        .withBucket(bucketArn);

    GetBucketLifecycleConfigurationResult respGetBucketLifecycle =
s3ControlClient.getBucketLifecycleConfiguration(reqGetBucketLifecycle);
    System.out.printf("GetBucketLifecycleConfiguration Response: %s%n",
respGetBucketLifecycle.toString());

}
```

Gerenciar o acesso a um bucket do Amazon S3 on Outposts usando uma política de bucket

Uma política de bucket é baseada em recursos do AWS Identity and Access Management (IAM) que você pode usar para conceder permissões de acesso ao bucket e aos objetos nele contidos. Só o proprietário do bucket pode associar uma política a um bucket. As permissões anexadas ao bucket se aplicam a todos os objetos do bucket que pertencem ao proprietário do bucket. As políticas de bucket são limitadas a 20 KB. Para obter mais informações, consulte [Política de bucket \(p. 1320\)](#).

Você pode atualizar sua política de bucket para gerenciar o acesso ao bucket do Amazon S3 on Outposts. Para obter mais informações, consulte os tópicos a seguir.

Tópicos

- [Adicionar ou editar uma política para um bucket do Amazon S3 on Outposts \(p. 1350\)](#)
- [Visualizar a política do bucket do Amazon S3 on Outposts \(p. 1351\)](#)
- [Excluir a política do bucket do Amazon S3 on Outposts \(p. 1352\)](#)

Adicionar ou editar uma política para um bucket do Amazon S3 on Outposts

Uma política de bucket é baseada em recursos do AWS Identity and Access Management (IAM) que você pode usar para conceder permissões de acesso ao bucket e aos objetos nele contidos. Só o proprietário do bucket pode associar uma política a um bucket. As permissões anexadas ao bucket se aplicam a todos os objetos do bucket que pertencem ao proprietário do bucket. As políticas de bucket são limitadas a 20 KB. Para obter mais informações, consulte [Política de bucket \(p. 1320\)](#).

Os tópicos a seguir mostram como atualizar sua política de bucket do Amazon S3 on Outposts usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) ou o AWS SDK for Java.

Uso do console do S3

Para criar ou editar uma política de bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts buckets (Buckets do Outposts).
3. Escolha o bucket do Outposts cuja política de bucket você deseja editar.
4. Escolha a guia Permissions (Permissões).
5. Na seção Outposts bucket policy (Política de bucket do Outposts), para criar ou editar uma nova política, escolha Edit (Editar).

Agora, você pode adicionar ou editar a política de bucket do S3 on Outposts. Para obter mais informações, consulte [Configurar o IAM com o S3 on Outposts \(p. 1327\)](#).

Usar a AWS CLI

O exemplo da AWS CLI a seguir coloca uma política em um bucket do Outposts.

1. Salve a política de bucket a seguir em um arquivo JSON. Neste exemplo, o nome do arquivo é `policy1.json`. Substitua os `user input placeholders` por suas próprias informações.

```
{  
    "Version": "2012-10-17",  
    "Id": "testBucketPolicy",  
    "Statement": [  
        {  
            "Sid": "st1",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "123456789012"  
            },  
            "Action": "s3-outposts:*",  
            "Resource": "arn:aws:s3-  
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-  
bucket"  
        }  
    ]  
}
```

2. Envie o arquivo JSON como parte do comando `put-bucket-policy` da CLI. Para executar esse comando, substitua os `user input placeholders` por suas próprias informações.

```
aws s3control put-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-  
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket  
--policy file://policy1.json
```

Usar o AWS SDK for Java

O exemplo do SDK for Java a seguir coloca uma política em um bucket do Outposts.

```
import com.amazonaws.services.s3control.model.*;  
  
public void putBucketPolicy(String bucketArn) {  
  
    String policy = "{\"Version\": \"2012-10-17\", \"Id\": \"testBucketPolicy\", \"Statement\":" +  
    \"\": [{\"Sid\": \"st1\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\": \"\" + AccountId + \"\"},  
    \"Action\": \"s3-outposts:*\", \"Resource\": \"\" + bucketArn + \"\"}]}";  
  
    PutBucketPolicyRequest reqPutBucketPolicy = new PutBucketPolicyRequest()  
        .withAccountId(AccountId)  
        .withBucket(bucketArn)  
        .withPolicy(policy);  
  
    PutBucketPolicyResult respPutBucketPolicy =  
    s3ControlClient.putBucketPolicy(reqPutBucketPolicy);  
    System.out.printf("PutBucketPolicy Response: %s%n", respPutBucketPolicy.toString());  
}
```

Visualizar a política do bucket do Amazon S3 on Outposts

Uma política de bucket é baseada em recursos do AWS Identity and Access Management (IAM) que você pode usar para conceder permissões de acesso ao bucket e aos objetos nele contidos. Só o proprietário do bucket pode associar uma política a um bucket. As permissões anexadas ao bucket se aplicam a todos

os objetos do bucket que pertencem ao proprietário do bucket. As políticas de bucket são limitadas a 20 KB. Para obter mais informações, consulte [Política de bucket \(p. 1320\)](#).

Os tópicos a seguir mostram como visualizar sua política de bucket do Amazon S3 on Outposts usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) ou o AWS SDK for Java.

Uso do console do S3

Para criar ou editar uma política de bucket

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts buckets (Buckets do Outposts).
3. Escolha o bucket do Outposts cuja permissão você deseja editar.
4. Escolha a guia Permissions.
5. Na seção Outposts bucket policy (Política de bucket do Outposts), você pode revisar sua política de bucket existente. Para obter mais informações, consulte [Configurar o IAM com o S3 on Outposts \(p. 1327\)](#).

Usar a AWS CLI

O exemplo da AWS CLI a seguir obtém uma política para um bucket do Outposts. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control get-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket
```

Usar o AWS SDK for Java

O exemplo do SDK for Java a seguir obtém uma política para um bucket do Outposts.

```
import com.amazonaws.services.s3control.model.*;

public void getBucketPolicy(String bucketArn) {

    GetBucketPolicyRequest reqGetBucketPolicy = new GetBucketPolicyRequest()
        .withAccountId(AccountId)
        .withBucket(bucketArn);

    GetBucketPolicyResult respGetBucketPolicy =
        s3ControlClient.getBucketPolicy(reqGetBucketPolicy);
    System.out.printf("GetBucketPolicy Response: %s%n", respGetBucketPolicy.toString());
}
```

Excluir a política do bucket do Amazon S3 on Outposts

Uma política de bucket é baseada em recursos do AWS Identity and Access Management (IAM) que você pode usar para conceder permissões de acesso ao bucket e aos objetos nele contidos. Só o proprietário do bucket pode associar uma política a um bucket. As permissões anexadas ao bucket se aplicam a todos os objetos do bucket que pertencem ao proprietário do bucket. As políticas de bucket são limitadas a 20 KB. Para obter mais informações, consulte [Política de bucket \(p. 1320\)](#).

Os tópicos a seguir mostram como visualizar sua política de bucket do Amazon S3 on Outposts usando o AWS Management Console ou a AWS Command Line Interface (AWS CLI).

Uso do console do S3

Para excluir uma política de bucket

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts buckets (Buckets do Outposts).
3. Escolha o bucket do Outposts cuja permissão você deseja editar.
4. Escolha a guia Permissions.
5. Na seção Outposts bucket policy (Política de bucket do Outposts), escolha Delete (Excluir).
6. Confirme a exclusão.

Uso do AWS CLI

O exemplo a seguir exclui a política de um bucket do S3 on Outposts (`s3-outposts>DeleteBucket`) usando a AWS CLI. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control delete-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket
```

Listar buckets do Amazon S3 on Outposts

Com o Amazon S3 on Outposts, é possível criar buckets do S3 no AWS Outposts, além de armazenar e recuperar facilmente objetos no local para aplicações que exigem acesso a dados locais, processamento de dados local e residência de dados. O S3 on Outposts fornece uma nova classe de armazenamento, o S3 Outposts (OUTPOSTS), que usa as APIs do Amazon S3 e é projetado para armazenar dados de forma duradoura e redundante em vários dispositivos e servidores em seu AWS Outposts. Você se comunica com o bucket do Outposts usando um ponto de acesso e uma conexão de endpoint em uma nuvem privada virtual (VPC). É possível usar os mesmos recursos e APIs nos buckets do Outposts da mesma maneira que em buckets do Amazon S3, incluindo políticas de acesso, criptografia e marcação. Só é possível usar o S3 on Outposts por meio do AWS Management Console, da AWS Command Line Interface (AWS CLI), de AWS SDKs ou da API REST. Para obter mais informações, consulte [O que é o Amazon S3 on Outposts? \(p. 1318\)](#)

Para obter mais informações sobre como trabalhar com buckets no S3 on Outposts, consulte [Trabalhar com buckets do S3 on Outposts \(p. 1339\)](#).

Os exemplos a seguir mostram como retornar uma lista dos buckets do S3 on Outposts usando o AWS Management Console, a AWS CLI e o AWS SDK for Java.

Uso do console do S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts buckets (Buckets do Outposts).
3. Em Outposts buckets (Buckets do Outposts), revise sua lista de buckets do S3 on Outposts.

Uso do AWS CLI

O exemplo a seguir da AWS CLI obtém uma lista de buckets em um Outpost. Para usar esse comando, substitua cada *user input placeholder* por suas próprias informações. Para obter mais informações sobre esse comando, consulte [list-regional-buckets](#) na Referência da AWS CLI.

```
aws s3control list-regional-buckets --account-id 123456789012 --outpost-id op-01ac5d28a6a232904
```

Usar o AWS SDK para Java

O exemplo do SDK para Java a seguir obtém uma lista de buckets em um Outpost. Para obter mais informações, consulte [ListRegionalBuckets](#) na Referência da API do Amazon Simple Storage Service.

```
import com.amazonaws.services.s3control.model.*;  
  
public void listRegionalBuckets() {  
  
    ListRegionalBucketsRequest reqListBuckets = new ListRegionalBucketsRequest()  
        .withAccountId(AccountId)  
        .withOutpostId(OutpostId);  
  
    ListRegionalBucketsResult respListBuckets =  
    s3ControlClient.listRegionalBuckets(reqListBuckets);  
    System.out.printf("ListRegionalBuckets Response: %s%n", respListBuckets.toString());  
}
```

Obter um bucket do S3 on Outposts usando a AWS CLI e o SDK para Java

Com o Amazon S3 on Outposts, é possível criar buckets do S3 no AWS Outposts, além de armazenar e recuperar facilmente objetos no local para aplicações que exigem acesso a dados locais, processamento de dados local e residência de dados. O S3 on Outposts fornece uma nova classe de armazenamento, o S3 Outposts (OUTPOSTS), que usa as APIs do Amazon S3 e é projetado para armazenar dados de forma duradoura e redundante em vários dispositivos e servidores em seu AWS Outposts. Você se comunica com o bucket do Outposts usando um ponto de acesso e uma conexão de endpoint em uma nuvem privada virtual (VPC). É possível usar os mesmos recursos e APIs nos buckets do Outposts da mesma maneira que em buckets do Amazon S3, incluindo políticas de acesso, criptografia e marcação. Só é possível usar o S3 on Outposts por meio do AWS Management Console, da AWS Command Line Interface (AWS CLI), de AWS SDKs ou da API REST. Para obter mais informações, consulte [O que é o Amazon S3 on Outposts? \(p. 1318\)](#)

Os exemplos a seguir mostram como obter um bucket do S3 on Outposts usando a AWS CLI e o AWS SDK for Java.

Note

Ao trabalhar com o Amazon S3 on Outposts por meio da AWS CLI ou de AWSSDKs, forneça o ARN do ponto de acesso do Outpost no lugar do nome do bucket. O ARN do ponto de acesso assume a forma a seguir, em que `region` é o código da Região da AWS em que se encontra o Outpost:

```
arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/  
accesspoint/example-outposts-access-point
```

Para obter mais informações sobre o S3 on Outposts, consulte [ARNs de recurso para S3 no Outposts \(p. 1328\)](#).

Uso do AWS CLI

O exemplo do S3 on Outposts a seguir obtém um bucket usando a AWS CLI. Para usar esse comando, substitua cada `user input placeholder` por suas próprias informações. Para obter mais informações sobre esse comando, consulte [get-bucket](#) na Referência da AWS CLI.

```
aws s3control get-bucket --account-id 123456789012 --bucket "arn:aws:s3-  
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket"
```

Usar o AWS SDK para Java

O exemplo do S3 no Outposts a seguir obtém um bucket usando o SDK para Java. Para obter mais informações, consulte [GetBucket](#) na Referência da API do Amazon Simple Storage Service.

```
import com.amazonaws.services.s3control.model.*;  
  
public void getBucket(String bucketArn) {  
  
    GetBucketRequest reqGetBucket = new GetBucketRequest()  
        .withBucket(bucketArn)  
        .withAccountId(AccountId);  
  
    GetBucketResult respGetBucket = s3ControlClient.getBucket(reqGetBucket);  
    System.out.printf("GetBucket Response: %s%n", respGetBucket.toString());  
}
```

Excluir seu bucket do S3 on Outposts

Com o Amazon S3 on Outposts, é possível criar buckets do S3 no AWS Outposts, além de armazenar e recuperar facilmente objetos no local para aplicações que exigem acesso a dados locais, processamento de dados local e residência de dados. O S3 on Outposts fornece uma nova classe de armazenamento, o S3 Outposts (OUTPOSTS), que usa as APIs do Amazon S3 e é projetado para armazenar dados de forma duradoura e redundante em vários dispositivos e servidores em seu AWS Outposts. Você se comunica com o bucket do Outposts usando um ponto de acesso e uma conexão de endpoint em uma nuvem privada virtual (VPC). É possível usar os mesmos recursos e APIs nos buckets do Outposts da mesma maneira que em buckets do Amazon S3, incluindo políticas de acesso, criptografia e marcação. Só é possível usar o S3 on Outposts por meio do AWS Management Console, da AWS Command Line Interface (AWS CLI), de AWS SDKs ou da API REST. Para obter mais informações, consulte [O que é o Amazon S3 on Outposts? \(p. 1318\)](#)

Para obter mais informações sobre como trabalhar com buckets no S3 on Outposts, consulte [Trabalhar com buckets do S3 on Outposts \(p. 1339\)](#).

A Conta da AWS que cria o bucket é proprietária dele e é a única que pode excluí-lo.

Note

- Os buckets do Outposts devem estar vazios antes de serem excluídos.

O console do Amazon S3 não é compatível com ações de objeto do S3 on Outposts. Para excluir objetos em um bucket do S3 on Outposts, você deve usar a API REST, a AWS CLI ou os AWS SDKs.

- Antes de excluir um bucket do Outposts, você deve excluir os pontos de acesso do Outposts para o bucket. Para obter mais informações, consulte [Excluir um ponto de acesso \(p. 1359\)](#).
- Você não pode recuperar um bucket depois que ele foi excluído.

Os exemplos a seguir mostram como excluir um bucket do S3 on Outposts usando o AWS Management Console e a AWS Command Line Interface (AWS CLI).

Uso do console do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts buckets (Buckets do Outposts).

3. Escolha o bucket que você deseja excluir e escolha Delete (Excluir).
4. Confirme a exclusão.

Uso do AWS CLI

O exemplo a seguir exclui um bucket do S3 on Outposts (`s3-outposts>DeleteBucket`) usando a AWS CLI. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control delete-bucket --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket
```

Trabalhar com pontos de acesso do Amazon S3 on Outposts

Para acessar o bucket do Amazon S3 on Outposts, você deve criar e configurar um ponto de acesso.

Os pontos de acesso simplificam o gerenciamento do acesso a dados em escala para conjuntos de dados compartilhados no Amazon S3. Os pontos de acesso são endpoints de rede nomeados e anexados a buckets que você pode usar para realizar operações de objeto do Amazon S3, como `GetObject` e `PutObject`. Com o S3 on Outposts, você deve usar pontos de acesso para acessar qualquer objeto em um bucket do Outposts. Os pontos de acesso são compatíveis apenas com o endereçamento em estilo de host virtual.

Note

A Conta da AWS que cria o bucket do Outposts é proprietária dele e é a única que pode atribuir pontos de acesso a ele.

As seções a seguir descrevem como criar e gerenciar pontos de acesso para buckets do S3 on Outposts.

Tópicos

- [Criar um ponto de acesso do S3 on Outposts \(p. 1356\)](#)
- [Exibir informações sobre uma configuração de ponto de acesso \(p. 1358\)](#)
- [Visualizar uma lista dos pontos de acesso do Amazon S3 on Outposts \(p. 1358\)](#)
- [Excluir um ponto de acesso \(p. 1359\)](#)
- [Adicionar ou editar uma política de ponto de acesso \(p. 1360\)](#)
- [Visualizar uma política para um ponto de acesso do S3 on Outposts \(p. 1361\)](#)

Criar um ponto de acesso do S3 on Outposts

Para acessar o bucket do Amazon S3 on Outposts, você deve criar e configurar um ponto de acesso.

Os pontos de acesso simplificam o gerenciamento do acesso a dados em escala para conjuntos de dados compartilhados no Amazon S3. Os pontos de acesso são endpoints de rede nomeados e anexados a buckets que você pode usar para realizar operações de objeto do Amazon S3, como `GetObject` e `PutObject`. Com o S3 on Outposts, você deve usar pontos de acesso para acessar qualquer objeto em um bucket do Outposts. Os pontos de acesso são compatíveis apenas com o endereçamento em estilo de host virtual.

Os exemplos a seguir mostram como criar um ponto de acesso do S3 on Outposts usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) e o AWS SDK for Java.

Note

A Conta da AWS que cria o bucket do Outposts é proprietária dele e é a única que pode atribuir pontos de acesso a ele.

Uso do console do S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts buckets (Buckets do Outposts).
3. Escolha o bucket do Outposts para o qual deseja criar um ponto de acesso do Outposts.
4. Escolha a aba Outposts access points (Pontos de acesso do Outposts).
5. Na seção Outposts access points (Pontos de acesso do Outposts), escolha Create Outposts access point (Criar ponto de acesso do Outposts).
6. Na seção Outposts access point settings (Configurações do ponto de acesso do Outposts), insira um nome para o ponto de acesso e escolha a nuvem privada virtual (VPC) dele.
7. Se você quiser adicionar uma política para o seu ponto de acesso, adicione-a na seção Outposts access point policy (Política de ponto de acesso do Outposts).

Para obter mais informações, consulte [Configurar o IAM com o S3 on Outposts \(p. 1327\)](#).

Usar a AWS CLI

Example

O exemplo da AWS CLI a seguir cria um ponto de acesso para um bucket do Outposts. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control create-access-point --account-id 123456789012 --name example-outposts-access-point --bucket "arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket" --vpc-configuration VpcId=example-vpc-12345
```

Usar o AWS SDK for Java

Example

O exemplo do SDK for Java a seguir cria um ponto de acesso para um bucket do Outposts. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
import com.amazonaws.services.s3control.model.*;

public String createAccessPoint(String bucketArn, String accessPointName) {

    CreateAccessPointRequest reqCreateAP = new CreateAccessPointRequest()
        .withAccountId(AccountId)
        .withBucket(bucketArn)
        .withName(accessPointName)
        .withVpcConfiguration(new VpcConfiguration().withVpcId("vpc-12345"));

    CreateAccessPointResult respCreateAP = s3ControlClient.createAccessPoint(reqCreateAP);
    System.out.printf("CreateAccessPoint Response: %s%n", respCreateAP.toString());

    return respCreateAP.getAccessPointArn();
}
```

Exibir informações sobre uma configuração de ponto de acesso

Os pontos de acesso simplificam o gerenciamento do acesso a dados em escala para conjuntos de dados compartilhados no Amazon S3. Os pontos de acesso são endpoints de rede nomeados e anexados a buckets que você pode usar para realizar operações de objeto do Amazon S3, como `GetObject` e `PutObject`. Com o S3 on Outposts, você deve usar pontos de acesso para acessar qualquer objeto em um bucket do Outposts. Os pontos de acesso são compatíveis apenas com o endereçamento em estilo de host virtual.

Os tópicos a seguir mostram como retornar informações de configuração para um ponto de acesso do S3 on Outposts usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) e o AWS SDK for Java.

Uso do console do S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts access points (Pontos de acesso do Outposts).
3. Escolha o ponto de acesso do Outposts para o qual deseja exibir os detalhes da configuração.
4. Em Outposts access point overview (Visão geral do ponto de acesso do Outposts), examine os detalhes da configuração do ponto de acesso.

Uso do AWS CLI

O exemplo da AWS CLI a seguir obtém um ponto de acesso para um bucket do Outposts. Substitua os *user input placeholders* por suas próprias informações.

```
aws s3control get-access-point --account-id 123456789012 --name arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

Usar o AWS SDK for Java

O exemplo do SDK for Java a seguir obtém um ponto de acesso para um bucket do Outposts.

```
import com.amazonaws.services.s3control.model.*;  
  
public void getAccessPoint(String accessPointArn) {  
  
    GetAccessPointRequest reqGetAP = new GetAccessPointRequest()  
        .withAccountId(AccountId)  
        .withName(accessPointArn);  
  
    GetAccessPointResult respGetAP = s3ControlClient.getAccessPoint(reqGetAP);  
    System.out.printf("GetAccessPoint Response: %s%n", respGetAP.toString());  
}
```

Visualizar uma lista dos pontos de acesso do Amazon S3 on Outposts

Os pontos de acesso simplificam o gerenciamento do acesso a dados em escala para conjuntos de dados compartilhados no Amazon S3. Os pontos de acesso são endpoints de rede nomeados e anexados a buckets que você pode usar para realizar operações de objeto do Amazon S3, como `GetObject` e `PutObject`. Com o S3 on Outposts, você deve usar pontos de acesso para acessar qualquer objeto em

um bucket do Outposts. Os pontos de acesso são compatíveis apenas com o endereçamento em estilo de host virtual.

Os tópicos a seguir mostram como retornar uma lista de seus pontos de acesso do S3 on Outposts usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) e o AWS SDK for Java.

Uso do console do S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts access points (Pontos de acesso do Outposts).
3. Em Outposts access points (Pontos de acesso do Outposts), analise sua lista de pontos de acesso do S3 on Outposts.

Uso do AWS CLI

O exemplo da AWS CLI a seguir lista os pontos de acesso de um bucket do Outposts. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control list-access-points --account-id 123456789012 --bucket arn:aws:s3-  
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket
```

Usar o AWS SDK for Java

O exemplo do SDK for Java a seguir lista os pontos de acesso para um bucket do Outposts.

```
import com.amazonaws.services.s3control.model.*;  
  
public void listAccessPoints(String bucketArn) {  
  
    ListAccessPointsRequest reqListAPs = new ListAccessPointsRequest()  
        .withAccountId(AccountId)  
        .withBucket(bucketArn);  
  
    ListAccessPointsResult respListAPs = s3ControlClient.listAccessPoints(reqListAPs);  
    System.out.printf("ListAccessPoints Response: %s%n", respListAPs.toString());  
}
```

Excluir um ponto de acesso

Os pontos de acesso simplificam o gerenciamento do acesso a dados em escala para conjuntos de dados compartilhados no Amazon S3. Os pontos de acesso são endpoints de rede nomeados e anexados a buckets que você pode usar para realizar operações de objeto do Amazon S3, como `GetObject` e `PutObject`. Com o S3 on Outposts, você deve usar pontos de acesso para acessar qualquer objeto em um bucket do Outposts. Os pontos de acesso são compatíveis apenas com o endereçamento em estilo de host virtual.

Os exemplos a seguir mostram como excluir um ponto de acesso usando o AWS Management Console e a AWS Command Line Interface (AWS CLI).

Uso do console do S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts access points (Pontos de acesso do Outposts).
3. Na seção Outposts access points (Pontos de acesso do Outposts), escolha o ponto de acesso do Outposts que deseja excluir.

4. Escolha Delete.
5. Confirme a exclusão.

Uso do AWS CLI

O exemplo da AWS CLI a seguir exclui um ponto de acesso do Outposts. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control delete-access-point --account-id 123456789012 --name arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

Adicionar ou editar uma política de ponto de acesso

Cada ponto de acesso tem permissões e controles de rede distintos que o Amazon S3 on Outposts aplica a qualquer solicitação feita por meio desse ponto de acesso. Cada ponto de acesso impõe uma política de ponto de acesso personalizada que funciona em conjunto com a política de bucket anexada ao bucket subjacente. Para obter mais informações, consulte [Pontos de acesso \(p. 1340\)](#).

Os tópicos a seguir mostram como adicionar ou editar a política de ponto de acesso do S3 on Outposts usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) e o AWS SDK for Java.

Uso do console do S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts buckets (Buckets do Outposts).
3. Escolha o bucket do Outposts para o qual você deseja editar a política de ponto de acesso.
4. Escolha a aba Outposts access points (Pontos de acesso do Outposts).
5. Na seção Outposts access points (Pontos de acesso do Outposts), selecione o ponto de acesso cuja política você quer editar e escolha Edit policy (Editar política).
6. Adicione ou edite a política na seção da Outposts access point policy (Política do ponto de acesso do Outposts). Para obter mais informações, consulte [Configurar o IAM com o S3 on Outposts \(p. 1327\)](#).

Usar a AWS CLI

O exemplo da AWS CLI a seguir coloca uma política para um bucket do Outposts.

1. Salve a política de ponto de acesso a seguir em um arquivo JSON. Neste exemplo, o nome do arquivo é `appolicy1.json`. Substitua os *user input placeholders* por suas próprias informações.

```
{
    "Version": "2012-10-17",
    "Id": "exampleAccessPointPolicy",
    "Statement": [
        {
            "Sid": "st1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "123456789012"
            },
            "Action": "s3-outposts:*",
            "Resource": "arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point"
        }
    ]
}
```

}

2. Envie o arquivo JSON como parte do comando `put-access-point-policy` da CLI. Substitua os `user input placeholders` por suas próprias informações.

```
aws s3control put-access-point-policy --account-id 123456789012 --name arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point --policy file://appolicy1.json
```

Usar o AWS SDK for Java

O exemplo do SDK for Java a seguir coloca uma política em um ponto de acesso do Outposts.

```
import com.amazonaws.services.s3control.model.*;

public void putAccessPointPolicy(String accessPointArn) {

    String policy = "{\"Version\":\"2012-10-17\", \"Id\":\"testAccessPointPolicy\",
\"Statement\":[{\"Sid\":\"st1\", \"Effect\":\"Allow\", \"Principal\":{\"AWS\":\"\" + AccountId
+ "\"\"}, \"Action\":\"s3-outposts:*\", \"Resource\":\"\" + accessPointArn + "\"\"]}";

    PutAccessPointPolicyRequest reqPutAccessPointPolicy = new PutAccessPointPolicyRequest()
        .withAccountId(AccountId)
        .withName(accessPointArn)
        .withPolicy(policy);

    PutAccessPointPolicyResult respPutAccessPointPolicy =
s3ControlClient.putAccessPointPolicy(reqPutAccessPointPolicy);
    System.out.printf("PutAccessPointPolicy Response: %s%n",
respPutAccessPointPolicy.toString());
    printWriter.printf("PutAccessPointPolicy Response: %s%n",
respPutAccessPointPolicy.toString());
}
```

Visualizar uma política para um ponto de acesso do S3 on Outposts

Cada ponto de acesso tem permissões e controles de rede distintos que o Amazon S3 on Outposts aplica a qualquer solicitação feita por meio desse ponto de acesso. Cada ponto de acesso impõe uma política de ponto de acesso personalizada que funciona em conjunto com a política de bucket anexada ao bucket subjacente. Para obter mais informações, consulte [Pontos de acesso \(p. 1340\)](#).

Para obter mais informações sobre como trabalhar com pontos de acesso no S3 on Outposts, consulte [Trabalhar com buckets do S3 on Outposts \(p. 1339\)](#).

Os tópicos a seguir mostram como visualizar ou editar a política de ponto de acesso do S3 on Outposts usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) e o AWS SDK for Java.

Uso do console do S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts access points (Pontos de acesso do Outposts).
3. Escolha o ponto de acesso do Outposts para o qual você deseja visualizar a política.
4. Na guia Permissions (Permissões), analise a política do ponto de acesso do S3 on Outposts.
5. Para editar uma política de ponto de acesso, consulte [Adicionar ou editar uma política de ponto de acesso \(p. 1360\)](#).

Uso do AWS CLI

O exemplo da AWS CLI a seguir obtém uma política para um ponto de acesso do Outposts. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control get-access-point-policy --account-id 123456789012 --name arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

Usar o AWS SDK for Java

O exemplo do SDK for Java a seguir obtém uma política para um ponto de acesso do Outposts.

```
import com.amazonaws.services.s3control.model.*;

public void getAccessPointPolicy(String accessPointArn) {

    GetAccessPointPolicyRequest reqGetAccessPointPolicy = new GetAccessPointPolicyRequest()
        .withAccountId(AccountId)
        .withName(accessPointArn);

    GetAccessPointPolicyResult respGetAccessPointPolicy =
        s3ControlClient.getAccessPointPolicy(reqGetAccessPointPolicy);
    System.out.printf("GetAccessPointPolicy Response: %s%n",
        respGetAccessPointPolicy.toString());
    printWriter.printf("GetAccessPointPolicy Response: %s%n",
        respGetAccessPointPolicy.toString());

}
```

Trabalhar com endpoints do Amazon S3 on Outposts

Para rotear solicitações para um ponto de acesso do Amazon S3 on Outposts, você deve criar e configurar um endpoint do S3 on Outposts. Cada nuvem privada virtual (VPC) em seu Outpost pode ter um endpoint associado. Para obter mais informações sobre cotas de endpoints, consulte [Requisitos de rede do S3 on Outposts \(p. 1326\)](#). Você deve criar esses endpoints para poder acessar seus buckets do Outposts e executar operações de objeto. Para obter mais informações, consulte [Endpoints \(p. 1340\)](#).

Para obter mais informações sobre como trabalhar com buckets no S3 on Outposts, consulte [Trabalhar com buckets do S3 on Outposts \(p. 1339\)](#).

As seções a seguir descrevem como criar e gerenciar endpoints para o S3 on Outposts.

Tópicos

- [Criar um endpoint em um Outpost \(p. 1362\)](#)
- [Visualizar uma lista de endpoints do Amazon S3 on Outposts \(p. 1364\)](#)
- [Excluir um endpoint do Amazon S3 on Outposts \(p. 1365\)](#)

Criar um endpoint em um Outpost

Para rotear solicitações para um ponto de acesso do Amazon S3 on Outposts, você deve criar e configurar um endpoint do S3 on Outposts. Cada nuvem privada virtual (VPC) em seu Outpost pode ter um endpoint associado. Para obter mais informações sobre cotas de endpoints, consulte [Requisitos de rede do S3 on Outposts \(p. 1326\)](#). Você deve criar esses endpoints para poder acessar seus buckets do Outposts e executar operações de objeto. Para obter mais informações, consulte [Endpoints \(p. 1340\)](#).

Permissões

Para obter mais informações sobre as permissões necessárias para criar um endpoint, consulte [Permissões para os endpoints do S3 on Outposts \(p. 1330\)](#).

Os exemplos a seguir mostram como criar um endpoint do S3 on Outposts usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) e o AWS SDK for Java.

Uso do console do S3

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts access points (Pontos de acesso do Outposts).
3. Escolha a guia Outposts endpoints (Endpoints do Outposts).
4. Escolha Create Outposts endpoint (Criar endpoint do Outposts).
5. Em Outpost, escolha o Outpost no qual o endpoint será criado.
6. Em VPC, escolha uma VPC que ainda não tenha um endpoint e que esteja em conformidade com as regras para endpoints do Outposts.

A Virtual Private Cloud (VPC) permite iniciar recursos da AWS em uma rede virtual definida por você. Essa rede virtual se assemelha a uma rede tradicional que você operaria no seu próprio datacenter, com os benefícios de usar a infraestrutura escalável da AWS.

Se você não tiver uma VPC, escolha Create VPC (Criar VPC). Para obter mais informações, consulte [Criar pontos de acesso restritos a uma nuvem privada virtual \(p. 327\)](#).

7. Escolha Create Outposts endpoint (Criar endpoint do Outposts).

Uso do AWS CLI

Example

O seguinte exemplo da AWS CLI cria um endpoint para um Outpost usando o tipo de acesso a recursos da VPC. A VPC é derivada da sub-rede. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
subnet-8c7a57c5 --security-group-id sg-ab19e0d1
```

O exemplo a seguir da AWS CLI cria um endpoint para um Outpost usando o tipo de acesso do grupo de endereços IP de propriedade do cliente (grupo do CoIP). Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
subnet-8c7a57c5 --security-group-id sg-ab19e0d1 --access-type CustomerOwnedIp --customer-
owned-ipv4-pool ipv4pool-coip-12345678901234567
```

Usar o AWS SDK para Java

Example

O exemplo do SDK para Java a seguir cria um endpoint para um Outpost. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.CreateEndpointRequest;
import com.amazonaws.services.s3outposts.model.CreateEndpointResult;
```

```
public void createEndpoint() {
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
        .standard().build();

    CreateEndpointRequest createEndpointRequest = new CreateEndpointRequest()
        .withOutpostId("op-0d79779cef3c30a40")
        .withSubnetId("subnet-8c7a57c5")
        .withSecurityGroupId("sg-ab19e0d1")
        .withAccessType("CustomerOwnedIp")
        .withCustomerOwnedIpv4Pool("ipv4pool-coip-12345678901234567");
    // Use .withAccessType and .withCustomerOwnedIpv4Pool only when the access type is
    // customer-owned IP address pool (CoIP pool)
    CreateEndpointResult createEndpointResult =
    s3OutpostsClient.createEndpoint(createEndpointRequest);
    System.out.println("Endpoint is created and its ARN is " +
    createEndpointResult.getEndpointArn());
}
```

Visualizar uma lista de endpoints do Amazon S3 on Outposts

Para rotear solicitações para um ponto de acesso do Amazon S3 on Outposts, você deve criar e configurar um endpoint do S3 on Outposts. Cada nuvem privada virtual (VPC) em seu Outpost pode ter um endpoint associado. Para obter mais informações sobre cotas de endpoints, consulte [Requisitos de rede do S3 on Outposts \(p. 1326\)](#). Você deve criar esses endpoints para poder acessar seus buckets do Outposts e executar operações de objeto. Para obter mais informações, consulte [Endpoints \(p. 1340\)](#).

Os exemplos a seguir mostram como retornar uma lista de seus endpoints do S3 on Outposts usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) e o AWS SDK for Java.

Uso do console do S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts access points (Pontos de acesso do Outposts).
3. Na página Outposts access points (Pontos de acesso do Outposts), escolha a guia Outposts endpoints (Endpoints do Outposts).
4. Em Outposts endpoints (Endpoints do Outposts), você pode visualizar uma lista de endpoints do S3 on Outposts.

Uso do AWS CLI

O exemplo a seguir da AWS CLI lista os endpoints para os recursos do AWS Outposts associados à sua conta. Para obter mais informações sobre esse comando, consulte [list-endpoints](#) na Referência da AWS CLI.

```
aws s3outposts list-endpoints
```

Usar o AWS SDK para Java

O exemplo do SDK para Java a seguir lista os endpoints para um Outpost. Para obter mais informações, consulte [ListEndpoints](#) na Referência da API do Amazon Simple Storage Service.

```
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.ListEndpointsRequest;
import com.amazonaws.services.s3outposts.model.ListEndpointsResult;

public void listEndpoints() {
```

```
AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
    .standard().build();

ListEndpointsRequest listEndpointsRequest = new ListEndpointsRequest();
ListEndpointsResult listEndpointsResult =
s3OutpostsClient.listEndpoints(listEndpointsRequest);
System.out.println("List endpoints result is " + listEndpointsResult);
}
```

Excluir um endpoint do Amazon S3 on Outposts

Para rotear solicitações para um ponto de acesso do Amazon S3 on Outposts, você deve criar e configurar um endpoint do S3 on Outposts. Cada nuvem privada virtual (VPC) em seu Outpost pode ter um endpoint associado. Para obter mais informações sobre cotas de endpoints, consulte [Requisitos de rede do S3 on Outposts \(p. 1326\)](#). Você deve criar esses endpoints para poder acessar seus buckets do Outposts e executar operações de objeto. Para obter mais informações, consulte [Endpoints \(p. 1340\)](#).

Os exemplos a seguir mostram como excluir os endpoints do S3 on Outposts usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) e o AWS SDK for Java.

Uso do console do S3

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts access points (Pontos de acesso do Outposts).
3. Na página Outposts access points (Pontos de acesso do Outposts), escolha a guia Outposts endpoints (Endpoints do Outposts).
4. Em Outposts endpoints (Endpoints do Outposts), escolha o endpoint que você deseja excluir e escolha Delete (Excluir).

Uso do AWS CLI

O exemplo da AWS CLI a seguir exclui um endpoint de um Outpost. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3outposts delete-endpoint --endpoint-id example-endpoint-id --outpost-
id op-01ac5d28a6a232904
```

Usar o AWS SDK for Java

O exemplo do SDK for Java a seguir exclui um endpoint de um Outpost. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
import com.amazonaws.arn.Arns;
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.DeleteEndpointRequest;

public void deleteEndpoint(String endpointArnInput) {
    String outpostId = "op-01ac5d28a6a232904";
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
        .standard().build();

    Arn endpointArn = Arn.fromString(endpointArnInput);
    String[] resourceParts = endpointArn.getResource().getResource().split("/");
    String endpointId = resourceParts[resourceParts.length - 1];
    DeleteEndpointRequest deleteEndpointRequest = new DeleteEndpointRequest()
        .withEndpointId(endpointId)
        .withOutpostId(outpostId);
```

```
s3OutpostsClient.deleteEndpoint(deleteEndpointRequest);
System.out.println("Endpoint with id " + endpointId + " is deleted.");
}
```

Trabalhar com objetos usando o S3 on Outposts

Com o Amazon S3 on Outposts, é possível criar buckets do S3 no AWS Outposts, além de armazenar e recuperar facilmente objetos no local para aplicações que exigem acesso a dados locais, processamento de dados local e residência de dados. O S3 on Outposts fornece uma nova classe de armazenamento, o S3 Outposts (OUTPOSTS), que usa as APIs do Amazon S3 e é projetado para armazenar dados de forma duradoura e redundante em vários dispositivos e servidores em seu AWS Outposts. Você se comunica com o bucket do Outposts usando um ponto de acesso e uma conexão de endpoint em uma nuvem privada virtual (VPC). É possível usar os mesmos recursos e APIs nos buckets do Outposts da mesma maneira que em buckets do Amazon S3, incluindo políticas de acesso, criptografia e marcação. Só é possível usar o S3 on Outposts por meio do AWS Management Console, da AWS Command Line Interface (AWS CLI), de AWS SDKs ou da API REST.

Objetos são as entidades fundamentais armazenadas no Amazon S3 on Outposts. Cada objeto está contido em um bucket. É necessário usar pontos de acesso para acessar qualquer objeto em um bucket do Outpost. Ao especificar o bucket para operações de objeto, você usa o nome do recurso da Amazon (ARN) do ponto de acesso, que inclui o código Região da AWS da região em que o Outpost está hospedado, o ID da Conta da AWS, o ID do Outpost e o nome do ponto de acesso. O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 on Outposts em operações de objeto:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obter mais informações sobre o S3 on Outposts, consulte [ARNs de recurso para S3 no Outposts \(p. 1328\)](#).

Os ARNs de objeto usam o formato a seguir, que inclui a Região da AWS na qual está o Outpost, o ID da Conta da AWS, o ID do Outpost, o nome do bucket e a chave do objeto:

```
arn:aws:s3-outposts:us-west-2:123456789012:outpost/op-01ac5d28a6a232904/bucket/DOC-EXAMPLE-BUCKET1/object/myobject
```

Com o Amazon S3 on Outposts, os dados do objeto são sempre armazenados no Outpost. Quando a AWS instala um rack do Outpost, seus dados permanecem no local do Outpost para atender aos requisitos de residência de dados. Seus objetos nunca saem do Outpost e não estão em uma Região da AWS. Como o AWS Management Console está hospedado na região, você não pode usá-lo para fazer upload de objetos no Outpost nem os gerenciar. No entanto, você pode usar a API REST, a AWS Command Line Interface (AWS CLI) e os SDKs para fazer upload de objetos e gerenciá-los por meio de seus pontos de acesso.

Para carregar um objeto, consulte [Etapa 4: Fazer upload de um objeto em um bucket do S3 on Outposts \(p. 1336\)](#). Para obter outras ações de objeto, consulte os tópicos a seguir.

Tópicos

- [Copiar um objeto em um bucket do Amazon S3 no Outposts usando o AWS SDK for Java \(p. 1367\)](#)
- [Obter um objeto de um bucket do Amazon S3 on Outposts \(p. 1368\)](#)
- [Listar os objetos em um bucket do Amazon S3 on Outposts \(p. 1370\)](#)
- [Excluir objetos em buckets do Amazon S3 on Outposts \(p. 1372\)](#)
- [Usar o HeadBucket para determinar se existe um bucket do S3 on Outposts e se você tem permissões de acesso \(p. 1375\)](#)
- [Realizar e gerenciar um carregamento fracionado com o SDK para Java \(p. 1376\)](#)

- [Uso de URLs pré-assinados para o S3 no Outposts \(p. 1381\)](#)

Copiar um objeto em um bucket do Amazon S3 no Outposts usando o AWS SDK for Java

Objetos são as entidades fundamentais armazenadas no Amazon S3 on Outposts. Cada objeto está contido em um bucket. É necessário usar pontos de acesso para acessar qualquer objeto em um bucket do Outpost. Ao especificar o bucket para operações de objeto, você usa o nome do recurso da Amazon (ARN) do ponto de acesso, que inclui o código Região da AWS da região em que o Outpost está hospedado, o ID da Conta da AWS, o ID do Outpost e o nome do ponto de acesso. O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 on Outposts em operações de objeto:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obter mais informações sobre o S3 on Outposts, consulte [ARNs de recurso para S3 no Outposts \(p. 1328\)](#).

Com o Amazon S3 on Outposts, os dados do objeto são sempre armazenados no Outpost. Quando a AWS instala um rack do Outpost, seus dados permanecem no local do Outpost para atender aos requisitos de residência de dados. Seus objetos nunca saem do Outpost e não estão em uma Região da AWS. Como o AWS Management Console está hospedado na região, você não pode usá-lo para fazer upload de objetos no Outpost nem os gerenciar. No entanto, você pode usar a API REST, a AWS Command Line Interface (AWS CLI) e os SDKs para fazer upload de objetos e gerenciá-los por meio de seus pontos de acesso.

Os exemplos a seguir mostram como copiar um objeto em um bucket do S3 on Outposts usando o AWS SDK for Java.

Usar o AWS SDK para Java

O exemplo do S3 on Outposts a seguir copia um objeto para um novo objeto no mesmo bucket usando o SDK para Java. Para usar esse exemplo, substitua os **user input placeholders** por suas próprias informações.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;

public class CopyObject {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String sourceKey = "*** Source object key ***";
        String destinationKey = "*** Destination object key ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            // Copy the object into a new object in the same bucket.
            CopyObjectRequest copyObjectRequest = new CopyObjectRequest(accessPointArn,
                sourceKey, accessPointArn, destinationKey);
            s3Client.copyObject(copyObjectRequest);
```

```
        } 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();
        }
    }
}
```

Obter um objeto de um bucket do Amazon S3 on Outposts

Objetos são as entidades fundamentais armazenadas no Amazon S3 on Outposts. Cada objeto está contido em um bucket. É necessário usar pontos de acesso para acessar qualquer objeto em um bucket do Outpost. Ao especificar o bucket para operações de objeto, você usa o nome do recurso da Amazon (ARN) do ponto de acesso, que inclui o código Região da AWS da região em que o Outpost está hospedado, o ID da Conta da AWS, o ID do Outpost e o nome do ponto de acesso. O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 on Outposts em operações de objeto:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obter mais informações sobre o S3 on Outposts, consulte [ARNs de recurso para S3 no Outposts \(p. 1328\)](#).

Com o Amazon S3 on Outposts, os dados do objeto são sempre armazenados no Outpost. Quando a AWS instala um rack do Outpost, seus dados permanecem no local do Outpost para atender aos requisitos de residência de dados. Seus objetos nunca saem do Outpost e não estão em uma Região da AWS. Como o AWS Management Console está hospedado na região, você não pode usá-lo para fazer upload de objetos no Outpost nem os gerenciar. No entanto, você pode usar a API REST, a AWS Command Line Interface (AWS CLI) e os SDKs para fazer upload de objetos e gerenciá-los por meio de seus pontos de acesso.

Os exemplos a seguir mostram como baixar (get) um objeto usando a AWS Command Line Interface (AWS CLI) e o AWS SDK for Java.

Uso do AWS CLI

O exemplo a seguir obtém um objeto chamado `sample-object.xml` de um bucket do S3 on Outposts (`s3-outposts:GetObject`) usando a AWS CLI. Para usar esse comando, substitua cada `user input placeholder` por suas próprias informações. Para obter mais informações sobre esse comando, consulte `get-object` na Referência de comandos da AWS CLI.

```
aws s3api get-object --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point --key testkey sample-object.xml
```

Usar o AWS SDK para Java

O exemplo do S3 on Outposts a seguir obtém um objeto usando o SDK para Java. Para usar esse exemplo, substitua cada `user input placeholder` por suas próprias informações. Para obter mais informações, consulte `GetObject` na Referência da API do Amazon Simple Storage Service.

```
import com.amazonaws.AmazonServiceException;
```

```
import com.amazonaws.SdkClientException;
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;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;

public class GetObject {
    public static void main(String[] args) throws IOException {
        String accessPointArn = "*** access point ARN ***";
        String key = "*** Object key ***";

        S3Object fullObject = null, objectPortion = null, headerOverrideObject = null;
        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            // Get an object and print its contents.
            System.out.println("Downloading an object");
            fullObject = s3Client.getObject(new GetObjectRequest(accessPointArn, 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(accessPointArn, 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(accessPointArn, 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();
}
}
```

Listar os objetos em um bucket do Amazon S3 on Outposts

Objetos são as entidades fundamentais armazenadas no Amazon S3 on Outposts. Cada objeto está contido em um bucket. É necessário usar pontos de acesso para acessar qualquer objeto em um bucket do Outpost. Ao especificar o bucket para operações de objeto, você usa o nome do recurso da Amazon (ARN) do ponto de acesso, que inclui o código Região da AWS da região em que o Outpost está hospedado, o ID da Conta da AWS, o ID do Outpost e o nome do ponto de acesso. O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 on Outposts em operações de objeto:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obter mais informações sobre o S3 on Outposts, consulte [ARNs de recurso para S3 no Outposts \(p. 1328\)](#).

Note

Com o Amazon S3 on Outposts, os dados do objeto são sempre armazenados no Outpost. Quando a AWS instala um rack do Outpost, seus dados permanecem no local do Outpost para atender aos requisitos de residência de dados. Seus objetos nunca saem do Outpost e não estão em uma Região da AWS. Como o AWS Management Console está hospedado na região, você não pode usá-lo para fazer upload de objetos no Outpost nem os gerenciar. No entanto, você pode usar a API REST, a AWS Command Line Interface (AWS CLI) e os SDKs para fazer upload de objetos e gerenciá-los por meio de seus pontos de acesso.

Os exemplos a seguir mostram como listar os objetos em um bucket do S3 on Outposts usando a AWS CLI e o AWS SDK for Java.

Uso do AWS CLI

O exemplo a seguir lista os objetos em um bucket do S3 on Outposts (`s3-outposts>ListObjectsV2`) usando a AWS CLI. Para usar esse comando, substitua cada `user input placeholder` por suas próprias informações. Para obter mais informações sobre esse comando, consulte [list-objects-v2](#) na Referência da AWS CLI.

```
aws s3api list-objects-v2 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

Note

Ao usar essa ação com o Amazon S3 on Outposts por meio de AWS SDKs, forneça o ARN do ponto de acesso do Outposts no lugar do nome do bucket, no seguinte formato: `arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/` accesspoint/`example-Outposts-Access-Point`. Para obter mais informações sobre o S3 on Outposts, consulte [ARNs de recurso para S3 no Outposts \(p. 1328\)](#).

Usar o AWS SDK para Java

O exemplo do S3 on Outposts a seguir lista objetos em um bucket usando o SDK para Java. Para usar esse exemplo, substitua cada `user input placeholder` por suas próprias informações.

Important

Este exemplo usa [ListObjectsV2](#), que é a última revisão da operação ListObjects da API. Recomendamos que você use essa operação de API revisada para o desenvolvimento de aplicações. Para compatibilidade com versões anteriores, o Amazon S3 continua a oferecer suporte à versão anterior desta operação de API.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
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 ListObjectsV2 {

    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            System.out.println("Listing objects");

            // maxKeys is set to 2 to demonstrate the use of
            // ListObjectsV2Result.getNextContinuationToken()
            ListObjectsV2Request req = new
ListObjectsV2Request().withBucketName(accessPointArn).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();
}
}
```

Excluir objetos em buckets do Amazon S3 on Outposts

Objetos são as entidades fundamentais armazenadas no Amazon S3 on Outposts. Cada objeto está contido em um bucket. É necessário usar pontos de acesso para acessar qualquer objeto em um bucket do Outpost. Ao especificar o bucket para operações de objeto, você usa o nome do recurso da Amazon (ARN) do ponto de acesso, que inclui o código Região da AWS da região em que o Outpost está hospedado, o ID da Conta da AWS, o ID do Outpost e o nome do ponto de acesso. O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 on Outposts em operações de objeto:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obter mais informações sobre o S3 on Outposts, consulte [ARNs de recurso para S3 no Outposts \(p. 1328\)](#).

Com o Amazon S3 on Outposts, os dados do objeto são sempre armazenados no Outpost. Quando a AWS instala um rack do Outpost, seus dados permanecem no local do Outpost para atender aos requisitos de residência de dados. Seus objetos nunca saem do Outpost e não estão em uma Região da AWS. Como o AWS Management Console está hospedado na região, você não pode usá-lo para fazer upload de objetos no Outpost nem os gerenciar. No entanto, você pode usar a API REST, a AWS Command Line Interface (AWS CLI) e os SDKs para fazer upload de objetos e gerenciá-los por meio de seus pontos de acesso.

Os exemplos a seguir mostram como excluir um ou vários objetos em um bucket do S3 on Outposts usando a AWS Command Line Interface (AWS CLI) e o AWS SDK for Java.

Uso do AWS CLI

Os exemplos a seguir mostram como excluir um único ou vários objetos de um bucket do S3 on Outposts.

delete-object

O exemplo a seguir exclui um objeto chamado `sample-object.xml` de um bucket do S3 on Outposts (`s3-outposts:DeleteObject`) usando a AWS CLI. Para usar esse comando, substitua cada `user input placeholder` por suas próprias informações. Para obter mais informações sobre esse comando, consulte `delete-object` na Referência de comandos da AWS CLI.

```
aws s3api delete-object --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point --key sample-object.xml
```

delete-objects

O exemplo a seguir exclui dois objetos chamados `sample-object.xml` e `test1.text` de um bucket do S3 on Outposts (`s3-outposts:DeleteObject`) usando a AWS CLI. Para usar esse

comando, substitua cada *user input placeholder* por suas próprias informações. Para obter mais informações sobre esse comando, consulte [delete-objects](#) na Referência de comandos da AWS CLI.

```
aws s3api delete-objects --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point --delete file://delete.json

delete.json
{
    "Objects": [
        {
            "Key": "test1.txt"
        },
        {
            "Key": "sample-object.xml"
        }
    ],
    "Quiet": false
}
```

Usar o AWS SDK para Java

Os exemplos a seguir mostram como excluir um único ou vários objetos de um bucket do S3 on Outposts.

DeleteObject

O exemplo do S3 on Outposts a seguir exclui um objeto de um bucket usando o SDK para Java. Para usar este exemplo, especifique o ARN do ponto de acesso para o Outpost e o nome da chave para o objeto que você deseja excluir. Para obter mais informações, consulte [DeleteObject](#) na Referência da API do Amazon Simple Storage Service.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.DeleteObjectRequest;

public class DeleteObject {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String keyName = "*** key name ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            s3Client.deleteObject(new DeleteObjectRequest(accessPointArn, 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();
        }
    }
}
```

```
}
```

DeleteObjects

O exemplo do S3 on Outposts a seguir faz upload e depois exclui objetos de um bucket usando o SDK para Java. Para usar este exemplo, especifique o ARN do ponto de acesso para o Outpost. Para obter mais informações, consulte [DeleteObjects](#) na Referência da API do Amazon Simple Storage Service.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
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;

import java.util.ArrayList;

public class DeleteObjects {

    public static void main(String[] args) {
        String accessPointArn = "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .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(accessPointArn, 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(accessPointArn)
                .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();
        }
    }
}
```

```
}
```

Usar o HeadBucket para determinar se existe um bucket do S3 on Outposts e se você tem permissões de acesso

Objetos são as entidades fundamentais armazenadas no Amazon S3 on Outposts. Cada objeto está contido em um bucket. É necessário usar pontos de acesso para acessar qualquer objeto em um bucket do Outpost. Ao especificar o bucket para operações de objeto, você usa o nome do recurso da Amazon (ARN) do ponto de acesso, que inclui o código Região da AWS da região em que o Outpost está hospedado, o ID da Conta da AWS, o ID do Outpost e o nome do ponto de acesso. O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 on Outposts em operações de objeto:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obter mais informações sobre o S3 on Outposts, consulte [ARNs de recurso para S3 no Outposts \(p. 1328\)](#).

Note

Com o Amazon S3 on Outposts, os dados do objeto são sempre armazenados no Outpost. Quando a AWS instala um rack do Outpost, seus dados permanecem no local do Outpost para atender aos requisitos de residência de dados. Seus objetos nunca saem do Outpost e não estão em uma Região da AWS. Como o AWS Management Console está hospedado na região, você não pode usá-lo para fazer upload de objetos no Outpost nem os gerenciar. No entanto, você pode usar a API REST, a AWS Command Line Interface (AWS CLI) e os SDKs para fazer upload de objetos e gerenciá-los por meio de seus pontos de acesso.

Os exemplos de AWS Command Line Interface (AWS CLI) e AWS SDK for Java a seguir mostram como usar a operação HeadBucket da API para determinar se existe um bucket do Amazon S3 on Outposts e se você tem permissão para acessá-lo. Para obter mais informações, consulte [HeadBucket](#) na Referência de API do Amazon Simple Storage Service.

Uso do AWS CLI

O exemplo de AWS CLI do S3 no Outposts a seguir usa o comando `head-bucket` para determinar se existe um bucket e se você tem permissão para acessá-lo. Para usar esse comando, substitua cada `user input placeholder` por suas próprias informações. Para obter mais informações sobre esse comando, consulte [head-bucket](#) na Referência da AWS CLI.

```
aws s3api head-bucket --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

Usar o AWS SDK para Java

O exemplo do S3 on Outposts a seguir mostra como determinar se existe um bucket e se você tem permissão para acessá-lo. Para usar este exemplo, especifique o ARN do ponto de acesso para o Outpost. Para obter mais informações, consulte [HeadBucket](#) na Referência de API do Amazon Simple Storage Service.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
```

```
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.HeadBucketRequest;

public class HeadBucket {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
            credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            s3Client.headBucket(new HeadBucketRequest(accessPointArn));
        } 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();
        }
    }
}
```

Realizar e gerenciar um carregamento fracionado com o SDK para Java

Com o Amazon S3 on Outposts, é possível criar buckets do S3 em seus recursos do AWS Outposts, além de armazenar e recuperar objetos no ambiente on-premises para aplicações que exigem acesso a dados locais, processamento de dados local e residência de dados. Só é possível usar o S3 on Outposts por meio do AWS Management Console, da AWS Command Line Interface (AWS CLI), de AWS SDKs ou da API REST. Para obter mais informações, consulte [O que é o Amazon S3 on Outposts? \(p. 1318\)](#)

Os exemplos a seguir mostram como usar o S3 on Outposts com o AWS SDK for Java para executar e gerenciar um carregamento fracionado.

Tópicos

- [Efetuar o upload fracionado de um objeto em um bucket do Amazon S3 on Outposts \(p. 1376\)](#)
- [Copiar um objeto grande em um bucket do S3 on Outposts usando carregamento fracionado \(p. 1378\)](#)
- [Listar partes de um objeto em um bucket do S3 no Outposts \(p. 1379\)](#)
- [Recuperar uma lista de multipart uploads em andamento em um bucket do S3 no Outposts \(p. 1380\)](#)

Efetuar o upload fracionado de um objeto em um bucket do Amazon S3 on Outposts

O exemplo do S3 on Outposts a seguir inicia, faz upload e conclui um carregamento fracionado em um bucket usando o SDK para Java. Para usar esse exemplo, substitua cada *user input placeholder* por suas próprias informações. Para obter mais informações, consulte [Fazer upload de um objeto usando multipart upload \(p. 187\)](#).

```
import com.amazonaws.AmazonServiceException;
```

```
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.ArrayList;
import java.util.List;

public class MultipartUploadCopy {
    public static void main(String[] args) {
        String accessPointArn = "*** Source access point ARN ***";
        String sourceObjectKey = "*** Source object key ***";
        String destObjectKey = "*** Target object key ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
InitiateMultipartUploadRequest(accessPointArn, destObjectKey);
            InitiateMultipartUploadResult initResult =
s3Client.initiateMultipartUpload(initRequest);

            // Get the object size to track the end of the copy operation.
            GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest(accessPointArn, 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(accessPointArn)
                    .withSourceKey(sourceObjectKey)
                    .withDestinationBucketName(accessPointArn)
                    .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(
                accessPointArn,
                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 grande em um bucket do S3 on Outposts usando carregamento fracionado

O exemplo do S3 on Outposts a seguir usa o SDK para Java para copiar um objeto em um bucket. Para usar esse exemplo, substitua cada *user input placeholder* por suas próprias informações. Este exemplo foi adaptado de [Copiar um objeto usando multipart upload \(p. 210\)](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.ArrayList;
import java.util.List;

public class MultipartUploadCopy {
    public static void main(String[] args) {
        String accessPointArn = "*** Source access point ARN ***";
        String sourceObjectKey = "*** Source object key ***";
        String destObjectKey = "*** Target object key ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
            InitiateMultipartUploadRequest(accessPointArn, destObjectKey);
            InitiateMultipartUploadResult initResult =
            s3Client.initiateMultipartUpload(initRequest);

            // Get the object size to track the end of the copy operation.
            GetObjectMetadataRequest metadataRequest = new
            GetObjectMetadataRequest(accessPointArn, 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(accessPointArn)
        .withSourceKey(sourceObjectKey)
        .withDestinationBucketName(accessPointArn)
        .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(
    accessPointArn,
    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;
}
```

Listar partes de um objeto em um bucket do S3 no Outposts

O exemplo do S3 on Outposts a seguir lista as partes de um objeto em um bucket usando o SDK para Java. Para usar esse exemplo, substitua cada **user input placeholder** por suas próprias informações.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
```

```
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.List;

public class ListParts {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String keyName = "*** Key name ***";
        String uploadId = "*** Upload ID ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
            credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            ListPartsRequest listPartsRequest = new ListPartsRequest(accessPointArn,
keyName, uploadId);
            PartListing partListing = s3Client.listParts(listPartsRequest);
            List<PartSummary> partSummaries = partListing.getParts();

            System.out.println(partSummaries.size() + " multipart upload parts");
            for (PartSummary p : partSummaries) {
                System.out.println("Upload part: Part number = \'" + p.getPartNumber() +
"\", ETag = " + p.getETag());
            }

        } 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();
        }
    }
}
```

Recuperar uma lista de multipart uploads em andamento em um bucket do S3 no Outposts

O exemplo do S3 on Outposts a seguir mostra como recuperar uma lista de carregamentos fracionados em andamento de um bucket do Outposts usando o SDK para Java. Para usar esse exemplo, substitua cada **user input placeholder** por suas próprias informações. Este exemplo foi adaptado do exemplo [Listar multipart uploads \(p. 202\)](#) do Amazon S3.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
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;

import java.util.List;

public class ListMultipartUploads {
    public static void main(String[] args) {
```

```
String accessPointArn = "*** access point ARN ***";

try {
    // This code expects that you have AWS credentials set up per:
    // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
    AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
        .enableUseArnRegion()
        .build();

    // Retrieve a list of all in-progress multipart uploads.
    ListMultipartUploadsRequest allMultipartUploadsRequest = new
ListMultipartUploadsRequest(accessPointArn);
    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();
}
}
```

Uso de URLs pré-assinados para o S3 no Outposts

Você pode usar um URL pré-assinado para conceder acesso por tempo limitado a objetos armazenados localmente em um Outpost sem atualizar sua política de bucket. Com URLs pré-assinados, como proprietário do bucket, você pode compartilhar objetos com indivíduos em sua nuvem privada virtual (VPC) ou conceder a eles a capacidade de carregar ou excluir objetos.

Ao criar um URL pré-assinado usando os AWS SDKs ou a AWS Command Line Interface (AWS CLI), você associa o URL a uma ação específica. Você também concede acesso por tempo limitado ao URL pré-assinado escolhendo um tempo de expiração personalizado a partir de 1 segundo e de até 7 dias. Ao compartilhar o URL pré-assinado, o indivíduo na VPC pode executar a ação incorporada no URL como se fosse o usuário da assinatura original. Ao atingir o tempo de expiração, o URL expira e não funciona mais.

Limitar recursos de pre-signed URLs

Os recursos de um URL pré-assinado são limitados pelas permissões do usuário que o criou. Em essência, os URLs pré-assinados são tokens ao portador que concedem acesso ao portador. Dessa forma, recomendamos que você os proteja adequadamente.

AWS Signature Version 4 (SigV4)

Para aplicar um comportamento específico quando solicitações de URL pré-assinado forem autenticadas usando AWS Signature Version 4 (SigV4), você pode usar chaves de condição nas políticas de bucket e políticas de ponto de acesso. Por exemplo, você pode criar uma política de bucket que use a condição `s3-outposts:signatureAge` para negar qualquer solicitação de URL pré-assinado do Amazon S3 no Outposts em objetos no bucket `example-outpost-bucket` se a assinatura tiver mais de 10

minutos de idade. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Deny a presigned URL request if the signature is more than 10 minutes old",
            "Effect": "Deny",
            "Principal": {"AWS":"44445556666"},
            "Action": "s3-outposts:*",
            "Resource": "arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/*",
            "Condition": {
                "NumericGreaterThan": {"s3-outposts:signatureAge": 600000},
                "StringEquals": {"s3-outposts:authType": "REST-QUERY-STRING"}
            }
        }
    ]
}
```

Para obter uma lista de chaves de condição e exemplos adicionais de política que você pode usar para aplicar um comportamento específico quando solicitações de URL pré-assinado forem autenticadas usando o Signature Version 4, consulte [Chaves de política específicas de autenticação do AWS Signature Version 4 \(SigV4\) \(p. 1395\)](#).

Restrição de caminho de rede

Se quiser restringir o uso de URLs pré-assinados e qualquer acesso do S3 no Outposts a caminhos específicos de rede, você poderá criar políticas que exijam um caminho específico de rede. Para definir a restrição na entidade principal do IAM que faz a chamada, você pode usar políticas do AWS Identity and Access Management (IAM) baseadas em identidade (p. ex., políticas de usuário, grupo ou função do IAM). Para definir a restrição no recurso do S3 no Outposts, você pode usar políticas baseadas em recursos (p. ex., políticas de bucket e ponto de acesso).

Uma restrição de caminho de rede na entidade principal do IAM exige que o usuário dessas credenciais faça solicitações pela rede especificada. Uma restrição no bucket ou no ponto de acesso exige que todas as solicitações encaminhadas para esse recurso tenham origem na rede especificada. Essas restrições também são aplicadas fora do cenário de URL pré-assinado.

A condição global do IAM que você usa depende do tipo de endpoint. Se estiver usando o endpoint público para o S3 no Outposts, use `aws:SourceIp`. Se estiver usando um endpoint da VPC para o S3 no Outposts, use `aws:SourceVpc` ou `aws:SourceVpce`.

A declaração de política do IAM a seguir requer que a entidade principal só acesse a AWS do intervalo de rede especificado. Com essa declaração de política, exige-se que qualquer acesso tenha origem nesse intervalo. Isso inclui o caso de alguém que esteja usando um URL pré-assinado para o S3 no Outposts. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
{
    "Sid": "NetworkRestrictionForIAMPPrincipal",
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
        "NotIpAddressIfExists": {"aws:SourceIp": "IP-address-range" },
        "BoolIfExists": {"aws:ViaAWSService": "false"}
    }
}
```

Para ver um exemplo de política de bucket que usa a chave de condição global `aws:SourceIP` da AWS para restringir o acesso a um bucket do S3 no Outposts a um intervalo específico de rede, consulte [Configurar o IAM com o S3 on Outposts \(p. 1327\)](#).

Quem pode criar um URL pré-assinado

Qualquer um com credenciais de segurança válidas pode criar um pre-signed URL. Porém, para que um usuário na VPC tenha êxito no acesso a um objeto, o URL pré-assinado deve ter sido criado por alguém que tenha permissão para executar a operação na qual o URL pré-assinado é baseado.

Você pode usar as seguintes credenciais para criar um URL pré-assinado:

- Perfil de instância do IAM: válido por até 6 horas.
- AWS Security Token Service: válido por até 36 horas quando assinado com credenciais permanentes, como as credenciais do usuário raiz da Conta da AWS ou um usuário do IAM.
- Usuário do IAM: válido por até 7 dias quando você estiver usando o AWS Signature Version 4.

Para criar um URL pré-assinado que seja válido por até 7 dias, primeiro delegue credenciais de usuário do IAM (a chave de acesso e a chave secreta) ao SDK que estiver usando. Em seguida, gere um URL pré-assinado usando o AWS Signature Version 4.

Note

- Se tiver criado um URL pré-assinado usando um token temporário, o URL expirará quando o token expirar, mesmo que você tenha criado o URL com um tempo de expiração posterior.
- Como os URLs pré-assinados concedem acesso aos seus buckets do S3 no Outposts a quem tiver o URL, recomendamos protegê-los adequadamente. Para obter mais informações sobre como proteger URLs pré-assinados, consulte [Limitar recursos de pre-signed URLs \(p. 1381\)](#).

Quando o S3 no Outposts confere a data e a hora de validade de um URL pré-assinado?

O S3 no Outposts confere a data e a hora de expiração de um URL assinado ao receber a solicitação HTTP. Por exemplo, se um cliente começar a baixar um arquivo grande imediatamente antes do tempo de expiração, o download continuará mesmo que o tempo de expiração acabe durante o download. No entanto, se a conexão cair e o cliente tentar reiniciar o download posteriormente ao término do tempo de expiração, o download vai falhar.

Para obter mais informações sobre como usar um URL pré-assinado para compartilhar ou carregar objetos, consulte os tópicos a seguir.

Tópicos

- [Compartilhar objetos usando URLs pré-assinados \(p. 1383\)](#)
- [Gerar um URL pré-assinado para carregar um objeto em um bucket do S3 no Outposts \(p. 1387\)](#)

Compartilhar objetos usando URLs pré-assinados

Você pode usar um URL pré-assinado para conceder acesso por tempo limitado a objetos armazenados localmente em um Outpost sem atualizar sua política de bucket. Com URLs pré-assinados, como proprietário do bucket, você pode compartilhar objetos com indivíduos em sua nuvem privada virtual (VPC) ou conceder a eles a capacidade de carregar ou excluir objetos.

Ao criar um URL pré-assinado usando os AWS SDKs ou a AWS Command Line Interface (AWS CLI), você associa o URL a uma ação específica. Você também concede acesso por tempo limitado ao URL

pré-assinado escolhendo um tempo de expiração personalizado a partir de 1 segundo e de até 7 dias. Ao compartilhar o URL pré-assinado, o indivíduo na VPC pode executar a ação incorporada no URL como se fosse o usuário da assinatura original. Ao atingir o tempo de expiração, o URL expira e não funciona mais.

Ao criar um URL pré-assinado, você deve fornecer suas credenciais de segurança e especificar o seguinte:

- Um nome do recurso da Amazon (ARN) de ponto de acesso para o bucket do Amazon S3 no Outposts
- Uma chave de objeto
- Um método HTTP (GET para baixar objetos)
- Data e hora de expiração

Os URLs pré-assinados só são válidos pela duração especificada. Ou seja, é necessário iniciar a ação permitida pelo URL antes da data e da hora de expiração. É possível usar o URL pré-assinado várias vezes até a data e a hora de expiração. Se tiver criado um URL pré-assinado usando um token temporário, o URL expirará quando o token expirar, mesmo que você tenha criado o URL com um tempo de expiração posterior.

Os usuários na nuvem privada virtual (VPC) que tiverem acesso ao URL pré-assinado poderão 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. Como os URLs pré-assinados concedem acesso aos seus buckets do S3 no Outposts a quem tiver o URL, recomendamos proteger os URLs adequadamente. Para obter mais detalhes sobre como proteger pre-signed URLs, consulte [Limitar recursos de pre-signed URLs \(p. 1381\)](#).

Qualquer um com credenciais de segurança válidas pode criar um pre-signed URL. No entanto, o URL pré-assinado deve ter sido criado por alguém que tenha permissão para executar a operação na qual o URL pré-assinado é baseado. Para mais informações, consulte [Quem pode criar um URL pré-assinado \(p. 1383\)](#).

É possível gerar um URL pré-assinado para compartilhar um objeto em um bucket do S3 no Outposts usando os AWS SDKs e a AWS CLI. Para obter mais informações, veja os exemplos a seguir:

Uso de AWS SDKs

Você pode usar os AWS SDKs para gerar um URL pré-assinado que você pode compartilhar com terceiros para que eles recuperem um objeto.

Note

Ao usar os AWS SDKs para gerar um URL pré-assinado, o tempo máximo de expiração de um URL pré-assinado é de sete dias a partir do momento da criação.

Java

Example

O exemplo a seguir gera um URL pré-assinado que você pode compartilhar com terceiros para que eles recuperem um objeto de um bucket do S3 no Outposts. Para mais informações, consulte [Uso de URLs pré-assinados para o S3 no Outposts \(p. 1381\)](#). Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3 \(p. 1267\)](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.HttpMethod;
import com.amazonaws.SdkClientException;
```

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

import java.io.IOException;
import java.net.URL;
import java.time.Instant;

public class GeneratePresignedURL {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String accessPointArn = "*** access point ARN ***";
        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 = Instant.now().toEpochMilli();
            expTimeMillis += 1000 * 60 * 60;
            expiration.setTime(expTimeMillis);

            // Generate the presigned URL.
            System.out.println("Generating pre-signed URL.");
            GeneratePresignedUrlRequest generatePresignedUrlRequest =
                new GeneratePresignedUrlRequest(accessPointArn, 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();
        }
    }
}
```

.NET

Example

O exemplo a seguir gera um URL pré-assinado que você pode compartilhar com terceiros para que eles recuperem um objeto de um bucket do S3 no Outposts. Para mais informações, consulte [Uso de URLs pré-assinados para o S3 no Outposts \(p. 1381\)](#). Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET \(p. 1268\)](#).

```
using Amazon;
```

```
using Amazon.S3;
using Amazon.S3.Model;
using System;

namespace Amazon.DocSamples.S3
{
    class GenPresignedURLTest
    {
        private const string accessPointArn = "*** access point ARN ***";
        private const string objectKey = "*** object key ***";
        // Specify how long the presigned URL lasts, in hours.
        private const double timeoutDuration = 12;
        // 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(timeoutDuration);
        }
        static string GeneratePreSignedURL(double duration)
        {
            string urlString = "";
            try
            {
                GetPreSignedUrlRequest request1 = new GetPreSignedUrlRequest
                {
                    BucketName = accessPointArn,
                    Key = objectKey,
                    Expires = DateTime.UtcNow.AddHours(duration)
                };
                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;
        }
    }
}
```

Python

O exemplo a seguir gera um URL pré-assinado para compartilhar um objeto usando o SDK para Python (Boto3). Por exemplo, use um cliente Boto3 e a função `generate_presigned_url` para gerar um URL pré-assinado que permite a fazer realizar o GET em um objeto.

```
import boto3
url = boto3.client('s3').generate_presigned_url(
    ClientMethod='get_object',
    Params={'Bucket': 'ACCESS_POINT_ARN', 'Key': 'OBJECT_KEY'},
    ExpiresIn=3600)
```

Para obter mais informações sobre como usar o SDK para Python (Boto3) para gerar um URL pré-assinado, consulte [Python](#) na Referência de API do AWS SDK for Python (Boto).

Uso do AWS CLI

O seguinte exemplo de comando da AWS CLI gera um URL pré-assinado para um bucket do S3 no Outposts. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

Note

Ao usar a AWS CLI para gerar um URL pré-assinado, o tempo máximo de expiração de um URL pré-assinado é de sete dias a partir do momento da criação.

```
aws s3 presign s3://arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/example-outpost-access-point/mydoc.txt --expires-in 604800
```

Para obter mais informações, consulte [presign](#) (pré-assinatura) na Referência de comandos da AWS CLI.

Gerar um URL pré-assinado para carregar um objeto em um bucket do S3 no Outposts

Você pode usar um URL pré-assinado para conceder acesso por tempo limitado a objetos armazenados localmente em um Outpost sem atualizar sua política de bucket. Com URLs pré-assinados, como proprietário do bucket, você pode compartilhar objetos com indivíduos em sua nuvem privada virtual (VPC) ou conceder a eles a capacidade de carregar ou excluir objetos.

Ao criar um URL pré-assinado usando os AWS SDKs ou a AWS Command Line Interface (AWS CLI), você associa o URL a uma ação específica. Você também concede acesso por tempo limitado ao URL pré-assinado escolhendo um tempo de expiração personalizado a partir de 1 segundo e de até 7 dias. Ao compartilhar o URL pré-assinado, o indivíduo na VPC pode executar a ação incorporada no URL como se fosse o usuário da assinatura original. Ao atingir o tempo de expiração, o URL expira e não funciona mais.

Ao criar um URL pré-assinado, você deve fornecer suas credenciais de segurança e especificar o seguinte:

- Um nome do recurso da Amazon (ARN) de ponto de acesso para o bucket do Amazon S3 no Outposts
- Uma chave de objeto
- Um método HTTP (PUT para carregar objetos)
- Data e hora de expiração

Os URLs pré-assinados só são válidos pela duração especificada. Ou seja, é necessário iniciar a ação permitida pelo URL antes da data e da hora de expiração. É possível usar o URL pré-assinado várias vezes até a data e a hora de expiração. Se tiver criado um URL pré-assinado usando um token temporário, o URL expirá quando o token expirar, mesmo que você tenha criado o URL com um tempo de expiração posterior.

Se a ação permitida por um URL pré-assinado consistir em várias etapas, como um multipart upload, é necessário iniciar todas as etapas antes do tempo de expiração. Você receberá um erro se o S3 no Outposts tentar iniciar uma etapa com um URL expirado.

Os usuários na nuvem privada virtual (VPC) que tiverem acesso ao URL pré-assinado poderão carregar objetos. Por exemplo, um usuário na VPC que tenha acesso ao URL pré-assinado pode carregar um objeto para o seu bucket. Como os URLs pré-assinados concedem acesso aos seus buckets do S3 no Outposts a qualquer usuário na VPC que tenha acesso ao URL pré-assinado, recomendamos proteger esses URLs adequadamente. Para obter mais detalhes sobre como proteger pre-signed URLs, consulte [Limitar recursos de pre-signed URLs \(p. 1381\)](#).

Qualquer um com credenciais de segurança válidas pode criar um pre-signed URL. No entanto, o URL pré-assinado deve ter sido criado por alguém que tenha permissão para executar a operação na

qual o URL pré-assinado é baseado. Para mais informações, consulte [Quem pode criar um URL pré-assinado \(p. 1383\)](#).

Usar os AWS SDKs para gerar um URL pré-assinado para uma operação de objeto do S3 no Outposts

Java

SDK para Java 2.x

Este exemplo mostra como gerar um URL pré-assinado que você pode usar para carregar um objeto em um bucket do S3 no Outposts por um tempo limitado. Para mais informações, consulte [Uso de URLs pré-assinados para o S3 no Outposts \(p. 1381\)](#).

```
public static void signBucket(S3Presigner presigner, String
outpostAccessPointArn, String keyName) {

    try {
        PutObjectRequest objectRequest = PutObjectRequest.builder()
            .bucket(accessPointArn)
            .key(keyName)
            .contentType("text/plain")
            .build();

        PutObjectPresignRequest presignRequest =
PutObjectPresignRequest.builder()
            .signatureDuration(Duration.ofMinutes(10))
            .putObjectRequest(objectRequest)
            .build();

        PresignedPutObjectRequest presignedRequest =
presigner.presignPutObject(presignRequest);

        String myURL = presignedRequest.url().toString();
        System.out.println("Presigned URL to upload a file to: " +myURL);
        System.out.println("Which HTTP method must be used when uploading a
file: " +
            presignedRequest.httpRequest().method());

        // Upload content to the S3 on Outposts bucket by using this URL.
        URL url = presignedRequest.url();

        // Create the connection and use it to upload the new object by using
the presigned URL.
        HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
        connection.setDoOutput(true);
        connection.setRequestProperty("Content-Type", "text/plain");
        connection.setRequestMethod("PUT");
        OutputStreamWriter out = new
OutputStreamWriter(connection.getOutputStream());
        out.write("This text was uploaded as an object by using a presigned
URL.");
        out.close();

        connection.getResponseCode();
        System.out.println("HTTP response code is " +
connection.getResponseCode());

    } catch (S3Exception e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

```
    }  
}
```

Python

SDK para Python (Boto3).

Este exemplo mostra como gerar um URL pré-assinado que pode executar uma ação do S3 no Outposts por um tempo limitado. Para mais informações, consulte [Uso de URLs pré-assinados para o S3 no Outposts \(p. 1381\)](#). Para fazer uma solicitação com o URL, use o pacote Requests.

```
import argparse
import logging
import boto3
from botocore.exceptions import ClientError
import requests

logger = logging.getLogger(__name__)

def generate_presigned_url(s3_client, client_method, method_parameters,
                           expires_in):
    """
    Generate a presigned S3 on Outposts URL that can be used to perform an action.

    :param s3_client: A Boto3 Amazon S3 client.
    :param client_method: The name of the client method that the URL performs.
    :param method_parameters: The parameters of the specified client method.
    :param expires_in: The number of seconds that the presigned URL is valid for.
    :return: The presigned URL.
    """
    try:
        url = s3_client.generate_presigned_url(
            ClientMethod=client_method,
            Params=method_parameters,
            ExpiresIn=expires_in
        )
        logger.info("Got presigned URL: %s", url)
    except ClientError:
        logger.exception(
            "Couldn't get a presigned URL for client method '%s'.", client_method)
        raise
    return url

def usage_demo():
    logging.basicConfig(level=logging.INFO, format='%(levelname)s: %(message)s')

    print('*'*88)
    print("Welcome to the Amazon S3 on Outposts presigned URL demo.")
    print('*'*88)

    parser = argparse.ArgumentParser()
    parser.add_argument('accessPointArn', help="The name of the S3 on Outposts access point ARN.")
    parser.add_argument(
        'key', help="For a GET operation, the key of the object in S3 on Outposts. For a "
                    "PUT operation, the name of a file to upload.")
    parser.add_argument(
        'action', choices=('get', 'put'), help="The action to perform.")
    args = parser.parse_args()
```

```
s3_client = boto3.client('s3')
client_action = 'get_object' if args.action == 'get' else 'put_object'
url = generate_presigned_url(
    s3_client, client_action, {'Bucket': args.accessPointArn, 'Key': args.key},
    1000)

print("Using the Requests package to send a request to the URL.")
response = None
if args.action == 'get':
    response = requests.get(url)
elif args.action == 'put':
    print("Putting data to the URL.")
    try:
        with open(args.key, 'r') as object_file:
            object_text = object_file.read()
        response = requests.put(url, data=object_text)
    except FileNotFoundError:
        print(f"Couldn't find {args.key}. For a PUT operation, the key must be"
the "
f"name of a file that exists on your computer.")

if response is not None:
    print("Got response:")
    print(f"Status: {response.status_code}")
    print(response.text)

print('*'*88)

if __name__ == '__main__':
    usage_demo()
```

Segurança no S3 on Outposts

A segurança da nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você se beneficiará de datacenters e arquiteturas de rede criados para atender aos requisitos das empresas com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem: a AWS é responsável pela proteção da infraestrutura que executa os Serviços da AWS na Nuvem AWS. A AWS também fornece serviços que você pode usar com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos [Programas de conformidade da AWS](#). Para saber mais sobre os programas de conformidade que se aplicam ao Amazon S3 on Outposts, consulte [Serviços da AWS em escopo por programa de conformidade](#).
- Segurança na nuvem: sua responsabilidade é determinada pelo AWS service (Serviço da AWS) que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da sua empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o S3 on Outposts. Os tópicos a seguir mostram como configurar o S3 on Outposts para atender aos seus objetivos de segurança e conformidade. Saiba também como usar outros Serviços da AWS que ajudam a monitorar e proteger os recursos do S3 on Outposts.

Tópicos

- [Criptografia de dados no S3 on Outposts \(p. 1391\)](#)

- [AWS PrivateLink para S3 on Outposts \(p. 1391\)](#)
- [Chaves de política específicas de autenticação do AWS Signature Version 4 \(SigV4\) \(p. 1395\)](#)

Criptografia de dados no S3 on Outposts

Por padrão, todos os dados armazenados no Amazon S3 on Outposts são criptografados usando criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 (SSE-S3). Para obter mais informações, consulte [Proteção de dados usando criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 \(SSE-S3\) \(p. 378\)](#).

Você também pode optar por usar a criptografia no lado do servidor com chaves de criptografia fornecidas pelo cliente (SSE-C). Para usar o SSE-C, especifique uma chave de criptografia como parte das solicitações da API de objeto. A criptografia no lado do servidor criptografa somente os dados de objeto, não os metadados de objeto. Para obter mais informações, consulte [Proteger dados usando a criptografia de servidor com chaves de criptografia fornecidas pelo cliente \(SSE-C\) \(p. 390\)](#).

Note

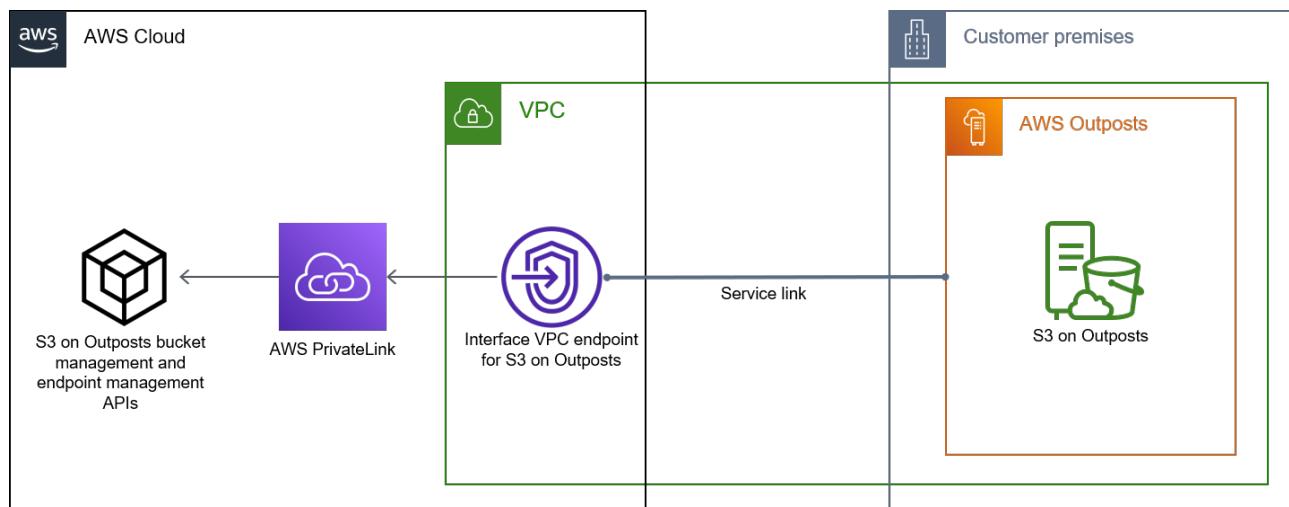
O S3 on Outposts não é compatível com a criptografia no lado do servidor usando chaves do AWS Key Management Service (AWS KMS) (SSE-KMS).

AWS PrivateLink para S3 on Outposts

Com AWS PrivateLink para o Amazon S3 on Outposts, você pode provisionar endpoints da VPC de interface em sua nuvem privada virtual (VPC) para gerenciar sua implantação do S3 on Outposts por meio das APIs de [gerenciamento de buckets \(p. 1405\)](#) e [gerenciamento de endpoints \(p. 1406\)](#) do S3 on Outposts. Os endpoints da VPC de interface são diretamente acessíveis por aplicações implantadas em sua VPC ou no ambiente on-premises por meio de sua rede privada virtual (VPN) ou do AWS Direct Connect. Você pode acessar as APIs de gerenciamento de buckets e endpoints por meio de AWS PrivateLink. O AWS PrivateLink não é compatível com operações de [transferência de dados \(p. 1405\)](#) da API, como GET, PUT e APIs semelhantes. Essas operações são transferidas de forma privada por meio da configuração do ponto de acesso e do endpoint do S3 no Outposts. Para obter mais informações, consulte [Redes para S3 on Outposts \(p. 1338\)](#).

Os endpoints de interface são representados por uma ou mais interfaces de rede elástica (ENIs) que recebem endereços IP privados de sub-redes em sua VPC. As solicitações realizadas para endpoints de interface para o S3 on Outposts são encaminhadas automaticamente para as APIs de gerenciamento de buckets e endpoints do S3 on Outposts na rede da Amazon. Você também pode acessar endpoints de interface em sua VPC via aplicações on-premises por meio do AWS Direct Connect ou da AWS Virtual Private Network (AWS VPN). Para obter mais informações sobre como conectar sua VPC à rede on-premises, consulte o [Guia do usuário do AWS Direct Connect](#) e o [Guia do usuário do AWS Site-to-Site VPN](#).

Os endpoints de interface encaminham as solicitações para APIs de gerenciamento de buckets e endpoints do S3 on Outposts pela rede da Amazon e por meio de AWS PrivateLink, conforme ilustrado no diagrama a seguir.



Para obter informações gerais sobre endpoints de interface, consulte [Endpoints da VPC de interface \(AWS PrivateLink\)](#) no Manual do AWS PrivateLink.

Tópicos

- [Restrições e limitações \(p. 1392\)](#)
- [Acessar endpoints da interface do S3 on Outposts \(p. 1392\)](#)
- [Atualizar uma configuração de DNS no local \(p. 1394\)](#)
- [Criar um endpoint da VPC para o S3 on Outposts \(p. 1394\)](#)
- [Criar políticas de bucket e políticas de endpoint da VPC para S3 on Outposts \(p. 1394\)](#)

Restrições e limitações

Quando você acessa as APIs de gerenciamento de buckets e endpoints do S3 on Outposts por meio de AWS PrivateLink, há limitações da VPC. Para obter mais informações, consulte [Propriedades e limitações de endpoints de interface e Cotas do AWS PrivateLink](#) no Guia do usuário do AWS PrivateLink.

Além disso, o AWS PrivateLink não oferece compatibilidade com:

- [Endpoints do Federal Information Processing Standard \(FIPS – Padrões Federais de Processamento de Informações\)](#)
- [APIs de transferência de dados do S3 on Outposts \(p. 1405\)](#); por exemplo, GET, PUT e operações de API de objetos semelhantes.

Acessar endpoints da interface do S3 on Outposts

Para acessar APIs de gerenciamento de buckets e endpoints do S3 on Outposts usando AWS PrivateLink, você precisa atualizar suas aplicações para usar nomes de DNS específicos de endpoint. Quando você cria um endpoint de interface, o AWS PrivateLink gera dois tipos de nome do S3 on Outposts específicos do endpoint: regional e por zona.

- Nomes de DNS regionais: incluem um ID de endpoint da VPC exclusivo, um identificador de serviço, a Região da AWS e `vpce.amazonaws.com`; por exemplo, `vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com`.
- Nomes de DNS por zona: incluem um ID de endpoint da VPC exclusivo, a zona de disponibilidade, um identificador de serviço, a Região da AWS e `vpce.amazonaws.com`; por exemplo, `vpce-1a2b3c4d-5e6f-us-east-1a.s3-outposts.us-east-1.vpce.amazonaws.com`. Você

pode usar essa opção se sua arquitetura isola zonas de disponibilidade. Por exemplo, você pode usar nomes DNS por zona para contenção de falhas ou para reduzir os custos da transferência de dados regional.

Os nomes de DNS do S3 on Outposts específicos do endpoint podem ser resolvidos no domínio DNS público do S3. Os endpoints de interface do S3 on Outposts também são compatíveis com o recurso DNS privado dos endpoints de interface. Para obter mais informações sobre [DNS privado para endpoints de interface](#), consulte o Guia do usuário do AWS PrivateLink.

Exemplos do AWS CLI

Use os parâmetros `--region` e `--endpoint-url` para acessar as APIs de gerenciamento de buckets e endpoints por meio de endpoints de interface do S3 on Outposts.

Example : usar o URL do endpoint para listar buckets com a API de controle do S3

No exemplo a seguir, substitua a região `us-east-1`, o URL do endpoint da VPC `vpce-1a2b3c4d-5e6f.s3.us-east-1_vpce.amazonaws.com` e o ID da conta `111122223333` pelas informações apropriadas.

```
aws s3control list-regional-buckets --region us-east-1 --endpoint-url
https://vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1_vpce.amazonaws.com --account-
id 111122223333
```

Exemplos de AWS SDK

Atualize seus SDKs para a versão mais recente e configure seus clientes para usar um URL de endpoint a fim de acessar a API de controle para endpoints de interface do S3 on Outposts. Para obter mais informações, consulte [Exemplos de AWS SDK para AWS PrivateLink](#).

SDK for Python (Boto3)

Example : use um URL de endpoint para acessar a API de controle do S3

No exemplo a seguir, substitua a região `us-east-1` e o URL de endpoint da VPC `vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1_vpce.amazonaws.com` por informações apropriadas.

```
control_client = session.client(
    service_name='s3control',
    region_name='us-east-1',
    endpoint_url='https://vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1_vpce.amazonaws.com'
)
```

SDK for Java 2.x

Example : use um URL de endpoint para acessar a API de controle do S3

No exemplo a seguir, substitua o URL do endpoint da VPC `vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1_vpce.amazonaws.com` e a região `Region.US_EAST_1` por informações apropriadas.

```
// control client
Region region = Region.US_EAST_1;
S3ControlClient = S3ControlClient.builder().region(region)

.endpointOverride(URI.create("https://vpce-1a2b3c4d-5e6f.s3-outposts.us-
east-1_vpce.amazonaws.com"))
.build()
```

Atualizar uma configuração de DNS no local

Ao usar nomes de DNS específicos do endpoint para acessar os endpoints de interface para as APIs de gerenciamento de buckets e endpoints do S3 on Outposts, você não precisa atualizar seu resolvedor de DNS on-premises. Você pode resolver o nome de DNS específico do endpoint com o endereço IP privado do endpoint de interface do domínio DNS público do S3 on Outposts.

Criar um endpoint da VPC para o S3 on Outposts

Para criar um endpoint da VPC de interface para S3 em Outposts, consulte [Criar um endpoint da VPC](#) no Guia do AWS PrivateLink.

Criar políticas de bucket e políticas de endpoint da VPC para S3 on Outposts

É possível anexar uma política de endpoint ao endpoint da VPC que controla o acesso ao S3 on Outposts. Você também pode usar a condição `aws:sourceVpce` em políticas de bucket do S3 on Outposts para restringir o acesso a buckets específicos de determinado endpoint da VPC. Com as políticas de endpoint da VPC, você pode controlar o acesso às APIs de gerenciamento de buckets e endpoints do S3 on Outposts. Com as políticas de bucket, você pode controlar o acesso às APIs de gerenciamento de buckets do S3 on Outposts. No entanto, você não pode gerenciar o acesso a ações de objeto para o S3 on Outposts usando `aws:sourceVpce`.

As políticas de acesso para o S3 on Outposts especificam as seguintes informações:

- A entidade principal do AWS Identity and Access Management (IAM) para a qual as ações são permitidas ou negadas.
- As ações de controle do S3 permitidas ou negadas.
- Os recursos do S3 on Outposts nos quais as ações são permitidas ou negadas.

Os exemplos a seguir mostram políticas que restringem o acesso a um bucket ou a um endpoint. Para obter mais informações sobre conectividade de VPC, consulte [Network-to-VPC connectivity options](#) (Opções de conectividade entre rede e VPC) no whitepaper da AWS [Amazon Virtual Private Cloud Connectivity Options](#) (Opções de conectividade do Amazon Virtual Private Cloud).

Important

- Ao aplicar as políticas de exemplo para endpoints da VPC descritas nesta seção, talvez você bloquee o acesso ao bucket acidentalmente. As permissões de bucket que limitam o acesso do bucket às conexões originadas de seu endpoint da VPC podem bloquear todas as conexões ao bucket. Para obter informações sobre como corrigir esse problema, consulte [Minha política de bucket tem o ID da VPC ou do endpoint da VPC incorreto. Como corrigir a política para que eu possa acessar o bucket?](#) na Central de conhecimento do AWS Support.
- Antes de usar a política de bucket de exemplo a seguir, substitua o ID do endpoint da VPC por um valor apropriado para o caso de uso. Caso contrário, não será possível acessar o bucket.
- Se a política permitir apenas o acesso a um bucket do S3 on Outposts por meio de um endpoint da VPC específico, ela desativará o acesso do console para esse bucket porque as solicitações do console não se originam do endpoint da VPC especificado.

Tópicos

- [Exemplo: restringir o acesso a um bucket específico a partir de um endpoint da VPC \(p. 417\)](#)
- [Exemplo: negar acesso usando um endpoint da VPC específico em uma política de bucket do S3 on Outposts \(p. 1395\)](#)

Exemplo: restringir o acesso a um bucket específico a partir de um endpoint da VPC

Você pode criar uma política de endpoint que restrinja o acesso apenas a buckets específicos do S3 on Outposts. A política a seguir restringe o acesso à ação GetBucketPolicy somente ao [example-outpost-bucket](#). Para usar essa política, substitua os valores de exemplo por seus próprios valores.

```
{  
    "Version": "2012-10-17",  
    "Id": "Policy1415115909151",  
    "Statement": [  
        { "Sid": "Access-to-specific-bucket-only",  
          "Principal": {"AWS": "111122223333"},  
          "Action": "s3-outposts:GetBucketPolicy",  
          "Effect": "Allow",  
          "Resource": "arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket"  
        }  
    ]  
}
```

Exemplo: negar acesso usando um endpoint da VPC específico em uma política de bucket do S3 on Outposts

A política de bucket do S3 on Outposts a seguir nega acesso a GetBucketPolicy no bucket do [example-outpost-bucket](#) por meio do endpoint da VPC [vpce-1a2b3c4d](#).

A condição `aws:sourceVpce` especifica o endpoint e não requer um nome do recurso da Amazon (ARN) para o recurso de endpoint da VPC, apenas o ID do endpoint. Para usar essa política, substitua os valores de exemplo por seus próprios valores.

```
{  
    "Version": "2012-10-17",  
    "Id": "Policy1415115909152",  
    "Statement": [  
        {  
            "Sid": "Deny-access-to-specific-VPCE",  
            "Principal": {"AWS": "111122223333"},  
            "Action": "s3-outposts:GetBucketPolicy",  
            "Effect": "Deny",  
            "Resource": "arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket",  
            "Condition": {  
                "StringEquals": {"aws:sourceVpce": "vpce-1a2b3c4d"}  
            }  
        }  
    ]  
}
```

Chaves de política específicas de autenticação do AWS Signature Version 4 (SigV4)

A tabela a seguir mostra as chaves de condição relacionadas à autenticação com o AWS Signature Version 4 (SigV4) que você pode usar com o Amazon S3 no Outposts. Em uma política de bucket, você pode adicionar essas condições para aplicar um comportamento específico quando as solicitações forem autenticadas usando o Signature Version 4. Para obter exemplos de políticas, consulte [Exemplos de](#)

[política de bucket que usam chaves de condição relacionadas ao Signature Version 4 \(p. 1397\)](#). Para obter mais informações sobre como autenticar solicitações usando o Signature Version 4, consulte [Authenticating requests \(AWS Signature Version 4\)](#) (Autenticação de solicitações [AWS Signature Version 4]) na Referência de API do Amazon Simple Storage Service.

Chaves aplicáveis para ações do **s3-outposts:*** ou qualquer uma das ações do S3 no Outposts

Chaves aplicáveis	Descrição
s3-outposts:authType	<p>O S3 no Outposts é compatível com vários métodos de autenticação. Você pode usar essa chave de condição opcional para restringir as solicitações recebidas a usar um método específico de autenticação. Por exemplo, é possível usar essa chave de condição para permitir que somente o cabeçalho HTTP <code>Authorization</code> seja usado na autenticação de solicitação.</p> <p>Valores válidos:</p> <ul style="list-style-type: none"> REST-HEADER REST-QUERY-STRING
s3-outposts:signatureAge	<p>A duração em milissegundos da validade de uma assinatura em uma solicitação autenticada.</p> <p>Essa condição funciona somente para URLs pré-assinados.</p> <p>No Signature Version 4, a chave de assinatura é válida por até 7 dias. Portanto, as assinaturas também são válidas por até 7 dias. Para obter mais informações, consulte Introduction to signing requests (Introdução à assinatura de solicitações) na Referência de API do Amazon Simple Storage Service. Você pode usar essa condição para limitar ainda mais a idade da assinatura.</p> <p>Valor de exemplo: 600000</p>
s3-outposts:x-amz-content-sha256	<p>É possível usar essa chave de condição para proibir conteúdo não assinado em seu bucket.</p> <p>Ao usar o Signature Version 4, para solicitações que usem o cabeçalho <code>Authorization</code>, você adiciona o cabeçalho <code>x-amz-content-sha256</code> no cálculo da assinatura e, em seguida, defina seu valor para a carga útil do hash.</p> <p>Você pode usar essa chave de condição em sua política de bucket para negar qualquer upload cujas cargas úteis não estejam assinadas. Por exemplo:</p> <ul style="list-style-type: none"> • Negar uploads que usem o cabeçalho <code>Authorization</code> para autenticar solicitações mas não assinem a carga útil. Para obter mais informações, consulte Transferring payload in a single chunk (Transferência de carga útil em um único bloco) na Referência de API do Amazon Simple Storage Service. • Negar uploads que usem URLs pré-assinados. Os URLs pré-assinados sempre têm um <code>UNSIGNED_PAYLOAD</code>. Para obter mais informações, consulte Authenticating requests (Autenticação de solicitações) e Authentication methods (Métodos de autenticação) na Referência de API do Amazon Simple Storage Service. <p>Valor válido: <code>UNSIGNED-PAYOUT</code></p>

Exemplos de política de bucket que usam chaves de condição relacionadas ao Signature Version 4

Para usar os exemplos a seguir, substitua `user input placeholders` por suas próprias informações.

Example : `s3-outposts:signatureAge`

A política de bucket a seguir nega qualquer solicitação de URL pré-assinado do S3 no Outposts em objetos em `example-outpost-bucket` se a assinatura tiver mais de 10 minutos de idade.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Deny a presigned URL request if the signature is more than 10 minutes old",  
            "Effect": "Deny",  
            "Principal": {"AWS":"444455556666"},  
            "Action": "s3-outposts:*",  
            "Resource": "arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/*",  
            "Condition": {  
                "NumericGreaterThan": {"s3-outposts:signatureAge": 600000},  
                "StringEquals": {"s3-outposts:authType": "REST-QUERY-STRING"}  
            }  
        }  
    ]  
}
```

Example : `s3-outposts:authType`

A política de bucket a seguir permite somente solicitações que usem o cabeçalho `Authorization` para autenticação de solicitação. Todas as solicitações de URL pré-assinado serão negadas, pois os URLs pré-assinados usam parâmetros de consulta para fornecer informações de solicitação e autenticação. Para obter mais informações, consulte [Authentication methods](#) (Métodos de autenticação) na Referência de API do Amazon Simple Storage Service.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Allow only requests that use the Authorization header for request authentication. Deny presigned URL requests.",  
            "Effect": "Deny",  
            "Principal": {"AWS":"111122223333"},  
            "Action": "s3-outposts:*",  
            "Resource": "arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/*",  
            "Condition": {  
                "StringNotEquals": {"s3-outposts:authType": "REST-HEADER"}  
            }  
        }  
    ]  
}
```

Example : `s3-outposts:x-amz-content-sha256`

A política de bucket a seguir nega qualquer upload com cargas úteis não assinadas, como uploads que estejam usando URLs pré-assinados. Para obter mais informações, consulte [Authenticating requests](#)

(Autenticação de solicitações) e [Authentication methods](#) (Métodos de autenticação) na Referência de API do Amazon Simple Storage Service.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Deny uploads with unsigned payloads.",  
            "Effect": "Deny",  
            "Principal": {"AWS": "111122223333"},  
            "Action": "s3-outposts:*",  
            "Resource": "arn:aws:s3-outposts:us-  
east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/*",  
            "Condition": {  
                "StringEquals": {  
                    "s3-outposts:x-amz-content-sha256": "UNSIGNED-PAYLOAD"  
                }  
            }  
        }  
    ]  
}
```

Gerenciar o armazenamento do S3 on Outposts

Com o Amazon S3 on Outposts, é possível criar buckets do S3 no AWS Outposts, além de armazenar e recuperar facilmente objetos no local para aplicações que exigem acesso a dados locais, processamento de dados local e residência de dados. O S3 on Outposts fornece uma nova classe de armazenamento, o S3 Outposts (OUTPOSTS), que usa as APIs do Amazon S3 e é projetado para armazenar dados de forma duradoura e redundante em vários dispositivos e servidores em seu AWS Outposts. Você se comunica com o bucket do Outposts usando um ponto de acesso e uma conexão de endpoint em uma nuvem privada virtual (VPC). É possível usar os mesmos recursos e APIs nos buckets do Outposts da mesma maneira que em buckets do Amazon S3, incluindo políticas de acesso, criptografia e marcação. Só é possível usar o S3 on Outposts por meio do AWS Management Console, da AWS Command Line Interface (AWS CLI), de AWS SDKs ou da API REST. Para obter mais informações, consulte [O que é o Amazon S3 on Outposts? \(p. 1318\)](#)

Para obter mais informações sobre como gerenciar, monitorar e compartilhar sua capacidade de armazenamento do Amazon S3 on Outposts, consulte os tópicos a seguir.

Tópicos

- [Como gerenciar a capacidade do S3 no Outposts com as métricas do Amazon CloudWatch \(p. 1398\)](#)
- [Como receber notificações de eventos do S3 no Outposts usando o Amazon CloudWatch Events \(p. 1399\)](#)
- [Gerenciar a capacidade do S3 on Outposts com logs do AWS CloudTrail \(p. 1400\)](#)
- [Compartilhar o S3 on Outposts usando o AWS RAM \(p. 1401\)](#)
- [Outros Serviços da AWS que usam o S3 on Outposts \(p. 1404\)](#)

Como gerenciar a capacidade do S3 no Outposts com as métricas do Amazon CloudWatch

Se não houver espaço suficiente para armazenar um objeto no Outpost, a API retornará uma isenção de capacidade insuficiente (ICE). Para evitar isso, você pode criar alertas do CloudWatch que avisem quando a utilização do armazenamento exceder determinado limite. Para obter mais informações, consulte [Métricas do Amazon S3 on Outposts no CloudWatch \(p. 1073\)](#).

Você pode usar esse método para liberar espaço excluindo explicitamente os dados, usando uma política de validade do ciclo de vida ou copiando os dados do seu bucket do Amazon S3 on Outposts em um bucket do S3 em uma Região da AWS usando o AWS DataSync. Para obter mais informações sobre como usar o DataSync, consulte [Conceitos básicos do AWS DataSync](#) no Guia do usuário do AWS DataSync.

Métricas do CloudWatch

O namespace S3Outposts inclui as seguintes métricas para buckets do Amazon S3 em Outposts. É possível monitorar o número total de bytes provisionados do S3 em Outposts, o total de bytes livres disponíveis para objetos e o tamanho total de todos os objetos de determinado bucket.

Note

O S3 on Outposts apenas é compatível com as métricas do Amazon S3 a seguir.

Como o S3 on Outposts tem capacidade limitada, é possível criar alertas do CloudWatch que avisam quando a utilização do armazenamento excede um limite.

Métrica	Descrição
OutpostTotalBytes	A capacidade provisionada total em bytes para um Outpost. Unidade: bytes Período: 5 minutos
OutpostFreeBytes	A contagem de bytes livres disponíveis em um Outpost para armazenar dados de clientes. Unidade: bytes Período: 5 minutos
BucketUsedBytes	O tamanho total de todos os objetos de determinado bucket. Unidade: contagens Período: 5 minutos
AccountUsedBytes	O tamanho total de todos os objetos da conta do Outposts especificada. Unidade: bytes Período: 5 minutos

Como receber notificações de eventos do S3 no Outposts usando o Amazon CloudWatch Events

É possível usar o Amazon CloudWatch Events para criar uma regra para qualquer evento da API do Amazon S3 on Outposts a fim de receber notificações por meio de todos os destinos compatíveis do CloudWatch. Por exemplo, o Amazon Simple Queue Service (Amazon SQS), o Amazon Simple Notification Service (Amazon SNS) e o AWS Lambda. Para obter mais informações, consulte a lista de [serviços da AWS que podem ser destinos para o CloudWatch Events](#) no Guia do usuário do Amazon CloudWatch Events. Para escolher um serviço de destino para trabalhar com o S3 on Outposts, consulte [Criar uma regra do CloudWatch Events que é acionada por uma chamada de API da AWS pelo AWS CloudTrail](#) no Guia do usuário do Amazon CloudWatch Events.

Note

Para operações de objeto do S3 on Outposts, os eventos de chamada da AWS API enviados pelo CloudTrail só corresponderão às suas regras, se você tiver trilhas (opcionalmente com seletores de eventos) configuradas para receber esses eventos. Para obter mais informações, consulte [Trabalhar com arquivos de log do CloudTrail](#) no Guia do usuário do AWS CloudTrail.

Example

Veja a seguir uma regra de amostra para a operação `DeleteObject`. Para usar essa regra de exemplo, substitua `DOC-EXAMPLE-BUCKET1` pelo nome do bucket do S3 on Outposts.

```
{  
    "source": [  
        "aws.s3-outposts"  
    ],  
    "detail-type": [  
        "AWS API call through CloudTrail"  
    ],  
    "detail": {  
        "eventSource": [  
            "s3-outposts.amazonaws.com"  
        ],  
        "eventName": [  
            "DeleteObject"  
        ],  
        "requestParameters": {  
            "bucketName": [  
                "DOC-EXAMPLE-BUCKET1"  
            ]  
        }  
    }  
}
```

Gerenciar a capacidade do S3 on Outposts com logs do AWS CloudTrail

Seus eventos de gerenciamento do Amazon S3 on Outposts estão disponíveis por meio de logs do AWS CloudTrail. Para obter mais informações, consulte [Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail \(p. 1023\)](#).

Além disso, é possível habilitar o registro em log para eventos de dados no CloudTrail opcionalmente. Para obter mais informações, consulte [Habilitar o registro em log de objetos em um bucket usando o console \(p. 1033\)](#).

Configurar o bucket do S3 on Outposts para registro no CloudTrail usando o console do S3

O procedimento a seguir mostra como configurar o bucket do S3 on Outposts para a emissão aos logs do AWS CloudTrail.

Note

A proprietária do bucket é a Conta da AWS que o criou e só ela pode configurar eventos de dados do Amazon S3 a serem enviados ao AWS CloudTrail.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. No painel de navegação à esquerda, escolha Outposts buckets (Buckets do Outposts).

3. Escolha o bucket do Outposts cujos eventos de dados você deseja registrar usando o CloudTrail.
4. Acesse a seção AWS CloudTrail data events (Eventos de dados do AWS CloudTrail) e escolha Configure in CloudTrail (Configurar no CloudTrail).

Para obter mais informações, consulte [Habilitar o log de eventos do CloudTrail para buckets e objetos do S3 \(p. 1033\)](#).

Compartilhar o S3 on Outposts usando o AWS RAM

O Amazon S3 on Outposts é compatível com o compartilhamento da capacidade do S3 entre várias contas de uma organização usando o AWS Resource Access Manager ([AWS RAM](#)). Com o compartilhamento do S3 on Outposts, você pode permitir que outras pessoas criem e gerenciem buckets, endpoints e pontos de acesso no Outpost.

Este tópico demonstra como usar o AWS RAM para compartilhar o S3 on Outposts e recursos relacionados com outra Conta da AWS da organização da AWS.

Pré-requisitos

- A conta proprietária do Outpost tem uma organização configurada no AWS Organizations. Para obter mais informações, consulte [Criar uma organização](#) no Guia do usuário do AWS Organizations.
- A organização inclui a Conta da AWS com a qual você deseja compartilhar a capacidade do S3 on Outposts. Para obter mais informações, consulte [Enviar convites para Contas da AWS](#) no Guia do usuário do AWS Organizations.
- Selecione uma das opções a seguir que você deseja compartilhar. O segundo recurso [Subnets (Sub-redes) ou Outposts] deve ser selecionado para que os endpoints também sejam acessíveis. Os endpoints são um requisito de rede para acessar dados armazenados no S3 on Outposts.

Opção 1	Opção 2
<p>S3 on Outposts</p> <p>Permite que o usuário crie buckets no Outposts e pontos de acesso, e adicione objetos a esses buckets.</p> <p>Sub-redes</p> <p>Permite que o usuário use a nuvem privada virtual (VPC) e os endpoints associados à sub-rede.</p>	<p>S3 on Outposts</p> <p>Permite que o usuário crie buckets no Outposts e pontos de acesso, e adicione objetos a esses buckets.</p> <p>Outposts</p> <p>Permite que o usuário veja os gráficos de capacidade do S3 e a página inicial do console do AWS Outposts. Também permite que os usuários criem sub-redes em Outposts compartilhados e criem endpoints.</p>

Procedimento

1. Faça login no AWS Management Console usando a Conta da AWS que é proprietária do Outpost, depois abra o console do AWS RAM em <https://console.aws.amazon.com/ram>.
2. Lembre-se de ativar o compartilhamento com o AWS Organizations em AWS RAM. Para obter mais informações, consulte [Habilitar compartilhamento de recursos no AWS Organizations](#) no Guia do usuário do AWS RAM.
3. Use a Opção 1 ou a Opção 2 nos [pré-requisitos \(p. 1401\)](#) para criar o compartilhamento de um recurso. Se você tiver vários recursos do S3 on Outposts, selecione os nomes de recurso da Amazon

(ARNs) dos recursos que você deseja compartilhar. Para ativar endpoints, compartilhe a sub-rede ou o Outpost.

Para obter mais informações sobre como criar um compartilhamento de recursos, consulte [Criar um compartilhamento de recursos](#) no Guia do usuário do AWS RAM.

4. A Conta da AWS com a qual você compartilhou recursos deve agora poder usar o S3 on Outposts. Dependendo da opção que você selecionou em [prerequisites \(p. 1401\)](#) (pré-requisitos), forneça as seguintes informações ao usuário da conta:

Opção 1	Opção 2
O ID do Outpost	O ID do Outpost
O ID da VPC	
O ID da sub-rede	
O ID do grupo de segurança	

Note

O usuário pode confirmar que os recursos foram compartilhados com ele usando o console do AWS RAM, a AWS Command Line Interface (AWS CLI), os AWS SDKs ou a API REST. O usuário pode visualizar os compartilhamentos de recursos existentes usando o comando [get-resource-shares](#) da CLI.

Exemplos de uso

Depois que você compartilha os recursos do S3 on Outposts com outra conta, essa conta pode gerenciar buckets e objetos no Outpost. Se você compartilhou o recurso Subnets (Sub-redes), essa conta poderá usar o endpoint que você criou. Os exemplos a seguir demonstram como um usuário pode usar o AWS CLI para interagir com o Outpost após você compartilhar esses recursos.

Example : criar um bucket

O exemplo a seguir cria um bucket chamado [DOC-EXAMPLE-BUCKET1](#) no Outpost [op-01ac5d28a6a232904](#). Antes de usar esse comando, substitua cada *user input placeholder* pelos valores apropriados para seu caso de uso.

```
aws s3control create-bucket --bucket DOC-EXAMPLE-BUCKET1 --outpost-id op-01ac5d28a6a232904
```

Para obter mais informações sobre esse comando, consulte [create-bucket](#) na Referência da AWS CLI.

Example : criar um ponto de acesso

O exemplo a seguir cria um ponto de acesso em um Outpost usando os parâmetros de exemplo na tabela a seguir. Antes de usar esse comando, substitua esses valores *user input placeholder* e o código da Região da AWS pelos valores apropriados para o caso de uso.

Parâmetro	Valor
ID da conta	111122223333
Nome do ponto de acesso	example-outpost-access-point

Parâmetro	Valor
ID do Outpost	<code>op-01ac5d28a6a232904</code>
Nome do bucket do Outpost	<code>DOC-EXAMPLE-BUCKET1</code>
ID da VPC	<code>vpc-1a2b3c4d5e6f7g8h9</code>

Note

O parâmetro Account ID (ID da conta) deve ser o ID da Conta da AWS do proprietário do bucket, que é o usuário compartilhado.

```
aws s3control create-access-point --account-id 111122223333 --name example-outpost-access-point \
--bucket arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/
bucket/DOC-EXAMPLE-BUCKET1 \
--vpc-configuration VpcId=vpc-1a2b3c4d5e6f7g8h9
```

Para obter mais informações sobre esse comando, consulte [create-access-point](#) na Referência da AWS CLI.

Example : carregar um objeto

O exemplo a seguir carrega o arquivo `my_image.jpg` do sistema de arquivos local do usuário em um objeto chamado `images/my_image.jpg` por meio do ponto de acesso `example-outpost-access-point` no Outpost `op-01ac5d28a6a232904`, pertencente à conta da AWS `111122223333`. Antes de usar esse comando, substitua esses valores `user input placeholder` e o código da Região da AWS pelos valores apropriados para o caso de uso.

```
aws s3api put-object --bucket arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/example-outpost-access-point \
--body my_image.jpg --key images/my_image.jpg
```

Para obter mais informações sobre esse comando, consulte [put-object](#) na Referência da AWS CLI.

Note

Se essa operação resultar em um erro Resource not found (Recurso não localizado) ou não responder, pode ser que sua VPC não tenha um endpoint compartilhado.

Para verificar se há um endpoint compartilhado, use o comando [list-shared-endpoints](#) da AWS CLI. Se não houver um endpoint compartilhado, atue com o proprietário do Outpost para criar um.

Para obter mais informações, consulte [ListSharedEndpoints](#) na Referência da API do Amazon Simple Storage Service.

Example : criar um endpoint

O exemplo a seguir cria um endpoint em um Outpost compartilhado. Antes de usar esse comando, substitua os valores `user input placeholder` para o ID do Outpost, o ID da sub-rede e o ID do grupo de segurança pelos valores apropriados para o caso de uso.

Note

O usuário só poderá executar essa operação se o compartilhamento de recursos incluir o recurso Outposts.

```
aws s3outposts create-endpoint --outposts-id op-01ac5d28a6a232904 --subnet-id XXXXXX --
security-group-id XXXXXX
```

Para obter mais informações sobre esse comando, consulte [create-endpoint](#) na Referência da AWS CLI.

Outros Serviços da AWS que usam o S3 on Outposts

Outros Serviços da AWS que são executados localmente para seu AWS Outposts também podem usar a capacidade do Amazon S3 on Outposts. No Amazon CloudWatch, o namespace `S3Outposts` mostra métricas detalhadas de buckets no S3 on Outposts, mas elas não incluem o uso de outros Serviços da AWS. Para gerenciar a capacidade do S3 on Outposts que é consumida por outros Serviços da AWS, consulte as informações na tabela a seguir.

AWS service (Serviço da AWS)	Descrição	Saiba mais
Amazon S3	Todo uso direto do S3 on Outposts tem uma métrica correspondente de conta e bucket no CloudWatch.	Consulte métricas
Amazon Elastic Block Store (Amazon EBS)	Para o Amazon EBS on Outposts, você pode escolher um AWSOutpost como destino de snapshots e armazená-los localmente no S3 on Outposts.	Saiba mais
Amazon Relational Database Service (Amazon RDS)	Você pode usar backups locais do Amazon RDS para armazenar seus backups do RDS localmente no Outpost.	Saiba mais

Desenvolver com o Amazon S3 on Outposts

Com o Amazon S3 on Outposts, é possível criar buckets do S3 no AWS Outposts, além de armazenar e recuperar facilmente objetos no local para aplicações que exigem acesso a dados locais, processamento de dados local e residência de dados. O S3 on Outposts fornece uma nova classe de armazenamento, o S3 Outposts (`OUTPOSTS`), que usa as APIs do Amazon S3 e é projetado para armazenar dados de forma duradoura e redundante em vários dispositivos e servidores em seu AWS Outposts. Você se comunica com o bucket do Outposts usando um ponto de acesso e uma conexão de endpoint em uma nuvem privada virtual (VPC). É possível usar os mesmos recursos e APIs nos buckets do Outposts da mesma maneira que em buckets do Amazon S3, incluindo políticas de acesso, criptografia e marcação. Só é possível usar o S3 on Outposts por meio do AWS Management Console, da AWS Command Line Interface (AWS CLI), de AWS SDKs ou da API REST. Para obter mais informações, consulte [O que é o Amazon S3 on Outposts? \(p. 1318\)](#)

Os tópicos a seguir fornecem informações sobre desenvolvimento com o S3 on Outposts.

Tópicos

- [Operações de API do Amazon S3 on Outposts \(p. 1404\)](#)
- [Configurar o cliente de controle do S3 para S3 on Outposts usando o SDK para Java \(p. 1406\)](#)

Operações de API do Amazon S3 on Outposts

Este tópico lista as operações de API do Amazon S3, do Amazon S3 Control e do Amazon S3 on Outposts que você pode usar com o Amazon S3 on Outposts.

Tópicos

- [Operações de API do Amazon S3 para gerenciar objetos \(p. 1405\)](#)
- [Operações de API do Amazon S3 Control para gerenciar buckets \(p. 1405\)](#)

- Operações de API do S3 on Outposts para gerenciar endpoints (p. 1406)

Operações de API do Amazon S3 para gerenciar objetos

O S3 on Outposts foi projetado para usar as mesmas operações de API de objetos que o Amazon S3. Portanto, é possível usar a maioria de seus códigos e muitas de suas políticas existentes transmitindo o nome do recurso da Amazon (ARN) do S3 no Outposts como identificador.

O Amazon S3 on Outposts é compatível com as seguintes operações de API do Amazon S3:

- [AbortMultipartUpload](#)
- [CompleteMultipartUpload](#)
- [CopyObject](#)
- [CreateMultipartUpload](#)
- [DeleteObject](#)
- [DeleteObjects](#)
- [DeleteObjectTagging](#)
- [GetObject](#)
- [GetObjectTagging](#)
- [HeadBucket](#)
- [HeadObject](#)
- [ListMultipartUploads](#)
- [ListObjects](#)
- [ListObjectsV2](#)
- [ListParts](#)
- [PutObject](#)
- [PutObjectTagging](#)
- [UploadPart](#)
- [UploadPartCopy](#)

Operações de API do Amazon S3 Control para gerenciar buckets

O S3 on Outposts é compatível com as operações de API a seguir do Amazon S3 Control para lidar com buckets.

- [CreateAccessPoint](#)
- [CreateBucket](#)
- [DeleteAccessPoint](#)
- [DeleteAccessPointPolicy](#)
- [DeleteBucket](#)
- [DeleteBucketLifecycleConfiguration](#)
- [DeleteBucketPolicy](#)
- [DeleteBucketTagging](#)
- [GetAccessPoint](#)
- [GetAccessPointPolicy](#)
- [GetBucket](#)
- [GetBucketLifecycleConfiguration](#)
- [GetBucketPolicy](#)

- [GetBucketTagging](#)
- [ListAccessPoints](#)
- [ListRegionalBuckets](#)
- [PutAccessPointPolicy](#)
- [PutBucketLifecycleConfiguration](#)
- [PutBucketPolicy](#)
- [PutBucketTagging](#)

Operações de API do S3 on Outposts para gerenciar endpoints

O S3 on Outposts é compatível com as operações de API a seguir do Amazon S3 on Outposts para o gerenciamento de endpoints.

- [CreateEndpoint](#)
- [DeleteEndpoint](#)
- [ListEndpoints](#)
- [ListSharedEndpoints](#)

Configurar o cliente de controle do S3 para S3 on Outposts usando o SDK para Java

No exemplo a seguir, o cliente de controle do Amazon S3 é configurado para o Amazon S3 on Outposts com o uso do AWS SDK for Java. Para usar esse exemplo, substitua cada *user input placeholder* por suas próprias informações.

```
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;

public AWSS3Control createS3ControlClient() {

    String accessKey = AWSAccessKey;
    String secretKey = SecretAccessKey;
    BasicAWSCredentials awsCreds = new BasicAWSCredentials(accessKey, secretKey);

    return AWSS3ControlClient.builder().enableUseArnRegion()
        .withCredentials(new AWSStaticCredentialsProvider(awsCreds))
        .build();
}
```

Exemplos de código do Amazon S3 usando AWS SDKs

Os exemplos de código a seguir mostram como usar o Amazon S3 com um Kit de Desenvolvimento de Software (SDK) da AWS.

Os exemplos são divididos nas seguintes categorias:

Ações

Trechos de código que mostram como chamar funções de serviço individuais.

Cenários

Exemplos de código que mostram como realizar uma tarefa específica chamando várias funções dentro do mesmo serviço.

Exemplos entre serviços

Amostras de aplicações que funcionam em vários Serviços da AWS.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de código

- [Ações do Amazon S3 usando AWS SDKs \(p. 1408\)](#)
 - [Adicionar regras de CORS a um bucket do Amazon S3 usando um AWS SDK \(p. 1409\)](#)
 - [Adicionar uma configuração de ciclo de vida a um bucket do Amazon S3 usando um AWS SDK \(p. 1414\)](#)
 - [Adicionar uma política a um bucket do Amazon S3 usando um AWS SDK \(p. 1417\)](#)
 - [Copiar um objeto de um bucket do Amazon S3 para outro usando um AWS SDK \(p. 1422\)](#)
 - [Criar um bucket do Amazon S3 usando um AWS SDK \(p. 1431\)](#)
 - [Excluir regras de CORS de um bucket do Amazon S3 usando um AWS SDK \(p. 1440\)](#)
 - [Excluir uma política de um bucket do Amazon S3 usando um AWS SDK \(p. 1441\)](#)
 - [Excluir um bucket vazio do Amazon S3 usando um AWS SDK \(p. 1445\)](#)
 - [Excluir um objeto do Amazon S3 usando um AWS SDK \(p. 1451\)](#)
 - [Excluir vários objetos de um bucket do Amazon S3 usando um AWS SDK \(p. 1457\)](#)
 - [Excluir a configuração de ciclo de vida de um bucket do Amazon S3 usando um AWS SDK \(p. 1465\)](#)
 - [Excluir a configuração de site de um bucket do Amazon S3 usando um AWS SDK \(p. 1466\)](#)
 - [Determinar a existência e o tipo de conteúdo de um objeto em um bucket do Amazon S3 usando um AWS SDK \(p. 1469\)](#)
 - [Determinar a existência de um bucket do Amazon S3 usando um AWS SDK \(p. 1470\)](#)
 - [Obter regras de CORS para um bucket do Amazon S3 usando um AWS SDK \(p. 1471\)](#)
 - [Obter um objeto de um bucket do Amazon S3 usando um AWS SDK \(p. 1473\)](#)
 - [Obter a ACL de um bucket do Amazon S3 usando um AWS SDK \(p. 1483\)](#)

- [Obter a ACL de um bucket do Amazon S3 usando um AWS SDK \(p. 1487\)](#)
- [Obter a região em que reside o bucket do Amazon S3 usando um AWS SDK \(p. 1491\)](#)
- [Obter a configuração de ciclo de vida de um bucket do Amazon S3 usando um AWS SDK \(p. 1492\)](#)
- [Obter a política de um bucket do Amazon S3 usando um AWS SDK \(p. 1493\)](#)
- [Obter a configuração de site de um bucket do Amazon S3 usando um AWS SDK \(p. 1497\)](#)
- [Listar buckets do Amazon S3 usando um AWS SDK \(p. 1499\)](#)
- [Listar carregamentos fracionados em andamento para um bucket do Amazon S3 usando um AWS SDK \(p. 1504\)](#)
- [Listar a versão de objetos em um bucket do Amazon S3 usando um AWS SDK \(p. 1504\)](#)
- [Listar objetos em um bucket do Amazon S3 usando um AWS SDK \(p. 1505\)](#)
- [Restaurar uma cópia arquivada de um objeto em um bucket do Amazon S3 usando um AWS SDK \(p. 1514\)](#)
- [Definir uma nova ACL para um bucket do Amazon S3 usando um AWS SDK \(p. 1515\)](#)
- [Definir a ACL de um objeto do Amazon S3 usando um AWS SDK \(p. 1518\)](#)
- [Definir a configuração de site de um bucket do Amazon S3 usando um AWS SDK \(p. 1519\)](#)
- [Carregar um objeto em um bucket do Amazon S3 usando um AWS SDK \(p. 1523\)](#)
- [Cenários do Amazon S3 usando AWS SDKs \(p. 1537\)](#)
 - [Criar um URL pré-assinado para o Amazon S3 usando um AWS SDK \(p. 1537\)](#)
 - [Começar a usar buckets e objetos do Amazon S3 usando um AWS SDK \(p. 1545\)](#)
 - [Gerenciar objetos do Amazon S3 versionados em lotes com uma função do Lambda usando um AWS SDK \(p. 1580\)](#)
 - [Usar um gerenciador de transferências para carregar e baixar arquivos de e para o Amazon S3 usando um AWS SDK \(p. 1581\)](#)
 - [Trabalhar com objetos versionados do Amazon S3 usando um AWS SDK \(p. 1594\)](#)
- [Exemplos do Amazon S3 entre serviços usando AWS SDKs \(p. 1599\)](#)
 - [Criar uma aplicação Amazon Transcribe \(p. 1599\)](#)
 - [Converter texto em fala e de volta em texto usando um AWS SDK \(p. 1600\)](#)
 - [Criar um cluster do Amazon EMR de longa duração e executar várias etapas usando um AWS SDK \(p. 1600\)](#)
 - [Criar um cluster do Amazon EMR de curta duração e executar uma etapa usando um AWS SDK \(p. 1601\)](#)
 - [Criar uma aplicação de exploração do Amazon Textract \(p. 1602\)](#)
 - [Detectar EPI em imagens com o Amazon Rekognition usando um AWS SDK \(p. 1603\)](#)
 - [Detectar entidades em texto extraído de uma imagem usando um AWS SDK \(p. 1604\)](#)
 - [Detectar faces em uma imagem usando um AWS SDK \(p. 1604\)](#)
 - [Detectar objetos em imagens com o Amazon Rekognition usando um AWS SDK \(p. 1605\)](#)
 - [Detectar pessoas e objetos em um vídeo com o Amazon Rekognition usando um AWS SDK \(p. 1607\)](#)
 - [Salvar o EXIF e outras informações de imagem usando um AWS SDK \(p. 1608\)](#)

Ações do Amazon S3 usando AWS SDKs

Os exemplos de código a seguir demonstram como realizar ações específicas do Amazon S3 com AWS SDKs. Estes trechos chamam a API do Amazon S3 e não devem ser executados isoladamente. Cada exemplo inclui um link para o GitHub, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Os exemplos a seguir incluem apenas as ações mais usadas. Para obter uma lista completa, consulte a Referência de API do Amazon Simple Storage Service.

Exemplos

- Adicionar regras de CORS a um bucket do Amazon S3 usando um AWS SDK (p. 1409)
- Adicionar uma configuração de ciclo de vida a um bucket do Amazon S3 usando um AWS SDK (p. 1414)
- Adicionar uma política a um bucket do Amazon S3 usando um AWS SDK (p. 1417)
- Copiar um objeto de um bucket do Amazon S3 para outro usando um AWS SDK (p. 1422)
- Criar um bucket do Amazon S3 usando um AWS SDK (p. 1431)
- Excluir regras de CORS de um bucket do Amazon S3 usando um AWS SDK (p. 1440)
- Excluir uma política de um bucket do Amazon S3 usando um AWS SDK (p. 1441)
- Excluir um bucket vazio do Amazon S3 usando um AWS SDK (p. 1445)
- Excluir um objeto do Amazon S3 usando um AWS SDK (p. 1451)
- Excluir vários objetos de um bucket do Amazon S3 usando um AWS SDK (p. 1457)
- Excluir a configuração de ciclo de vida de um bucket do Amazon S3 usando um AWS SDK (p. 1465)
- Excluir a configuração de site de um bucket do Amazon S3 usando um AWS SDK (p. 1466)
- Determinar a existência e o tipo de conteúdo de um objeto em um bucket do Amazon S3 usando um AWS SDK (p. 1469)
- Determinar a existência de um bucket do Amazon S3 usando um AWS SDK (p. 1470)
- Obter regras de CORS para um bucket do Amazon S3 usando um AWS SDK (p. 1471)
- Obter um objeto de um bucket do Amazon S3 usando um AWS SDK (p. 1473)
- Obter a ACL de um bucket do Amazon S3 usando um AWS SDK (p. 1483)
- Obter a ACL de um bucket do Amazon S3 usando um AWS SDK (p. 1487)
- Obter a região em que reside o bucket do Amazon S3 usando um AWS SDK (p. 1491)
- Obter a configuração de ciclo de vida de um bucket do Amazon S3 usando um AWS SDK (p. 1492)
- Obter a política de um bucket do Amazon S3 usando um AWS SDK (p. 1493)
- Obter a configuração de site de um bucket do Amazon S3 usando um AWS SDK (p. 1497)
- Listar buckets do Amazon S3 usando um AWS SDK (p. 1499)
- Listar carregamentos fracionados em andamento para um bucket do Amazon S3 usando um AWS SDK (p. 1504)
- Listar a versão de objetos em um bucket do Amazon S3 usando um AWS SDK (p. 1504)
- Listar objetos em um bucket do Amazon S3 usando um AWS SDK (p. 1505)
- Restaurar uma cópia arquivada de um objeto em um bucket do Amazon S3 usando um AWS SDK (p. 1514)
- Definir uma nova ACL para um bucket do Amazon S3 usando um AWS SDK (p. 1515)
- Definir a ACL de um objeto do Amazon S3 usando um AWS SDK (p. 1518)
- Definir a configuração de site de um bucket do Amazon S3 usando um AWS SDK (p. 1519)
- Carregar um objeto em um bucket do Amazon S3 usando um AWS SDK (p. 1523)

Adicionar regras de CORS a um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como adicionar regras de compartilhamento de recursos de origem cruzada (CORS) a um bucket do S3.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static void deleteBucketCorsInformation(S3Client s3, String bucketName,
String accountId) {
    try {
        DeleteBucketCorsRequest bucketCorsRequest =
DeleteBucketCorsRequest.builder()
        .bucket(bucketName)
        .expectedBucketOwner(accountId)
        .build();

        s3.deleteBucketCors(bucketCorsRequest) ;

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void getBucketCorsInformation(S3Client s3, String bucketName,
String accountId) {

    try {
        GetBucketCorsRequest bucketCorsRequest = GetBucketCorsRequest.builder()
        .bucket(bucketName)
        .expectedBucketOwner(accountId)
        .build();

        GetBucketCorsResponse corsResponse =
s3.getBucketCors(bucketCorsRequest);
        List<CORSRule> corsRules = corsResponse.corsRules();
        for (CORSRule rule: corsRules) {
            System.out.println("allowOrigins: "+rule.allowedOrigins());
            System.out.println("AllowedMethod: "+rule.allowedMethods());
        }
    } catch (S3Exception e) {

        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void setCorsInformation(S3Client s3, String bucketName, String
accountId) {

    List<String> allowMethods = new ArrayList<>();
    allowMethods.add("PUT");
    allowMethods.add("POST");
    allowMethods.add("DELETE");

    List<String> allowOrigins = new ArrayList<>();
    allowOrigins.add("http://example.com");
    try {
        // Define CORS rules.
        CORSRule corsRule = CORSRule.builder()
        .allowedMethods(allowMethods)
        .allowedOrigins(allowOrigins)
        .build();
    }
}
```

```
List<CORSRule> corsRules = new ArrayList<>();
corsRules.add(corsRule);
CORSConfiguration configuration = CORSConfiguration.builder()
    .corsRules(corsRules)
    .build();

PutBucketCorsRequest putBucketCorsRequest =
PutBucketCorsRequest.builder()
    .bucket(bucketName)
    .corsConfiguration(configuration)
    .expectedBucketOwner(accountId)
    .build();

s3.putBucketCors(putBucketCorsRequest);

} catch (S3Exception e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
}
```

- Para obter detalhes da API, consulte [PutBucketCors](#) na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.
import { S3Client } from "@aws-sdk/client-s3";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an Amazon S3 service client object.
const s3Client = new S3Client({ region: REGION });
export { s3Client };
```

Adicione uma regra de CORS.

```
// Import required AWS-SDK clients and commands for Node.js.
import { PutBucketCorsCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
Amazon S3 service client module.

// Set parameters.
// Create initial parameters JSON for putBucketCors.
const thisConfig = {
    AllowedHeaders: ["Authorization"],
    AllowedMethods: [],
    AllowedOrigins: ["*"],
    ExposeHeaders: [],
    MaxAgeSeconds: 3000,
};

// Assemble the list of allowed methods based on command line parameters
```

```
const allowedMethods = [];
process.argv.forEach(function (val, index, array) {
  if (val.toUpperCase() === "POST") {
    allowedMethods.push("POST");
  }
  if (val.toUpperCase() === "GET") {
    allowedMethods.push("GET");
  }
  if (val.toUpperCase() === "PUT") {
    allowedMethods.push("PUT");
  }
  if (val.toUpperCase() === "PATCH") {
    allowedMethods.push("PATCH");
  }
  if (val.toUpperCase() === "DELETE") {
    allowedMethods.push("DELETE");
  }
  if (val.toUpperCase() === "HEAD") {
    allowedMethods.push("HEAD");
  }
});
// Copy the array of allowed methods into the config object
thisConfig.AllowMethods = allowedMethods;

// Create an array of configs then add the config object to it.
const corsRules = new Array(thisConfig);

// Create CORS parameters.
export const corsParams = {
  Bucket: "BUCKET_NAME",
  CORSConfiguration: { CORSRules: corsRules },
};

export async function run() {
  try {
    const data = await s3Client.send(new PutBucketCorsCommand(corsParams));
    console.log("Success", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
}
run();
```

- Para obter mais informações, consulte o [AWS SDK for JavaScript Guia do desenvolvedor](#).
- Para obter detalhes da API, consulte [PutBucketCors](#) na Referência da API do AWS SDK for JavaScript.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""
    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
```

```
        that wraps bucket actions in a class-like structure.  
        """  
        self.bucket = bucket  
        self.name = bucket.name  
  
    def put_cors(self, cors_rules):  
        """  
        Apply CORS rules to the bucket. CORS rules specify the HTTP actions that  
are  
allowed from other domains.  
  
:param cors_rules: The CORS rules to apply.  
"""  
        try:  
            self.bucket.Cors().put(CORSConfiguration={'CORSRules': cors_rules})  
            logger.info(  
                "Put CORS rules %s for bucket '%s'.", cors_rules, self.bucket.name)  
        except ClientError:  
            logger.exception("Couldn't put CORS rules for bucket %s.",  
self.bucket.name)  
            raise
```

- Para obter detalhes da API, consulte [PutBucketCors](#) na Referência da API do AWS SDK para Python (Boto3).

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
require "aws-sdk-s3"  
  
# Wraps Amazon S3 bucket CORS configuration.  
class BucketCorsWrapper  
    attr_reader :bucket_cors  
  
    # @param bucket_cors [Aws::S3::BucketCors] A bucket CORS object configured with  
an existing bucket.  
    def initialize(bucket_cors)  
        @bucket_cors = bucket_cors  
    end  
  
    # Sets CORS rules on a bucket.  
    #  
    # @param allowed_methods [Array<String>] The types of HTTP requests to allow.  
    # @param allowed_origins [Array<String>] The origins to allow.  
    # @returns [Boolean] True if the CORS rules were set; otherwise, false.  
    def set_cors(allowed_methods, allowed_origins)  
        @bucket_cors.put(  
            cors_configuration: {  
                cors_rules: [  
                    {  
                        allowed_methods: allowed_methods,  
                        allowed_origins: allowed_origins,  
                        allowed_headers: %w[*],  
                        max_age_seconds: 3600  
                    }  
                ]  
            }  
    
```

```
)  
    true  
rescue Aws::Errors::ServiceError => e  
    puts "Couldn't set CORS rules for #{@bucket_cors.bucket.name}. Here's why:  
#{e.message}"  
    false  
end  
  
end
```

- Para obter detalhes da API, consulte [PutBucketCors](#) na Referência da API do AWS SDK for Ruby.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Adicionar uma configuração de ciclo de vida a um bucket do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como adicionar a configuração de ciclo de vida a um bucket do S3.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static void setLifecycleConfig(S3Client s3, String bucketName, String  
accountId) {  
  
    try {  
        // Create a rule to archive objects with the "glacierobjects/" prefix  
        // to Amazon S3 Glacier.  
        LifecycleRuleFilter ruleFilter = LifecycleRuleFilter.builder()  
            .prefix("glacierobjects/")  
            .build();  
  
        Transition transition = Transition.builder()  
            .storageClass(TransitionStorageClass.GLACIER)  
            .days(0)  
            .build();  
  
        LifecycleRule rule1 = LifecycleRule.builder()  
            .id("Archive immediately rule")  
            .filter(ruleFilter)  
            .transitions(transition)  
            .status(ExpirationStatus.ENABLED)  
            .build();  
  
        // Create a second rule.  
        Transition transition2 = Transition.builder()  
            .storageClass(TransitionStorageClass.GLACIER)  
            .days(0)  
            .build();  
  
        List<Transition> transitionList = new ArrayList<>();  
        transitionList.add(transition2);
```

```
LifecycleRuleFilter ruleFilter2 = LifecycleRuleFilter.builder()  
    .prefix("glacierobjects/")  
    .build();  
  
LifecycleRule rule2 = LifecycleRule.builder()  
    .id("Archive and then delete rule")  
    .filter(ruleFilter2)  
    .transitions(transitionList)  
    .status(ExpirationStatus.ENABLED)  
    .build();  
  
// Add the LifecycleRule objects to an ArrayList.  
ArrayList<LifecycleRule> ruleList = new ArrayList<>();  
ruleList.add(rule1);  
ruleList.add(rule2);  
  
BucketLifecycleConfiguration lifecycleConfiguration =  
BucketLifecycleConfiguration.builder()  
    .rules(ruleList)  
    .build();  
  
PutBucketLifecycleConfigurationRequest  
putBucketLifecycleConfigurationRequest =  
PutBucketLifecycleConfigurationRequest.builder()  
    .bucket(bucketName)  
    .lifecycleConfiguration(lifecycleConfiguration)  
    .expectedBucketOwner(accountId)  
    .build();  
  
s3.putBucketLifecycleConfiguration(putBucketLifecycleConfigurationRequest);  
  
} catch (S3Exception e) {  
    System.err.println(e.awsErrorDetails().errorMessage());  
    System.exit(1);  
}  
}  
  
// Retrieve the configuration and add a new rule.  
public static void getLifecycleConfig(S3Client s3, String bucketName, String  
accountId){  
  
    try {  
        GetBucketLifecycleConfigurationRequest  
getBucketLifecycleConfigurationRequest =  
GetBucketLifecycleConfigurationRequest.builder()  
    .bucket(bucketName)  
    .expectedBucketOwner(accountId)  
    .build();  
  
        GetBucketLifecycleConfigurationResponse response =  
s3.getBucketLifecycleConfiguration(getBucketLifecycleConfigurationRequest);  
        List<LifecycleRule> newList = new ArrayList<>();  
        List<LifecycleRule> rules = response.rules();  
        for (LifecycleRule rule: rules) {  
            newList.add(rule);  
        }  
  
        // Add a new rule with both a prefix predicate and a tag predicate.  
        LifecycleRuleFilter ruleFilter = LifecycleRuleFilter.builder()  
            .prefix("YearlyDocuments/")  
            .build();  
  
        Transition transition = Transition.builder()  
            .storageClass(TransitionStorageClass.GLACIER)
```

```
.days(3650)
.build();

LifecycleRule rule1 = LifecycleRule.builder()
.id("NewRule")
.filter(ruleFilter)
.transitions(transition)
.status(ExpirationStatus.ENABLED)
.build();

// Add the new rule to the list.
newList.add(rule1);
BucketLifecycleConfiguration lifecycleConfiguration =
BucketLifecycleConfiguration.builder()
.rules(newList)
.build();

PutBucketLifecycleConfigurationRequest
putBucketLifecycleConfigurationRequest =
PutBucketLifecycleConfigurationRequest.builder()
.bucket(bucketName)
.lifecycleConfiguration(lifecycleConfiguration)
.expectedBucketOwner(accountId)
.build();

s3.putBucketLifecycleConfiguration(putBucketLifecycleConfigurationRequest);

} catch (S3Exception e) {
System.err.println(e.awsErrorDetails().errorMessage());
System.exit(1);
}
}

// Delete the configuration from the Amazon S3 bucket.
public static void deleteLifecycleConfig(S3Client s3, String bucketName, String
accountId) {

try {
DeleteBucketLifecycleRequest deleteBucketLifecycleRequest =
DeleteBucketLifecycleRequest.builder()
.bucket(bucketName)
.expectedBucketOwner(accountId)
.build();

s3.deleteBucketLifecycle(deleteBucketLifecycleRequest);

} catch (S3Exception e) {
System.err.println(e.awsErrorDetails().errorMessage());
System.exit(1);
}
}
```

- Para obter detalhes da API, consulte [PutBucketLifecycleConfiguration](#) em Referência da API do AWS SDK for Java 2.x .

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class BucketWrapper:  
    """Encapsulates S3 bucket actions."""  
    def __init__(self, bucket):  
        """  
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in  
        Boto3  
                    that wraps bucket actions in a class-like structure.  
        """  
        self.bucket = bucket  
        self.name = bucket.name  
  
    def put_lifecycle_configuration(self, lifecycle_rules):  
        """  
        Apply a lifecycle configuration to the bucket. The lifecycle configuration  
        can  
            be used to archive or delete the objects in the bucket according to  
        specified  
            parameters, such as a number of days.  
  
        :param lifecycle_rules: The lifecycle rules to apply.  
        """  
        try:  
            self.bucket.LifecycleConfiguration().put(  
                LifecycleConfiguration={'Rules': lifecycle_rules})  
            logger.info(  
                "Put lifecycle rules %s for bucket '%s'.", lifecycle_rules,  
                self.bucket.name)  
        except ClientError:  
            logger.exception(  
                "Couldn't put lifecycle rules for bucket '%s'.", self.bucket.name)  
            raise
```

- Para obter detalhes da API, consulte [PutBucketLifecycleConfiguration](#) na Referência da API do AWS SDK para Python (Boto3).

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Adicionar uma política a um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como adicionar uma política a um bucket do S3.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
bool AwsDoc::S3::PutBucketPolicy(const Aws::String& bucketName,  
                                 const Aws::String& policyBody,  
                                 const Aws::String& region)  
{
```

```
Aws::Client::ClientConfiguration config;

if (!region.empty())
{
    config.region = region;
}

Aws::S3::S3Client s3_client(config);

std::shared_ptr<Aws::StringStream> request_body =
    Aws::MakeShared<Aws::StringStream>("");
*request_body << policyBody;

Aws::S3::Model::PutBucketPolicyRequest request;
request.SetBucket(bucketName);
request.SetBody(request_body);

Aws::S3::Model::PutBucketPolicyOutcome outcome =
    s3_client.PutBucketPolicy(request);

if (outcome.IsSuccess()) {
    std::cout << "Set the following policy body for the bucket '" <<
        bucketName << ":" << std::endl << std::endl;
    std::cout << policyBody << std::endl;

    return true;
}
else {
    std::cout << "Error: PutBucketPolicy: "
        << outcome.GetError().GetMessage() << std::endl;

    return false;
}
}

int main()
{
    Aws::SDKOptions options;
    Aws::InitAPI(options);
    {

        //TODO: Change bucket_name to the name of a bucket in your account.
        const Aws::String bucket_name = "DOC-EXAMPLE-BUCKET";
        //TODO: Set to the AWS Region in which the bucket was created.
        const Aws::String region = "us-east-1";

        // Get the caller's AWS account ID to be used in the bucket policy.
        Aws::STS::STSCient sts_client;
        Aws::STS::Model::GetCallerIdentityRequest request;
        Aws::STS::Model::GetCallerIdentityOutcome outcome =
            sts_client.GetCallerIdentity(request);

        if (!outcome.IsSuccess())
        {
            std::cout << "Error: GetBucketPolicy setup: Get identity information:
"
            << outcome.GetError().GetMessage() << std::endl;

            return 1;
        }

        // Extract the caller's AWS account ID from the call to AWS STS.
        Aws::String account_id = outcome.GetResult().GetAccount();

        // Use the account ID and bucket name to form the bucket policy to be
        added.
```

```
Aws::String policy_string = GetPolicyString(account_id, bucket_name);

if (!AwsDoc::S3::PutBucketPolicy(bucket_name, policy_string, region))
{
    return 1;
}
Aws::ShutdownAPI(options);

return 0;
}
```

- Para obter detalhes da API, consulte [PutBucketPolicy](#) na Referência da API do AWS SDK for C++.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static void setPolicy(S3Client s3, String bucketName, String policyText)
{

    System.out.println("Setting policy:");
    System.out.println("----");
    System.out.println(policyText);
    System.out.println("----");
    System.out.format("On Amazon S3 bucket: \"%s\"\n", bucketName);

    try {
        PutBucketPolicyRequest policyReq = PutBucketPolicyRequest.builder()
            .bucket(bucketName)
            .policy(policyText)
            .build();

        s3.putBucketPolicy(policyReq);

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }

    System.out.println("Done!");
}

// Loads a JSON-formatted policy from a file
public static String getBucketPolicyFromFile(String policyFile) {

    StringBuilder fileText = new StringBuilder();
    try {
        List<String> lines = Files.readAllLines(Paths.get(policyFile),
StandardCharsets.UTF_8);
        for (String line : lines) {
            fileText.append(line);
        }
    } catch (IOException e) {
        System.out.format("Problem reading file: \"%s\"", policyFile);
    }
}
```

```
        System.out.println(e.getMessage());
    }

    try {
        final JsonParser parser = new
ObjectMapper().getFactory().createParser(fileText.toString());
        while (parser.nextToken() != null) {
        }

    } catch (IOException jpe) {
        jpe.printStackTrace();
    }
    return fileText.toString();
}
```

- Para obter detalhes da API, consulte [PutBucketPolicy](#) na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.
import { S3Client } from "@aws-sdk/client-s3";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an Amazon S3 service client object.
const s3Client = new S3Client({ region: REGION });
export { s3Client };
```

Adicione a política.

```
// Import required AWS SDK clients and commands for Node.js.
import { CreateBucketCommand, PutBucketPolicyCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
Amazon S3 service client module.

const BUCKET_NAME = "BUCKET_NAME";
export const bucketParams = {
  Bucket: BUCKET_NAME,
};
// Create the policy in JSON for the S3 bucket.
const readOnlyAnonUserPolicy = {
  Version: "2012-10-17",
  Statement: [
    {
      Sid: "AddPerm",
      Effect: "Allow",
      Principal: "*",
      Action: ["s3:GetObject"],
      Resource: ["*"],
    },
  ],
};
```

```
// Create selected bucket resource string for bucket policy.
const bucketResource = "arn:aws:s3:::" + BUCKET_NAME + "/*"; //BUCKET_NAME
readOnlyAnonUserPolicy.Statement[0].Resource[0] = bucketResource;

// Convert policy JSON into string and assign into parameters.
const bucketPolicyParams = {
    Bucket: BUCKET_NAME,
    Policy: JSON.stringify(readOnlyAnonUserPolicy),
};

export const run = async () => {
    try {
        const data = await s3Client.send(
            new CreateBucketCommand(bucketParams)
        );
        console.log('Success, bucket created.', data)
        try {
            const response = await s3Client.send(
                new PutBucketPolicyCommand(bucketPolicyParams)
            );
            console.log("Success, permissions added to bucket", response);
            return response;
        }
        catch (err) {
            console.log("Error adding policy to S3 bucket.", err);
        }
    } catch (err) {
        console.log("Error creating S3 bucket.", err);
    }
};
run();
```

- Para obter mais informações, consulte o [AWS SDK for JavaScript Guia do desenvolvedor](#).
- Para obter detalhes da API, consulte [PutBucketPolicy](#) na Referência da API do AWS SDK for JavaScript.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""
    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                    that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def put_policy(self, policy):
        """
        Apply a security policy to the bucket. Policies control users' ability
        to perform specific actions, such as listing the objects in the bucket.
```

```
:param policy: The policy to apply to the bucket.  
"""  
try:  
    self.bucket.Policy().put(Policy=json.dumps(policy))  
    logger.info("Put policy %s for bucket '%s'.", policy, self.bucket.name)  
except ClientError:  
    logger.exception("Couldn't apply policy to bucket '%s'.",  
self.bucket.name)  
    raise
```

- Para obter detalhes da API, consulte [PutBucketPolicy](#) na Referência da API do AWS SDK para Python (Boto3).

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
# Wraps an Amazon S3 bucket policy.  
class BucketPolicyWrapper  
    attr_reader :bucket_policy  
  
    # @param bucket_policy [Aws::S3::BucketPolicy] A bucket policy object configured  
    # with an existing bucket.  
    def initialize(bucket_policy)  
        @bucket_policy = bucket_policy  
    end  
  
    # Sets a policy on a bucket.  
    #  
    def set_policy(policy)  
        @bucket_policy.put(policy: policy)  
        true  
    rescue Aws::Errors::ServiceError => e  
        puts "Couldn't set the policy for #{@bucket_policy.bucket.name}. Here's why:  
#{e.message}"  
        false  
    end  
end
```

- Para obter detalhes da API, consulte [PutBucketPolicy](#) na Referência da API do AWS SDK for Ruby.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Copiar um objeto de um bucket do Amazon S3 para outro usando um AWS SDK

Os exemplos de código a seguir mostram como copiar um objeto do S3 de um bucket para outro.

.NET

AWS SDK for .NET

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
/// <summary>
/// Copies an object in an Amazon S3 bucket to a folder within the
/// same bucket.
/// </summary>
/// <param name="client">An initialized Amazon S3 client object.</param>
/// <param name="bucketName">The name of the Amazon S3 bucket where the
/// object to copy is located.</param>
/// <param name="objectName">The object to be copied.</param>
/// <param name="folderName">The folder to which the object will
/// be copied.</param>
/// <returns>A boolean value that indicates the success or failure of
/// the copy operation.</returns>
public static async Task<bool> CopyObjectInBucketAsync(
    IAmazonS3 client,
    string bucketName,
    string objectName,
    string folderName)
{
    try
    {
        var request = new CopyObjectRequest
        {
            SourceBucket = bucketName,
            SourceKey = objectName,
            DestinationBucket = bucketName,
            DestinationKey = $"{folderName}\\{objectName}",
        };
        var response = await client.CopyObjectAsync(request);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error copying object: '{ex.Message}'");
        return false;
    }
}
```

- Para obter detalhes da API, consulte [CopyObject](#) em Referência de API do AWS SDK for .NET.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
bool AwsDoc::S3::CopyObject(const Aws::String &objectKey, const Aws::String &fromBucket, const Aws::String &toBucket,
                            const Aws::String &region) {
    Aws::Client::ClientConfiguration clientConfig;
    if (!region.empty()) {
        clientConfig.region = region;
    }

    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::CopyObjectRequest request;

    request.WithCopySource(fromBucket + "/" + objectKey)
        .WithKey(objectKey)
        .WithBucket(toBucket);

    Aws::S3::Model::CopyObjectOutcome outcome = client.CopyObject(request);

    if (!outcome.IsSuccess())
    {
        const auto err = outcome.GetError();
        std::cout << "Error: CopyObject: " <<
                    err.GetExceptionName() << ":" << err.GetMessage() << std::endl;
        return false;
    }

    return true;
}

int main() {
    Aws::SDKOptions options;
    Aws::InitAPI(options);

    //TODO: Name of object already in bucket.
    Aws::String objectKey = "<enter object key>";

    //TODO: Change from _bucket to the name of your bucket that already contains
    //      "my-file.txt".
    Aws::String fromBucket = "<Enter bucket name>";

    //TODO: Change to the name of another bucket in your account.
    Aws::String toBucket = "<Enter bucket name>";

    //TODO: Set to the AWS Region in which the bucket was created.
    Aws::String region = "us-east-1";

    AwsDoc::S3::CopyObject(objectKey, fromBucket, toBucket, region);

    ShutdownAPI(options);
    return 0;
}
```

- Para obter detalhes da API, consulte [CopyObject](#) em Referência de API do AWS SDK for C++.

Go

SDK for Go V2

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
// Copy an object to another name.
```

```
// CopyObject is "Pull an object from the source bucket + path".
// The semantics of CopySource varies depending on whether you're using Amazon S3
on Outposts,
// or through access points.
// See https://docs.aws.amazon.com/AmazonS3/latest/API/
API_CopyObject.html#API_CopyObject_RequestSyntax
fmt.Println("Copy an object from another bucket to our bucket.")
-, err = client.CopyObject(context.TODO(), &s3.CopyObjectInput{
Bucket:      aws.String(name),
CopySource:  aws.String(name + "/path/myfile.jpg"),
Key:         aws.String("other/file.jpg"),
})

if err != nil {
panic("Couldn't copy the object to a new key")
}
```

- Para obter detalhes da API, consulte [CopyObject](#) em Referência de API do AWS SDK for Go.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static String copyBucketObject (S3Client s3, String fromBucket, String
objectKey, String toBucket) {

    String encodedUrl = "";
    try {
        encodedUrl = URLEncoder.encode(fromBucket + "/" + objectKey,
StandardCharsets.UTF_8.toString());

    } catch (UnsupportedEncodingException e) {
        System.out.println("URL could not be encoded: " + e.getMessage());
    }

    CopyObjectRequest copyReq = CopyObjectRequest.builder()
        .copySourceIfMatch(encodedUrl)
        .destinationBucket(toBucket)
        .destinationKey(objectKey)
        .build();

    try {
        CopyObjectResponse copyRes = s3.copyObject(copyReq);
        return copyRes.copyObjectResult().toString();

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

- Para obter detalhes da API, consulte [CopyObject](#) em Referência de API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.
import { S3Client } from "@aws-sdk/client-s3";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an Amazon S3 service client object.
const s3Client = new S3Client({ region: REGION });
export { s3Client };
```

Copie o objeto.

```
// Get service clients module and commands using ES6 syntax.
import { CopyObjectCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js";

// Set the bucket parameters.

export const params = {
  Bucket: "DESTINATION_BUCKET_NAME",
  CopySource: "/SOURCE_BUCKET_NAME/OBJECT_NAME",
  Key: "OBJECT_NAME"
};

// Create the Amazon S3 bucket.
export const run = async () => {
  try {
    const data = await s3Client.send(new CopyObjectCommand(params));
    console.log("Success", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Para obter detalhes da API, consulte [CopyObject](#) em Referência de API do AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

Note

Essa documentação é de pré-lançamento para um recurso em versão de pré-visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
suspend fun copyBucketObject(
```

```
        fromBucket: String,
        objectKey: String,
        toBucket: String
    ) {

        var encodedUrl = ""
        try {
            encodedUrl = URLEncoder.encode("$fromBucket/$objectKey",
StandardCharsets.UTF_8.toString())
        } catch (e: UnsupportedEncodingException) {
            println("URL could not be encoded: " + e.message)
        }

        val request = CopyObjectRequest {
            copySource = encodedUrl
            bucket = toBucket
            key = objectKey
        }
        S3Client { region = "us-east-1" }.use { s3 ->
            s3.copyObject(request)
        }
    }
}
```

- Para obter detalhes da API, consulte [CopyObject](#) na Referência da API do AWS SDK para Kotlin.

PHP

SDK for PHP

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Cópia simples de um objeto.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2', 'version' => 'latest']);

try {
    $folder = "copied-folder";
    $s3client->copyObject([
        'Bucket' => $bucket_name,
        'CopySource' => "$bucket_name/$file_name",
        'Key' => "$folder/$file_name-copy",
    ]);
    echo "Copied $file_name to $folder/$file_name-copy.\n";
} catch (Exception $exception) {
    echo "Failed to copy $file_name with error: " . $exception->getMessage();
    exit("Please fix error with object copying before continuing.");
}
```

- Para obter detalhes da API, consulte [CopyObject](#) em Referência de API do AWS SDK for PHP.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class ObjectWrapper:  
    """Encapsulates S3 object actions."""  
    def __init__(self, s3_object):  
        """  
        :param s3_object: A Boto3 Object resource. This is a high-level resource in  
        Boto3  
                    that wraps object actions in a class-like structure.  
        """  
        self.object = s3_object  
        self.key = self.object.key  
  
    def copy(self, dest_object):  
        """  
        Copies the object to another bucket.  
  
        :param dest_object: The destination object initialized with a bucket and  
        key.  
                    This is a Boto3 Object resource.  
        """  
        try:  
            dest_object.copy_from(CopySource={  
                'Bucket': self.object.bucket_name,  
                'Key': self.object.key  
            })  
            dest_object.wait_until_exists()  
            logger.info(  
                "Copied object from %s:%s to %s:%s.",  
                self.object.bucket_name, self.object.key,  
                dest_object.bucket_name, dest_object.key)  
        except ClientError:  
            logger.exception(  
                "Couldn't copy object from %s/%s to %s/%s.",  
                self.object.bucket_name, self.object.key,  
                dest_object.bucket_name, dest_object.key)  
            raise
```

- Para obter detalhes da API, consulte [CopyObject](#) na Referência da API do AWS SDK para Python (Boto3).

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Copie um objeto.

```
require "aws-sdk-s3"  
  
# Wraps Amazon S3 object actions.  
class ObjectCopyWrapper  
    attr_reader :source_object  
  
    # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is  
    used as the source object for  
    #                                         copy actions.  
    def initialize(source_object)  
        @source_object = source_object
```

```
end

# Copy the source object to the specified target bucket and rename it with the
target key.
#
# @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
object is copied.
# @param target_object_key [String] The key to give the copy of the object.
# @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
nil.
def copy_object(target_bucket, target_object_key)
  @source_object.copy_to(bucket: target_bucket.name, key: target_object_key)
  target_bucket.object(target_object_key)
rescue Aws::Errors::ServiceError => e
  puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
#{e.message}"
end
end

# Replace the source and target bucket names with existing buckets you own and
replace the source object key
# with an existing object in the source bucket.
def run_demo
  source_bucket_name = "doc-example-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "doc-example-bucket2"
  target_key = "my-target-file.txt"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key)
  return unless target_object

  puts "Copied #{source_key} from #{source_bucket_name} to
#{target_object.bucket_name}:#{target_object.key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Copie um objeto e adicione criptografia no lado do servidor ao objeto de destino.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyEncryptWrapper
  attr_reader :source_object

  # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is
  used as the source object for
  #                                         copy actions.
  def initialize(source_object)
    @source_object = source_object
  end

  # Copy the source object to the specified target bucket, rename it with the
  target key, and encrypt it.
  #
  # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
  object is copied.
  # @param target_object_key [String] The key to give the copy of the object.
  # @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
  nil.
  def copy_object(target_bucket, target_object_key, encryption)
```

```
    @source_object.copy_to(bucket: target_bucket.name, key: target_object_key,
server_side_encryption: encryption)
    target_bucket.object(target_object_key)
rescue Aws::Errors::ServiceError => e
    puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
#{e.message}"
end
end

# Replace the source and target bucket names with existing buckets you own and
# replace the source object key
# with an existing object in the source bucket.
def run_demo
    source_bucket_name = "doc-example-bucket1"
    source_key = "my-source-file.txt"
    target_bucket_name = "doc-example-bucket2"
    target_key = "my-target-file.txt"
    target_encryption = "AES256"

    source_bucket = Aws::S3::Bucket.new(source_bucket_name)
    wrapper = ObjectCopyEncryptWrapper.new(source_bucket.object(source_key))
    target_bucket = Aws::S3::Bucket.new(target_bucket_name)
    target_object = wrapper.copy_object(target_bucket, target_key, target_encryption)
    return unless target_object

    puts "Copied #{source_key} from #{source_bucket_name} to
#{target_object.bucket_name}:#{target_object.key} and \"\
        encrypted the target with #{target_object.server_side_encryption}
encryption."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- Para obter detalhes da API, consulte [CopyObject](#) em Referência de API do AWS SDK for Ruby.

Rust

SDK for Rust

Note

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
pub async fn copy_object(
    client: &Client,
    bucket_name: &str,
    object_key: &str,
    target_key: &str,
) -> Result<(), Error> {
    let mut source_bucket_and_object: String = "".to_owned();
    source_bucket_and_object.push_str(bucket_name);
    source_bucket_and_object.push('/');
    source_bucket_and_object.push_str(object_key);

    client
        .copy_object()
```

```
.copy_source(source_bucket_and_object)
.bucket(bucket_name)
.key(target_key)
.send()
.await?;

Ok(())
}
```

- Para obter detalhes da API, consulte [CopyObject](#) em Referência de API do AWS SDK para Rust.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public func copyFile(from sourceBucket: String, name: String, to destBucket: String) async throws {
    let srcUrl = "\(sourceBucket)/\("\(name)").addingPercentEncoding(withAllowedCharacters: .urlPathAllowed)

    let input = CopyObjectInput(
        bucket: destBucket,
        copySource: srcUrl,
        key: name
    )
    _ = try await client.copyObject(input: input)
}
```

- Para obter detalhes da API, consulte [CopyObject](#) na Referência da API do AWS SDK para Swift.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Criar um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como criar um bucket do S3.

.NET

AWS SDK for .NET

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
/// <summary>
/// Shows how to create a new Amazon S3 bucket.
/// </summary>
/// <param name="client">An initialized Amazon S3 client object.</param>
/// <param name="bucketName">The name of the bucket to create.</param>
/// <returns>A boolean value representing the success or failure of
/// the bucket creation process.</returns>
public static async Task<bool> CreateBucketAsync(IAmazonS3 client, string
bucketName)
{
    try
    {
        var request = new PutBucketRequest
        {
            BucketName = bucketName,
            UseClientRegion = true,
        };

        var response = await client.PutBucketAsync(request);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error creating bucket: '{ex.Message}'");
        return false;
    }
}
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API do AWS SDK for .NET.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
bool AwsDoc::S3::CreateBucket(const Aws::String &bucketName, const Aws::String
&region) {
    // Create the bucket.
    Aws::Client::ClientConfiguration clientConfig;
    if (!region.empty())
        clientConfig.region = region;

    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::CreateBucketRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::CreateBucketOutcome outcome = client.CreateBucket(request);
    if (!outcome.IsSuccess())
    {
        auto err = outcome.GetError();
        std::cout << "Error: CreateBucket: " <<
                    err.GetExceptionName() << ":" << err.GetMessage() << std::endl;
        return false;
    }
}
```

```
        }
    else
    {
        std::cout << "Created bucket " << bucketName <<
                    " in the specified AWS Region." << std::endl;
        return true;
    }

}
int main()
{
    Aws::SDKOptions options;
    InitAPI(options);
    //TODO: Set to the AWS Region of your account. If you don't, you will get
    a runtime
    //IllegalLocationConstraintException Message: "The unspecified location
    constraint is incompatible
    //for the Region specific endpoint this request was sent to."
    Aws::String region = "us-east-1";
    // Create a unique bucket name to increase the chance of success
    // when trying to create the bucket.
    // Format: "doc-example-bucket-" + lowercase UUID.
    Aws::String uuid = Aws::Utils::UUID::RandomUUID();
    Aws::String bucketName = "doc-example-bucket-" +
                            Aws::Utils::StringUtils::ToLower(uuid.c_str());

    AwsDoc::S3::CreateBucket(bucketName, region);

    ShutdownAPI(options);
}
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API do AWS SDK for C++.

Go

SDK for Go V2

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
// Create a bucket: We're going to create a bucket to hold content.
// Best practice is to use the preset private access control list (ACL).
// If you are not creating a bucket from us-east-1, you must specify a bucket
location constraint.
// Bucket names must conform to several rules; read more at https://
docs.aws.amazon.com/AmazonS3/latest/userguide/bucketnamingrules.html
_, err := client.CreateBucket(context.TODO(), &s3.CreateBucketInput{
    Bucket:                 aws.String(name),
    ACL:                     types.BucketCannedACLPrivate,
    CreateBucketConfiguration: &types.CreateBucketConfiguration{LocationConstraint:
types.BucketLocationConstraintUsWest2},
})

if err != nil {
    panic("could not create bucket: " + err.Error())
}
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API do AWS SDK for Go.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static void createBucket( S3Client s3Client, String bucketName) {

    try {
        S3Waiter s3Waiter = s3Client.waiter();
        CreateBucketRequest bucketRequest = CreateBucketRequest.builder()
            .bucket(bucketName)
            .build();

        s3Client.createBucket(bucketRequest);
        HeadBucketRequest bucketRequestWait = HeadBucketRequest.builder()
            .bucket(bucketName)
            .build();

        // Wait until the bucket is created and print out the response.
        WaiterResponse<HeadBucketResponse> waiterResponse =
s3Waiter.waitUntilBucketExists(bucketRequestWait);
        waiterResponse.matched().response().ifPresent(System.out::println);
        System.out.println(bucketName + " is ready");

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.
import { S3Client } from "@aws-sdk/client-s3";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an Amazon S3 service client object.
const s3Client = new S3Client({ region: REGION });
export { s3Client };
```

Crie o bucket.

```
// Get service clients module and commands using ES6 syntax.
import { CreateBucketCommand } from "@aws-sdk/client-s3";
```

```
import { s3Client } from "./libs/s3Client.js";

// Set the bucket parameters.

export const bucketParams = { Bucket: "BUCKET_NAME" };

// Create the Amazon S3 bucket.
export const run = async () => {
    try {
        const data = await s3Client.send(new CreateBucketCommand(bucketParams));
        console.log("Success", data);
        return data; // For unit tests.
    } catch (err) {
        console.log("Error", err);
    }
};
run();
```

- Para obter mais informações, consulte o [AWS SDK for JavaScript Guia do desenvolvedor](#).
- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API do AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

Note

Essa documentação é de pré-lançamento para um recurso em versão de pré-visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
suspend fun createNewBucket(bucketName: String) {

    val request = CreateBucketRequest {
        bucket = bucketName
    }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.createBucket(request)
        println("$bucketName is ready")
    }
}
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API do AWS SDK para Kotlin.

PHP

SDK for PHP

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Criar um bucket.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2', 'version' => 'latest']);

try {
    $s3client->createBucket([
        'Bucket' => $bucket_name,
        'CreateBucketConfiguration' => ['LocationConstraint' => $region],
    ]);
    echo "Created bucket named: $bucket_name \n";
} catch (Exception $exception) {
    echo "Failed to create bucket $bucket_name with error: " . $exception-
>getMessage();
    exit("Please fix error with bucket creation before continuing.");
}
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API do AWS SDK for PHP.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie um bucket com as configurações padrão.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""
    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                    that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def create(self, region_override=None):
        """
        Create an Amazon S3 bucket in the default Region for the account or in the
        specified Region.

        :param region_override: The Region in which to create the bucket. If this
        is
                    not specified, the Region configured in your shared
                    credentials is used.
        """
        if region_override is not None:
            region = region_override
        else:
            region = self.bucket.meta.client.meta.region_name
        try:
            self.bucket.create(
                CreateBucketConfiguration={'LocationConstraint': region})

            self.bucket.wait_until_exists()
            logger.info(
                "Created bucket '%s' in region=%s", self.bucket.name, region)
        except ClientError as error:
```

```
logger.exception(
    "Couldn't create bucket named '%s' in region=%s.",
    self.bucket.name, region)
raise error
```

Crie um bucket versionado com uma configuração de ciclo de vida.

```
def create_versioned_bucket(bucket_name, prefix):
    """
    Creates an Amazon S3 bucket, enables it for versioning, and configures a
    lifecycle
    that expires noncurrent object versions after 7 days.

    Adding a lifecycle configuration to a versioned bucket is a best practice.
    It helps prevent objects in the bucket from accumulating a large number of
    noncurrent versions, which can slow down request performance.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket_name: The name of the bucket to create.
    :param prefix: Identifies which objects are automatically expired under the
                   configured lifecycle rules.
    :return: The newly created bucket.
    """
    try:
        bucket = s3.create_bucket(
            Bucket=bucket_name,
            CreateBucketConfiguration={
                'LocationConstraint': s3.meta.client.meta.region_name
            }
        )
        logger.info("Created bucket %s.", bucket.name)
    except ClientError as error:
        if error.response['Error']['Code'] == 'BucketAlreadyOwnedByYou':
            logger.warning("Bucket %s already exists! Using it.", bucket_name)
            bucket = s3.Bucket(bucket_name)
        else:
            logger.exception("Couldn't create bucket %s.", bucket_name)
            raise

    try:
        bucket.Versioning().enable()
        logger.info("Enabled versioning on bucket %s.", bucket.name)
    except ClientError:
        logger.exception("Couldn't enable versioning on bucket %s.", bucket.name)
        raise

    try:
        expiration = 7
        bucket.LifecycleConfiguration().put(
            LifecycleConfiguration={
                'Rules': [
                    {
                        'Status': 'Enabled',
                        'Prefix': prefix,
                        'NoncurrentVersionExpiration': {'NoncurrentDays': expiration}
                    }
                ]
            }
        )
        logger.info("Configured lifecycle to expire noncurrent versions after %s
days "
                   "on bucket %s.", expiration, bucket.name)
    except ClientError as error:
        logger.warning("Couldn't configure lifecycle on bucket %s because %s. "
```

```
    "Continuing anyway.", bucket.name, error)

return bucket
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API do AWS SDK para Python (Boto3).

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
require "aws-sdk-s3"

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

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

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

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

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

  puts "Created bucket #{wrapper.bucket.name}."
```

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

run_demo if $PROGRAM_NAME == __FILE__
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API do AWS SDK for Ruby.

Rust

SDK for Rust

Note

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
pub async fn create_bucket(client: &Client, bucket_name: &str, region: &str) ->
Result<(), Error> {
    let constraint = BucketLocationConstraint::from(region);
    let cfg = CreateBucketConfiguration::builder()
        .location_constraint(constraint)
        .build();
    client
        .create_bucket()
        .create_bucket_configuration(cfg)
        .bucket(bucket_name)
        .send()
        .await?;
    println!("{}", bucket_name);
    Ok(())
}
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API do AWS SDK para Rust.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public func createBucket(name: String) async throws {
    let config = S3ClientTypes.CreateBucketConfiguration(
```

```
        locationConstraint: .usEast2
    )
let input = CreateBucketInput(
    bucket: name,
    createBucketConfiguration: config
)
_ = try await client.createBucket(input: input)
}
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API do AWS SDK para Swift.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Excluir regras de CORS de um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como excluir regras de CORS de um bucket do S3.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""
    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                    that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def delete_cors(self):
        """
        Delete the CORS rules from the bucket.

        :param bucket_name: The name of the bucket to update.
        """
        try:
            self.bucket.Cors().delete()
            logger.info("Deleted CORS from bucket '%s'.", self.bucket.name)
        except ClientError:
            logger.exception("Couldn't delete CORS from bucket '%s'.",
                            self.bucket.name)
            raise
```

- Para obter detalhes da API, consulte [DeleteBucketCors](#) na Referência da API do AWS SDK para Python (Boto3).

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket CORS configuration.
class BucketCorsWrapper
  attr_reader :bucket_cors

  # @param bucket_cors [Aws::S3::BucketCors] A bucket CORS object configured with
  # an existing bucket.
  def initialize(bucket_cors)
    @bucket_cors = bucket_cors
  end

  # Deletes the CORS configuration of a bucket.
  #
  # @return [Boolean] True if the CORS rules were deleted; otherwise, false.
  def delete_cors
    @bucket_cors.delete
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't delete CORS rules for #{@bucket_cors.bucket.name}. Here's why:
#{e.message}"
    false
  end
end
```

- Para obter detalhes da API, consulte [DeleteBucketCors](#) na Referência da API do AWS SDK for Ruby.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Excluir uma política de um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como excluir uma política de um bucket do S3.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
bool AwsDoc::S3::DeleteBucketPolicy(const Aws::String &bucketName, const
Aws::String &region) {
```

```
Aws::Client::ClientConfiguration clientConfig;
if (!region.empty()) {
    clientConfig.region = region;
}

Aws::S3::S3Client client(clientConfig);

Aws::S3::Model::DeleteBucketPolicyRequest request;
request.SetBucket(bucketName);

Aws::S3::Model::DeleteBucketPolicyOutcome outcome =
client.DeleteBucketPolicy(request);

if (!outcome.IsSuccess())
{
    auto err = outcome.GetError();
    std::cout << "Error: DeleteBucketPolicy: " <<
        err.GetExceptionName() << ":" << err.GetMessage() << std::endl;

    return false;
}
else
{
    std::cout << "Policy was deleted from the bucket." << std::endl;
    return true;
}

int main()
{
    //TODO: Change bucket_name to the name of a bucket in your account.
    const Aws::String bucketName = "<Enter bucket name>";
    //TODO: Set to the AWS Region in which the bucket was created.
    const Aws::String region = "us-east-1";

    Aws::SDKOptions options;
    Aws::InitAPI(options);

    AwsDoc::S3::DeleteBucketPolicy(bucketName, region);

    ShutdownAPI(options);
}
```

- Para obter detalhes da API, consulte [DeleteBucketPolicy](#) na Referência da API do AWS SDK for C++.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
// Delete the bucket policy.
public static void deleteS3BucketPolicy(S3Client s3, String bucketName) {

    DeleteBucketPolicyRequest delReq = DeleteBucketPolicyRequest.builder()
        .bucket(bucketName)
        .build();
```

```
        try {
            s3.deleteBucketPolicy(delReq);
            System.out.println("Done!");
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte [DeleteBucketPolicy](#) na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.
import { S3Client } from "@aws-sdk/client-s3";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an Amazon S3 service client object.
const s3Client = new S3Client({ region: REGION });
export { s3Client };
```

Exclua a política de bucket.

```
// Import required AWS SDK clients and commands for Node.js.
import { DeleteBucketPolicyCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
Amazon S3 service client module.

// Set the bucket parameters
export const bucketParams = { Bucket: "BUCKET_NAME" };

export const run = async () => {
    try {
        const data = await s3Client.send(new DeleteBucketPolicyCommand(bucketParams));
        console.log("Success", data + ", bucket policy deleted");
        return data; // For unit tests.
    } catch (err) {
        console.log("Error", err);
    }
};
// Invoke run() so these examples run out of the box.
run();
```

- Para obter mais informações, consulte o [AWS SDK for JavaScript Guia do desenvolvedor](#).
- Para obter detalhes da API, consulte [DeleteBucketPolicy](#) na Referência da API do AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

Note

Essa documentação é de pré-lançamento para um recurso em versão de pré-visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
suspend fun deleteS3BucketPolicy(bucketName: String?) {  
  
    val request = DeleteBucketPolicyRequest {  
        bucket = bucketName  
    }  
  
    S3Client { region = "us-east-1" }.use { s3 ->  
        s3.deleteBucketPolicy(request)  
        println("Done!")  
    }  
}
```

- Para obter detalhes da API, consulte [DeleteBucketPolicy](#) na Referência da API do AWS SDK para Kotlin.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class BucketWrapper:  
    """Encapsulates S3 bucket actions."""  
    def __init__(self, bucket):  
        """  
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in  
        Boto3  
                    that wraps bucket actions in a class-like structure.  
        """  
        self.bucket = bucket  
        self.name = bucket.name  
  
    def delete_policy(self):  
        """  
        Delete the security policy from the bucket.  
        """  
        try:  
            self.bucket.Policy().delete()  
            logger.info("Deleted policy for bucket '%s'.", self.bucket.name)  
        except ClientError:  
            logger.exception("Couldn't delete policy for bucket '%s'.",  
self.bucket.name)  
            raise
```

- Para obter detalhes da API, consulte [DeleteBucketPolicy](#) na Referência da API do AWS SDK para Python (Boto3).

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
# Wraps an Amazon S3 bucket policy.
class BucketPolicyWrapper
  attr_reader :bucket_policy

  # @param bucket_policy [Aws::S3::BucketPolicy] A bucket policy object configured
  # with an existing bucket.
  def initialize(bucket_policy)
    @bucket_policy = bucket_policy
  end

  def delete_policy
    @bucket_policy.delete
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't delete the policy from #{@bucket_policy.bucket.name}. Here's
why: #{e.message}"
    false
  end
end
```

- Para obter detalhes da API, consulte [DeleteBucketPolicy](#) na Referência da API do AWS SDK for Ruby.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Excluir um bucket vazio do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como excluir um bucket vazio do S3.

.NET

AWS SDK for .NET

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
/// <summary>
/// Shows how to delete an Amazon S3 bucket.
/// </summary>
```

```
    /// <param name="client">An initialized Amazon S3 client object.</param>
    /// <param name="bucketName">The name of the Amazon S3 bucket to delete.</param>
    /// <returns>A boolean value that represents the success or failure of
    /// the delete operation.</returns>
    public static async Task<bool> DeleteBucketAsync(IAmazonS3 client, string
bucketName)
    {
        var request = new DeleteBucketRequest
        {
            BucketName = bucketName,
        };

        var response = await client.DeleteBucketAsync(request);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API do AWS SDK for .NET.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
using namespace Aws;

bool AwsDoc::S3::DeleteBucket(const Aws::String &bucketName, const Aws::String
&region)
{
    Aws::Client::ClientConfiguration clientConfig;
    if (!region.empty())
    {
        clientConfig.region = region;
    }

    Aws::S3::S3Client client(clientConfig);

    Aws::S3::Model::DeleteBucketRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::DeleteBucketOutcome outcome =
        client.DeleteBucket(request);

    if (!outcome.IsSuccess())
    {
        auto err = outcome.GetError();
        std::cout << "Error: DeleteBucket: " <<
                    err.GetExceptionName() << ":" << err.GetMessage() << std::endl;
        return false;
    }
    else
    {
        std::cout << "The bucket was deleted" << std::endl;
        return true;
    }
}
```

```
int main()
{
    //TODO: Change bucket_name to the name of a bucket in your account.
    //If the bucket is not in your account, you will get one of two errors:
    Aws::String bucketName = "<Bucket Name>";
    //TODO: Set to the AWS Region of the bucket bucket_name.
    Aws::String region = "us-east-1";

    Aws::SDKOptions options;
    Aws::InitAPI(options);

    AwsDoc::S3::DeleteBucket(bucketName, region);

    ShutdownAPI(options);
}
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API do AWS SDK for C++.

Go

SDK for Go V2

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
fmt.Println("Delete a bucket")
// Delete the bucket.

_, err = client.DeleteBucket(context.TODO(), &s3.DeleteBucketInput{
    Bucket: aws.String(name),
})
if err != nil {
    panic("Couldn't delete bucket: " + err.Error())
}
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API do AWS SDK for Go.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
DeleteBucketRequest deleteBucketRequest = DeleteBucketRequest.builder()
    .bucket(bucket)
    .build();

s3.deleteBucket(deleteBucketRequest);
s3.close();
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.  
import { S3Client } from "@aws-sdk/client-s3";  
// Set the AWS Region.  
const REGION = "us-east-1";  
// Create an Amazon S3 service client object.  
const s3Client = new S3Client({ region: REGION });  
export { s3Client };
```

Excluir o bucket.

```
// Import required AWS SDK clients and commands for Node.js.  
import { DeleteBucketCommand } from "@aws-sdk/client-s3";  
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an  
Amazon S3 service client module.  
  
// Set the bucket parameters  
export const bucketParams = { Bucket: "BUCKET_NAME" };  
  
export const run = async () => {  
    try {  
        const data = await s3Client.send(new DeleteBucketCommand(bucketParams));  
        return data; // For unit tests.  
        console.log("Success - bucket deleted");  
    } catch (err) {  
        console.log("Error", err);  
    }  
};  
// Invoke run() so these examples run out of the box.  
run();
```

- Para obter mais informações, consulte o [AWS SDK for JavaScript Guia do desenvolvedor](#).
- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API do AWS SDK for JavaScript.

PHP

SDK for PHP

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Exclua um bucket vazio.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2', 'version' => 'latest']);  
  
try {  
    $s3client->deleteBucket([  
        'Bucket' => $bucket_name,  
    ]);
```

```
    echo "Deleted bucket $bucket_name.\n";
} catch (Exception $exception) {
    echo "Failed to delete $bucket_name with error: " . $exception->getMessage();
    exit("Please fix error with bucket deletion before continuing.");
}
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API do AWS SDK for PHP.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""
    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                    that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def delete(self):
        """
        Delete the bucket. The bucket must be empty or an error is raised.
        """
        try:
            self.bucket.delete()
            self.bucket.wait_until_not_exists()
            logger.info("Bucket %s successfully deleted.", self.bucket.name)
        except ClientError:
            logger.exception("Couldn't delete bucket %s.", self.bucket.name)
            raise
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API do AWS SDK para Python (Boto3).

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
# Deletes the objects in an Amazon S3 bucket and deletes the bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket to empty and delete.
def delete_bucket(bucket)
    puts("\nDo you want to delete all of the objects as well as the bucket (y/n)?")
    answer = gets.chomp.downcase
    if answer == "y"
        bucket.objects.batch_delete!
```

```
    bucket.delete
    puts("Emptied and deleted bucket #{bucket.name}.\n")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't empty and delete bucket #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API do AWS SDK for Ruby.

Rust

SDK for Rust

Note

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
pub async fn delete_bucket(client: &Client, bucket_name: &str) -> Result<(), Error>
{
  client.delete_bucket().bucket(bucket_name).send().await?;
  println!("bucket deleted");
  Ok(())
}
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API do AWS SDK para Rust.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public func deleteBucket(name: String) async throws {
  let input = DeleteBucketInput(
    bucket: name
  )
  _ = try await client.deleteBucket(input: input)
}
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API do AWS SDK para Swift.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Excluir um objeto do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como excluir um objeto do S3.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
bool AwsDoc::S3::DeleteObject(const Aws::String &objectKey, const Aws::String
&fromBucket, const Aws::String &region) {
    Aws::Client::ClientConfiguration clientConfig;
    if (!region.empty()) {
        clientConfig.region = region;
    }

    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::DeleteObjectRequest request;

    request.WithKey(objectKey)
        .WithBucket(fromBucket);

    Aws::S3::Model::DeleteObjectOutcome outcome =
        client.DeleteObject(request);

    if (!outcome.IsSuccess())
    {
        auto err = outcome.GetError();
        std::cout << "Error: DeleteObject: " <<
            err.GetExceptionName() << ":" << err.GetMessage() << std::endl;
        return false;
    }
    else
    {
        std::cout << "Successfully deleted the object." << std::endl;
        return true;
    }
}

int main()
{
    //TODO: The object_key is the unique identifier for the object in the bucket.
    In this example set,
    //it is the filename you added in put_object.cpp.
    Aws::String objectKey = "<Enter object key>";
    //TODO: Change from_bucket to the name of a bucket in your account.
    Aws::String fromBucket = "<Enter bucket name>";
    //TODO: Set to the AWS Region in which the bucket was created.
    Aws::String region = "us-east-1";

    Aws::SDKOptions options;
    Aws::InitAPI(options);

    AwsDoc::S3::DeleteObject(objectKey, fromBucket, region);
```

```
        ShutdownAPI(options);

        return 0;
}
```

- Para obter detalhes da API, consulte [DeleteObject](#) na Referência da API do AWS SDK for C++.

Go

SDK for Go V2

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
// Delete a single object.
fmt.Println("Delete an object from a bucket")
_, err := client.DeleteObject(context.TODO(), &s3.DeleteObjectInput{
    Bucket: aws.String(name),
    Key:    aws.String("other/file.jpg"),
})
if err != nil {
    panic("Couldn't delete object!")
}
```

- Para obter detalhes da API, consulte [DeleteObject](#) na Referência da API do AWS SDK for Go.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.
import { S3Client } from "@aws-sdk/client-s3";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an Amazon S3 service client object.
const s3Client = new S3Client({ region: REGION });
export { s3Client };
```

Exclua um objeto.

```
import { DeleteObjectCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js" // Helper function that creates an
                                             Amazon S3 service client module.

export const bucketParams = { Bucket: "BUCKET_NAME", Key: "KEY" };

export const run = async () => {
    try {
        const data = await s3Client.send(new DeleteObjectCommand(bucketParams));
```

```
        console.log("Success. Object deleted.", data);
        return data; // For unit tests.
    } catch (err) {
        console.log("Error", err);
    }
};

run();
```

- Para obter detalhes da API, consulte [DeleteObject](#) na Referência da API do AWS SDK for JavaScript.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Exclua um objeto.

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""
    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource in
        Boto3
                    that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

    def delete(self):
        """
        Deletes the object.
        """
        try:
            self.object.delete()
            self.object.wait_until_not_exists()
            logger.info(
                "Deleted object '%s' from bucket '%s'.",
                self.object.key, self.object.bucket_name)
        except ClientError:
            logger.exception(
                "Couldn't delete object '%s' from bucket '%s'.",
                self.object.key, self.object.bucket_name)
            raise
```

Reverta um objeto para uma versão anterior excluindo versões posteriores do objeto.

```
def rollback_object(bucket, object_key, version_id):
    """
    Rolls back an object to an earlier version by deleting all versions that
    occurred after the specified rollback version.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket: The bucket that holds the object to roll back.
    :param object_key: The object to roll back.
    :param version_id: The version ID to roll back to.
```

```
"""
# Versions must be sorted by last_modified date because delete markers are
# at the end of the list even when they are interspersed in time.
versions = sorted(bucket.object_versions.filter(Prefix=object_key),
                  key=attrgetter('last_modified'), reverse=True)

logger.debug(
    "Got versions:\n%s",
    '\n'.join([f"\t{version.version_id}, last modified {version.last_modified}"
              for version in versions]))

if version_id in [ver.version_id for ver in versions]:
    print(f"Rolling back to version {version_id}")
    for version in versions:
        if version.version_id != version_id:
            version.delete()
            print(f"Deleted version {version.version_id}")
        else:
            break

    print(f"Active version is now {bucket.Object(object_key).version_id}")
else:
    raise KeyError(f"{version_id} was not found in the list of versions for "
                   f"{object_key}.")
```

Restaure um objeto excluído removendo o marcador de exclusão ativo do objeto.

```
def revive_object(bucket, object_key):
    """
    Revives a versioned object that was deleted by removing the object's active
    delete marker.
    A versioned object presents as deleted when its latest version is a delete
    marker.
    By removing the delete marker, we make the previous version the latest version
    and the object then presents as *not* deleted.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to revive.
    """
    # Get the latest version for the object.
    response = s3.meta.client.list_object_versions(
        Bucket=bucket.name, Prefix=object_key, MaxKeys=1)

    if 'DeleteMarkers' in response:
        latest_version = response['DeleteMarkers'][0]
        if latest_version['IsLatest']:
            logger.info("Object %s was indeed deleted on %s. Let's revive it.",
                        object_key, latest_version['LastModified'])
            obj = bucket.Object(object_key)
            obj.Version(latest_version['VersionId']).delete()
            logger.info("Revived %s, active version is now %s with body '%s'",
                        object_key, obj.version_id, obj.get()['Body'].read())
        else:
            logger.warning("Delete marker is not the latest version for %s!",
                           object_key)
    elif 'Versions' in response:
        logger.warning("Got an active version for %s, nothing to do.", object_key)
    else:
        logger.error("Couldn't get any version info for %s.", object_key)
```

Crie um manipulador do Lambda que remova um marcador de exclusão de um objeto do S3. Esse manipulador pode ser usado para limpar com eficiência marcadores de exclusão estranhos em um bucket versionado.

```
import logging
from urllib import parse
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
logger.setLevel('INFO')

s3 = boto3.client('s3')


def lambda_handler(event, context):
    """
    Removes a delete marker from the specified versioned object.

    :param event: The S3 batch event that contains the ID of the delete marker
                  to remove.
    :param context: Context about the event.
    :return: A result structure that Amazon S3 uses to interpret the result of the
            operation. When the result code is TemporaryFailure, S3 retries the
            operation.
    """
    # Parse job parameters from Amazon S3 batch operations
    invocation_id = event['invocationId']
    invocation_schema_version = event['invocationSchemaVersion']

    results = []
    result_code = None
    result_string = None

    task = event['tasks'][0]
    task_id = task['taskId']

    try:
        obj_key = parse.unquote(task['s3Key'], encoding='utf-8')
        obj_version_id = task['s3VersionId']
        bucket_name = task['s3BucketArn'].split(':')[ -1]

        logger.info("Got task: remove delete marker %s from object %s.",
                   obj_version_id, obj_key)

        try:
            # If this call does not raise an error, the object version is not a
            # marker and should not be deleted.
            response = s3.head_object(
                Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id)
            result_code = 'PermanentFailure'
            result_string = f"Object {obj_key}, ID {obj_version_id} is not " \
                           f"a delete marker."
        except ClientError as error:
            delete_marker = error.response['ResponseMetadata']['HTTPHeaders'] \
                           .get('x-amz-delete-marker', 'false')
            if delete_marker == 'true':
                logger.info("Object %s, version %s is a delete marker.",
                           obj_key, obj_version_id)
                try:
                    s3.delete_object(
```

```

        Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id)
    result_code = 'Succeeded'
    result_string = f"Successfully removed delete marker " \
                    f"\{obj_version_id} from object \{obj_key}\."
    logger.info(result_string)
except ClientError as error:
    # Mark request timeout as a temporary failure so it will be
retried.
    if error.response['Error']['Code'] == 'RequestTimeout':
        result_code = 'TemporaryFailure'
        result_string = f"Attempt to remove delete marker from " \
                        f"object \{obj_key} timed out."
        logger.info(result_string)
    else:
        raise
else:
    raise ValueError(f"The x-amz-delete-marker header is either not "
                     f"present or is not 'true'.")
except Exception as error:
    # Mark all other exceptions as permanent failures.
    result_code = 'PermanentFailure'
    result_string = str(error)
    logger.exception(error)
finally:
    results.append({
        'taskId': task_id,
        'resultCode': result_code,
        'resultString': result_string
    })
return {
    'invocationSchemaVersion': invocation_schema_version,
    'treatMissingKeysAs': 'PermanentFailure',
    'invocationId': invocation_id,
    'results': results
}

```

- Para obter detalhes da API, consulte [DeleteObject](#) na Referência da API do AWS SDK para Python (Boto3).

Rust

SDK for Rust

Note

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
async fn remove_object(client: &Client, bucket: &str, key: &str) -> Result<(),  
    Error> {  
    client  
        .delete_object()  
        .bucket(bucket)  
        .key(key)  
        .send()  
        .await?;
```

```
    println!("Object deleted.");
}
Ok(())
}
```

- Para obter detalhes da API, consulte [DeleteObject](#) na Referência da API do AWS SDK para Rust.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public func deleteFile(bucket: String, key: String) async throws {
    let input = DeleteObjectInput(
        bucket: bucket,
        key: key
    )

    do {
        _ = try await client.deleteObject(input: input)
    } catch {
        throw error
    }
}
```

- Para obter detalhes da API, consulte [DeleteObject](#) na Referência da API do AWS SDK para Swift.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Excluir vários objetos de um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como excluir vários objetos de um bucket do S3.

.NET

AWS SDK for .NET

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Exclua todos os objetos de um bucket do S3.

```
/// <summary>
/// Delete all of the objects stored in an existing Amazon S3 bucket.
/// </summary>
/// <param name="client">An initialized Amazon S3 client object.</param>
/// <param name="bucketName">The name of the bucket from which the
/// contents will be deleted.</param>
/// <returns>A boolean value that represents the success or failure of
/// deleting all of the objects in the bucket.</returns>
public static async Task<bool> DeleteBucketContentsAsync(IAmazonS3 client,
string bucketName)
{
    // Iterate over the contents of the bucket and delete all objects.
    var request = new ListObjectsV2Request
    {
        BucketName = bucketName,
    };

    try
    {
        var response = await client.ListObjectsV2Async(request);

        do
        {
            response.S3Objects
                .ForEach(async obj => await
client.DeleteObjectAsync(bucketName, obj.Key));

            // If the response is truncated, set the request
ContinuationToken
                // from the NextContinuationToken property of the response.
                request.ContinuationToken = response.NextContinuationToken;
        }
        while (response.IsTruncated);

        return true;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error deleting objects: {ex.Message}");
        return false;
    }
}
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API do AWS SDK for .NET.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static void deleteBucketObjects(S3Client s3, String bucketName) {

    // Upload three sample objects to the specified Amazon S3 bucket.
```

```
ArrayList<ObjectIdentifier> keys = new ArrayList<>();
PutObjectRequest putOb;
ObjectIdentifier objectId;

for (int i = 0; i < 3; i++) {
    String keyName = "delete object example " + i;
    objectId = ObjectIdentifier.builder()
        .key(keyName)
        .build();

    putOb = PutObjectRequest.builder()
        .bucket(bucketName)
        .key(keyName)
        .build();

    s3.putObject(putOb, RequestBody.fromString(keyName));
    keys.add(objectId);
}

System.out.println(keys.size() + " objects successfully created.");

// Delete multiple objects in one request.
Delete del = Delete.builder()
    .objects(keys)
    .build();

try {
    DeleteObjectsRequest multiObjectDeleteRequest =
DeleteObjectsRequest.builder()
    .bucket(bucketName)
    .delete(del)
    .build();

    s3.deleteObjects(multiObjectDeleteRequest);
    System.out.println("Multiple objects are deleted!");

} catch (S3Exception e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
}
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.
import { S3Client } from "@aws-sdk/client-s3";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an Amazon S3 service client object.
const s3Client = new S3Client({ region: REGION });
export { s3Client };
```

Exclua vários objetos.

```
import { DeleteObjectsCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js" // Helper function that creates an
Amazon S3 service client module.

export const bucketParams = {
  Bucket: "BUCKET_NAME",
  Delete: {
    Objects: [
      {
        Key: "KEY_1",
      },
      {
        Key: "KEY_2",
      },
    ],
  },
};

export const run = async () => {
  try {
    const data = await s3Client.send(new DeleteObjectsCommand(bucketParams));
    return data; // For unit tests.
    console.log("Success. Object deleted.");
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Exclua todos os objetos em um bucket.

```
import { ListObjectsCommand, DeleteObjectCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
Amazon S3 service client module.

export const bucketParams = { Bucket: "BUCKET_NAME" };

export const run = async () => {
  try {
    const data = await s3Client.send(new ListObjectsCommand(bucketParams));
    return data; // For unit tests.
    let i = 0;
    let noOfObjects = data.Contents;
    for (let i = 0; i < noOfObjects.length; i++) {
      const data = await s3Client.send(
        new DeleteObjectCommand({
          Bucket: bucketParams.Bucket,
          Key: noOfObjects[i].Key,
        })
      );
    }
    console.log("Success. Objects deleted.");
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API do AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

Note

Essa documentação é de pré-lançamento para um recurso em versão de pré-visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
suspend fun deleteBucketObjects(bucketName: String, objectName: String) {  
  
    val objectId = ObjectIdentifier {  
        key = objectName  
    }  
  
    val delOb = Delete {  
        objects = listOf(objectId)  
    }  
  
    val request = DeleteObjectsRequest {  
        bucket = bucketName  
        delete = delOb  
    }  
  
    S3Client { region = "us-east-1" }.use { s3 ->  
        s3.deleteObjects(request)  
        println("$objectName was deleted from $bucketName")  
    }  
}
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API do AWS SDK para Kotlin.

PHP

SDK for PHP

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Exclua um conjunto de objetos de uma lista de chaves.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2', 'version' => 'latest']);  
  
try {  
    $objects = [];  
    foreach ($contents['Contents'] as $content) {  
        $objects[] = [  
            'Key' => $content['Key'],  
        ];  
    }  
    $s3client->deleteObjects([  
        'Bucket' => $bucket_name,
```

```
'Key' => $file_name,
'Delete' => [
    'Objects' => $objects,
],
]);
$check = $s3client->listObjects([
    'Bucket' => $bucket_name,
]);
if (count($check) <= 0) {
    throw new Exception("Bucket wasn't empty.");
}
echo "Deleted all objects and folders from $bucket_name.\n";
} catch (Exception $exception) {
    echo "Failed to delete $file_name from $bucket_name with error: " . $exception-
>getMessage();
    exit("Please fix error with object deletion before continuing.");
}
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API do AWS SDK for PHP.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Exclua um conjunto de objetos de uma lista de chaves.

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""
    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource in
        Boto3
                    that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

    @staticmethod
    def delete_objects(bucket, object_keys):
        """
        Removes a list of objects from a bucket.
        This operation is done as a batch in a single request.

        :param bucket: The bucket that contains the objects. This is a Boto3 Bucket
        resource.
        :param object_keys: The list of keys that identify the objects to remove.
        :return: The response that contains data about which objects were deleted
        and any that could not be deleted.
        """
        try:
            response = bucket.delete_objects(Delete={
                'Objects': [
                    {
                        'Key': key
                    } for key in object_keys
                ]
            })
            if 'Deleted' in response:
                logger.info(
                    "Deleted objects '%s' from bucket '%s'.",
                    response['Deleted'],
                    bucket.name
                )
        except Exception as e:
            logger.error("Error deleting objects: %s", str(e))

```

```
[del_obj['Key'] for del_obj in response['Deleted']],
bucket.name)
    if 'Errors' in response:
        logger.warning(
            "Could not delete objects '%s' from bucket '%s'.",
            f"{{del_obj['Key']}: {del_obj['Code']}}"
            for del_obj in response['Errors']),
        bucket.name)
    except ClientError:
        logger.exception("Couldn't delete any objects from bucket %s.",
        bucket.name)
        raise
    else:
        return response
```

Exclua todos os objetos em um bucket.

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""
    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource in
        Boto3
                           that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

    @staticmethod
    def empty_bucket(bucket):
        """
        Remove all objects from a bucket.

        :param bucket: The bucket to empty. This is a Boto3 Bucket resource.
        """
        try:
            bucket.objects.delete()
            logger.info("Emptied bucket '%s'.", bucket.name)
        except ClientError:
            logger.exception("Couldn't empty bucket '%s'.", bucket.name)
            raise
```

Exclua permanentemente um objeto versionado excluindo todas as suas versões.

```
def permanently_delete_object(bucket, object_key):
    """
    Permanently deletes a versioned object by deleting all of its versions.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to delete.
    """
    try:
        bucket.object_versions.filter(Prefix=object_key).delete()
        logger.info("Permanently deleted all versions of object %s.", object_key)
    except ClientError:
        logger.exception("Couldn't delete all versions of %s.", object_key)
        raise
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API do AWS SDK para Python (Boto3).

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
# Deletes the objects in an Amazon S3 bucket and deletes the bucket.  
#  
# @param bucket [Aws::S3::Bucket] The bucket to empty and delete.  
def delete_bucket(bucket)  
    puts("\nDo you want to delete all of the objects as well as the bucket (y/n)?")  
    answer = gets.chomp.downcase  
    if answer == "y"  
        bucket.objects.batch_delete!  
        bucket.delete  
        puts("Emptied and deleted bucket #{bucket.name}.\n")  
    end  
rescue Aws::Errors::ServiceError => e  
    puts("Couldn't empty and delete bucket #{bucket.name}.")  
    puts("\t#{e.code}: #{e.message}")  
    raise  
end
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API do AWS SDK for Ruby.

Rust

SDK for Rust

Note

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
pub async fn delete_objects(client: &Client, bucket_name: &str) -> Result<(), Error> {  
    let objects = client.list_objects_v2().bucket(bucket_name).send().await?  
  
    let mut delete_objects: Vec<ObjectIdentifier> = vec![];  
    for obj in objects.contents().unwrap_or_default() {  
        let obj_id = ObjectIdentifier::builder()  
            .set_key(Some(obj.key().unwrap().to_string()))  
            .build();  
        delete_objects.push(obj_id);  
    }  
    client  
        .delete_objects()  
        .bucket(bucket_name)  
        .delete(Delete::builder().set_objects(Some(delete_objects)).build())
```

```
.send()  
.await?;  
  
let objects: ListObjectsV2Output =  
client.list_objects_v2().bucket(bucket_name).send().await?;  
match objects.key_count {  
    0 => Ok(()),  
    _ => Err(Error::Unhandled(Box::from(  
        "There were still objects left in the bucket.",  
    ))),  
}  
}
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API do AWS SDK para Rust.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public func deleteObjects(bucket: String, keys: [String]) async throws {  
    let input = DeleteObjectsInput(  
        bucket: bucket,  
        delete: S3ClientTypes.Delete(  
            objects: keys.map({ S3ClientTypes.ObjectIdentifier(key: $0) }),  
            quiet: true  
        )  
    )  
  
    do {  
        _ = try await client.deleteObjects(input: input)  
    } catch {  
        throw error  
    }  
}
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API do AWS SDK para Swift.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Excluir a configuração de ciclo de vida de um bucket do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como excluir a configuração de ciclo de vida de um bucket do S3.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""
    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                    that wraps bucket actions in a class-like structure.

        """
        self.bucket = bucket
        self.name = bucket.name

    def delete_lifecycle_configuration(self):
        """
        Remove the lifecycle configuration from the specified bucket.
        """
        try:
            self.bucket.LifecycleConfiguration().delete()
            logger.info(
                "Deleted lifecycle configuration for bucket '%s'.",
                self.bucket.name)
        except ClientError:
            logger.exception(
                "Couldn't delete lifecycle configuration for bucket '%s'.",
                self.bucket.name)
        raise
```

- Para obter detalhes da API, consulte [DeleteBucketLifecycle](#) na Referência da API do AWS SDK para Python (Boto3).

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Excluir a configuração de site de um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como excluir a configuração de site de um bucket do S3.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
bool AwsDoc::S3::DeleteBucketWebsite(const Aws::String &bucketName, const
Aws::String &region) {
    // Create the bucket.
```

```
Aws::Client::ClientConfiguration clientConfig;
if (!region.empty()) {
    clientConfig.region = region;
}

Aws::S3::S3Client client(clientConfig);
Aws::S3::Model::DeleteBucketWebsiteRequest request;
request.SetBucket(bucketName);

Aws::S3::Model::DeleteBucketWebsiteOutcome outcome =
    client.DeleteBucketWebsite(request);

if (!outcome.IsSuccess())
{
    auto err = outcome.GetError();
    std::cout << "Error: DeleteBucketWebsite: " <<
        err.GetExceptionName() << ":" << err.GetMessage() << std::endl;
    return false;
}
else
{
    std::cout << "Website configuration was removed." << std::endl;
    return true;
}
}

int main()
{
    //TODO: Change bucket_name to the name of a bucket in your account.
    const Aws::String bucketName = "<Enter bucket name>";
    //TODO: Set to the AWS Region in which the bucket was created.
    const Aws::String region = "us-east-1";

    Aws::SDKOptions options;
    Aws::InitAPI(options);

    AwsDoc::S3::DeleteBucketWebsite(bucketName, region);

    ShutdownAPI(options);
}
```

- Para obter detalhes da API, consulte [DeleteBucketWebsite](#) na Referência da API do AWS SDK for C++.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static void deleteBucketWebsiteConfig(S3Client s3, String bucketName ) {

    DeleteBucketWebsiteRequest delReq = DeleteBucketWebsiteRequest.builder()
        .bucket(bucketName)
        .build();

    try {
        s3.deleteBucketWebsite(delReq);
    } catch (S3Exception e) {
```

```
        System.err.println(e.awsErrorDetails().errorMessage());
        System.out.println("Failed to delete website configuration!");
        System.exit(1);
    }
}
```

- Para obter detalhes da API, consulte [DeleteBucketWebsite](#) na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.
import { S3Client } from "@aws-sdk/client-s3";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an Amazon S3 service client object.
const s3Client = new S3Client({ region: REGION });
export { s3Client };
```

Exclua a configuração de site do bucket.

```
// Import required AWS SDK clients and commands for Node.js.

import { DeleteBucketWebsiteCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
                                               // Amazon S3 service client module.

// Create the parameters for calling
export const bucketParams = { Bucket: "BUCKET_NAME" };

export const run = async () => {
    try {
        const data = await s3Client.send(new DeleteBucketWebsiteCommand(bucketParams));
        return data; // For unit tests.
        console.log("Success", data);
    } catch (err) {
        console.log("Error", err);
    }
};
run();
```

- Para obter mais informações, consulte o [AWS SDK for JavaScript Guia do desenvolvedor](#).
- Para obter detalhes da API, consulte [DeleteBucketWebsite](#) na Referência da API do AWS SDK for JavaScript.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Determinar a existência e o tipo de conteúdo de um objeto em um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como determinar a existência e o tipo de conteúdo de um objeto em um bucket do S3.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static void getContentType (S3Client s3, String bucketName, String
keyName) {

    try {
        HeadObjectRequest objectRequest = HeadObjectRequest.builder()
            .key(keyName)
            .bucket(bucketName)
            .build();

        HeadObjectResponse objectHead = s3.headObject(objectRequest);
        String type = objectHead.contentType();
        System.out.println("The object content type is "+type);

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Para obter detalhes da API, consulte [HeadObject](#) em Referência da API do AWS SDK for Java 2.x.

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectExistsWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Checks whether the object exists.
```

```
#  
# @return [Boolean] True if the object exists; otherwise false.  
def exists?  
    @object.exists?  
rescue Aws::Errors::ServiceError => e  
    puts "Couldn't check existence of object #{@object.bucket.name}:#{@object.key}.  
Here's why: #{e.message}"  
    false  
end  
  
# Replace bucket name and object key with an existing bucket and object that you  
own.  
def run_demo  
    bucket_name = "doc-example-bucket"  
    object_key = "my-object.txt"  
  
    wrapper = ObjectExistsWrapper.new(Aws::S3::Object.new(bucket_name, object_key))  
    exists = wrapper.exists?  
  
    puts "Object #{object_key} #{exists ? 'does' : 'does not'} exist."  
end  
  
run_demo if $PROGRAM_NAME == __FILE__
```

- Para obter detalhes da API, consulte [HeadObject](#) em Referência da API do AWS SDK for Ruby.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Determinar a existência de um bucket do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como determinar a existência de um bucket do S3.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class BucketWrapper:  
    """Encapsulates S3 bucket actions."""  
    def __init__(self, bucket):  
        """  
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in  
        Boto3  
                    that wraps bucket actions in a class-like structure.  
        """  
        self.bucket = bucket  
        self.name = bucket.name  
  
    def exists(self):  
        """  
        Determine whether the bucket exists and you have access to it.
```

```
:return: True when the bucket exists; otherwise, False.  
"""  
try:  
    self.bucket.meta.client.head_bucket(Bucket=self.bucket.name)  
    logger.info("Bucket %s exists.", self.bucket.name)  
    exists = True  
except ClientError:  
    logger.warning(  
        "Bucket %s doesn't exist or you don't have access to it.",  
        self.bucket.name)  
    exists = False  
return exists
```

- Para obter detalhes da API, consulte [HeadBucket](#) na Referência da API do AWS SDK para Python (Boto3).

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Obter regras de CORS para um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como obter regras de compartilhamento de recursos de origem cruzada (CORS) para um bucket do S3.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.  
import { S3Client } from "@aws-sdk/client-s3";  
// Set the AWS Region.  
const REGION = "us-east-1";  
// Create an Amazon S3 service client object.  
const s3Client = new S3Client({ region: REGION });  
export { s3Client };
```

Obtenha a política de CORS para o bucket.

```
// Import required AWS SDK clients and commands for Node.js.  
import { GetBucketCorsCommand } from "@aws-sdk/client-s3";  
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an  
// Amazon S3 service client module.  
  
// Create the parameters for calling  
export const bucketParams = { Bucket: "BUCKET_NAME" };  
  
export const run = async () => {  
    try {  
        const data = await s3Client.send(new GetBucketCorsCommand(bucketParams));
```

```
        console.log("Success", JSON.stringify(data.CORSRules));
        return data; // For unit tests.
    } catch (err) {
        console.log("Error", err);
    }
};

run();
```

- Para obter mais informações, consulte o [AWS SDK for JavaScript Guia do desenvolvedor](#).
- Para obter detalhes da API, consulte [GetBucketCors](#) na Referência da API do AWS SDK for JavaScript.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""
    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                           that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def get_cors(self):
        """
        Get the CORS rules for the bucket.

        :return The CORS rules for the specified bucket.
        """
        try:
            cors = self.bucket.Cors()
            logger.info(
                "Got CORS rules %s for bucket '%s'.", cors.cors_rules,
                self.bucket.name)
        except ClientError:
            logger.exception(("Couldn't get CORS for bucket %s.", self.bucket.name))
            raise
        else:
            return cors
```

- Para obter detalhes da API, consulte [GetBucketCors](#) na Referência da API do AWS SDK para Python (Boto3).

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket CORS configuration.
class BucketCorsWrapper
  attr_reader :bucket_cors

  # @param bucket_cors [Aws::S3::BucketCors] A bucket CORS object configured with
  # an existing bucket.
  def initialize(bucket_cors)
    @bucket_cors = bucket_cors
  end

  # Gets the CORS configuration of a bucket.
  #
  # @return [Aws::S3::Type::GetBucketCorsOutput, nil] The current CORS
  # configuration for the bucket.
  def get_cors
    @bucket_cors.data
    rescue Aws::Errors::ServiceError => e
      puts "Couldn't get CORS configuration for #{@bucket_cors.bucket.name}. Here's
why: #{e.message}"
      nil
  end
end
```

- Para obter detalhes da API, consulte [GetBucketCors](#) na Referência da API do AWS SDK for Ruby.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Obter um objeto de um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como ler dados de um objeto em um bucket do S3.

.NET

AWS SDK for .NET

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
/// <summary>
/// Shows how to download an object from an Amazon S3 bucket to the
/// local computer.
/// </summary>
/// <param name="client">An initialized Amazon S3 client object.</param>
/// <param name="bucketName">The name of the bucket where the object is
/// currently stored.</param>
/// <param name="objectName">The name of the object to download.</param>
/// <param name="filePath">The path, including filename, where the
/// downloaded object will be stored.</param>
```

```
/// <returns>A boolean value indicating the success or failure of the
/// download process.</returns>
public static async Task<bool> DownloadObjectFromBucketAsync(
    IAmazonS3 client,
    string bucketName,
    string objectName,
    string filePath)
{
    // Create a GetObject request
    var request = new GetObjectRequest
    {
        BucketName = bucketName,
        Key = objectName,
    };

    // Issue request and remember to dispose of the response
    // using GetObjectResponse response = await
    client.GetObjectAsync(request);

    try
    {
        // Save object to local file
        await response.WriteResponseStreamToFileAsync($"{filePath}\\" +
            $"{objectName}", true, System.Threading.CancellationToken.None);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error saving {objectName}: {ex.Message}");
        return false;
    }
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API do AWS SDK for .NET.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
bool AwsDoc::S3::GetObject(const Aws::String& objectKey,
    const Aws::String& fromBucket, const Aws::String& region)
{
    Aws::Client::ClientConfiguration config;

    if (!region.empty())
    {
        config.region = region;
    }

    Aws::S3::S3Client s3_client(config);

    Aws::S3::Model::GetObjectRequest object_request;
    object_request.SetBucket(fromBucket);
    object_request.SetKey(objectKey);

    Aws::S3::Model::GetObjectOutcome get_object_outcome =
```

```
s3_client.GetObject(object_request);

if (get_object_outcome.IsSuccess())
{
    auto& retrieved_file = get_object_outcome.GetResultWithOwnership().
        GetBody();

    // Print a beginning portion of the text file.
    std::cout << "Beginning of file contents:\n";
    char file_data[255] = { 0 };
    retrieved_file.getline(file_data, 254);
    std::cout << file_data << std::endl;

    return true;
}
else
{
    auto err = get_object_outcome.GetError();
    std::cout << "Error: GetObject: " <<
        err.GetExceptionName() << ":" << err.GetMessage() << std::endl;

    return false;
}

int main()
{
    Aws::SDKOptions options;
    Aws::InitAPI(options);
    {

        //TODO: Change bucket_name to the name of a bucket in your account.
        const Aws::String bucket_name = "<Enter bucket name>";

        //TODO: The bucket "DOC-EXAMPLE-BUCKET" must have been created and
        //previously loaded with "my-file.txt".
        //See create_bucket.cpp and put_object.cpp to create a bucket and load an
        object into that bucket.
        const Aws::String object_name = "<Enter object name>";

        //TODO: Set to the AWS Region in which the bucket was created.
        const Aws::String region = "us-east-1";

        if (!AwsDoc::S3::GetObject(object_name, bucket_name, region))
        {
            return 1;
        }
    }
    Aws::ShutdownAPI(options);

    return 0;
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API do AWS SDK for C++.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Leia dados como uma matriz de bytes.

```
public static void getObjectBytes (S3Client s3, String bucketName, String
keyName, String path) {

    try {
        GetObjectRequest objectRequest = GetObjectRequest
            .builder()
            .key(keyName)
            .bucket(bucketName)
            .build();

        ResponseBytes<GetObjectResponse> objectBytes =
s3.getObjectAsBytes(objectRequest);
        byte[] data = objectBytes.asByteArray();

        // Write the data to a local file.
        File myFile = new File(path );
        OutputStream os = new FileOutputStream(myFile);
        os.write(data);
        System.out.println("Successfully obtained bytes from an S3 object");
        os.close();

    } catch (IOException ex) {
        ex.printStackTrace();
    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

Leia etiquetas que pertencem a um objeto.

```
public static void listTags(S3Client s3, String bucketName, String keyName) {

    try {
        GetObjectTaggingRequest getTaggingRequest = GetObjectTaggingRequest
            .builder()
            .key(keyName)
            .bucket(bucketName)
            .build();

        GetObjectTaggingResponse tags = s3.getObjectTagging(getTaggingRequest);
        List<Tag> tagSet= tags.tagSet();
        for (Tag tag : tagSet) {
            System.out.println(tag.key());
            System.out.println(tag.value());
        }

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

Obtenha um URL de um objeto.

```
public static void getURL(S3Client s3, String bucketName, String keyName ) {

    try {
        GetUrlRequest request = GetUrlRequest.builder()
            .bucket(bucketName)
            .key(keyName)
```

```
        .build();

        URL url = s3.utilities().getUrl(request);
        System.out.println("The URL for " + keyName + " is " + url);

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

Obtenha um objeto usando o objeto de cliente S3Presigner.

```
public static void getPresignedUrl(S3Presigner presigner, String bucketName,
String keyName ) {

    try {
        GetObjectRequest getObjectRequest = GetObjectRequest.builder()
            .bucket(bucketName)
            .key(keyName)
            .build();

        GetObjectPresignRequest getObjectPresignRequest =
GetObjectPresignRequest.builder()
            .signatureDuration(Duration.ofMinutes(60))
            .getObjectRequest(getObjectRequest)
            .build();

        PresignedGetObjectRequest presignedGetObjectRequest =
presigner.presignGetObject(getObjectPresignRequest);
        String theUrl = presignedGetObjectRequest.url().toString();
        System.out.println("Presigned URL: " + theUrl);
        HttpURLConnection connection = (HttpURLConnection)
presignedGetObjectRequest.url().openConnection();
        presignedGetObjectRequest.httpRequest().headers().forEach((header,
values) -> {
            values.forEach(value -> {
                connection.addRequestProperty(header, value);
            });
        });

        // Send any request payload that the service needs (not needed when
isBrowserExecutable is true).
        if (presignedGetObjectRequest.signedPayload().isPresent()) {
            connection.setDoOutput(true);

            try (InputStream signedPayload =
presignedGetObjectRequest.signedPayload().get().asInputStream();
                OutputStream httpOutputStream = connection.getOutputStream()) {
                IoUtils.copy(signedPayload, httpOutputStream);
            }
        }

        // Download the result of executing the request.
        try (InputStream content = connection.getInputStream()) {
            System.out.println("Service returned response: ");
            IoUtils.copy(content, System.out);
        }

    } catch (S3Exception | IOException e) {
        e.printStackTrace();
    }
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.  
import { S3Client } from "@aws-sdk/client-s3";  
// Set the AWS Region.  
const REGION = "us-east-1";  
// Create an Amazon S3 service client object.  
const s3Client = new S3Client({ region: REGION });  
export { s3Client };
```

Baixe o objeto.

```
// Import required AWS SDK clients and commands for Node.js.  
import { GetObjectCommand } from "@aws-sdk/client-s3";  
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an  
Amazon S3 service client module.  
  
export const bucketParams = {  
    Bucket: "BUCKET_NAME",  
    Key: "KEY",  
};  
  
export const run = async () => {  
    try {  
        // Create a helper function to convert a ReadableStream to a string.  
        const streamToString = (stream) =>  
            new Promise((resolve, reject) => {  
                const chunks = [];  
                stream.on("data", (chunk) => chunks.push(chunk));  
                stream.on("error", reject);  
                stream.on("end", () => resolve(Buffer.concat(chunks).toString("utf8")));  
            });  
  
        // Get the object} from the Amazon S3 bucket. It is returned as a  
ReadableStream.  
        const data = await s3Client.send(new GetObjectCommand(bucketParams));  
        return data; // For unit tests.  
        // Convert the ReadableStream to a string.  
        const bodyContents = await streamToString(data.Body);  
        console.log(bodyContents);  
        return bodyContents;  
    } catch (err) {  
        console.log("Error", err);  
    }  
};  
run();
```

- Para obter mais informações, consulte o [AWS SDK for JavaScript Guia do desenvolvedor](#).
- Para obter detalhes da API, consulte [GetObject](#) na Referência da API do AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

Note

Essa documentação é de pré-lançamento para um recurso em versão de pré-visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
suspend fun getObjectBytes(bucketName: String, keyName: String, path: String) {  
  
    val request = GetObjectRequest {  
        key = keyName  
        bucket = bucketName  
    }  
  
    S3Client { region = "us-east-1" }.use { s3 ->  
        s3.getObject(request) { resp ->  
            val myFile = File(path)  
            resp.body?.writeToFile(myFile)  
            println("Successfully read $keyName from $bucketName")  
        }  
    }  
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API do AWS SDK for Kotlin.

PHP

SDK for PHP

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Obtenha um objeto.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2', 'version' => 'latest']);  
  
try {  
    $file = $s3client->getObject([  
        'Bucket' => $bucket_name,  
        'Key' => $file_name,  
    ]);  
    $body = $file->get('Body');  
    $body->rewind();  
    echo "Downloaded the file and it begins with: {$body->read(26)}.\n";  
} catch (Exception $exception) {  
    echo "Failed to download $file_name from $bucket_name with error: " .  
    $exception->getMessage();  
    exit("Please fix error with file downloading before continuing.");  
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API do AWS SDK for PHP.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class ObjectWrapper:  
    """Encapsulates S3 object actions."""  
    def __init__(self, s3_object):  
        """  
        :param s3_object: A Boto3 Object resource. This is a high-level resource in  
        Boto3  
                           that wraps object actions in a class-like structure.  
        """  
        self.object = s3_object  
        self.key = self.object.key  
  
    def get(self):  
        """  
        Gets the object.  
  
        :return: The object data in bytes.  
        """  
        try:  
            body = self.object.get()['Body'].read()  
            logger.info(  
                "Got object '%s' from bucket '%s'.",  
                self.object.key, self.object.bucket_name)  
        except ClientError:  
            logger.exception(  
                "Couldn't get object '%s' from bucket '%s'.",  
                self.object.key, self.object.bucket_name)  
            raise  
        else:  
            return body
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API do AWS SDK para Python (Boto3).

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Obtenha um objeto.

```
require "aws-sdk-s3"  
  
# Wraps Amazon S3 object actions.  
class ObjectGetWrapper  
  attr_reader :object
```

```
# @param object [Aws::S3::Object] An existing Amazon S3 object.
def initialize(object)
  @object = object
end

# Gets the object directly to a file.
#
# @param target_path [String] The path to the file where the object is
downloaded.
# @return [Aws::S3::Types::GetObjectOutput, nil] The retrieved object data if
successful; otherwise nil.
def get_object(target_path)
  @object.get(response_target: target_path)
rescue Aws::Errors::ServiceError => e
  puts "Couldn't get object #{@object.key}. Here's why: #{e.message}"
end
end

# Replace bucket name and object key with an existing bucket and object that you
own.
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object.txt"
  target_path = "my-object-as-file.txt"

  wrapper = ObjectGetWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  obj_data = wrapper.get_object(target_path)
  return unless obj_data

  puts "Object #{object_key} (#{obj_data.content_length} bytes) downloaded to
#{target_path}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Obtenha um objeto e relate seu estado de criptografia no lado do servidor.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectGetEncryptionWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Gets the object into memory.
  #
  # @return [Aws::S3::Types::GetObjectOutput, nil] The retrieved object data if
successful; otherwise nil.
  def get_object
    @object.get
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get object #{@object.key}. Here's why: #{e.message}"
  end
end

# Replace bucket name and object key with an existing bucket and object that you
own.
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object.txt"
```

```
wrapper = ObjectGetEncryptionWrapper.new(Aws::S3::Object.new(bucket_name,  
object_key))  
obj_data = wrapper.get_object  
return unless obj_data  
  
encryption = obj_data.server_side_encryption.nil? ? "no" :  
obj_data.server_side_encryption  
puts "Object #{object_key} uses #{encryption} encryption."  
end  
  
run_demo if $PROGRAM_NAME == __FILE__
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API do AWS SDK for Ruby.

Rust

SDK for Rust

Note

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
pub async fn download_object(client: &Client, bucket_name: &str, key: &str) ->  
Result<(), Error> {  
    let resp = client  
        .get_object()  
        .bucket(bucket_name)  
        .key(key)  
        .send()  
        .await?  
    let data = resp.body.collect().await;  
    println!(  
        "Data from downloaded object: {:?}",  
        data.unwrap().into_bytes().slice(0..20)  
    );  
    Ok(())
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API do AWS SDK para Rust.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Baixe um objeto de um bucket para um arquivo local.

```
public func downloadFile(bucket: String, key: String, to: String) async throws {
    let fileUrl = URL(fileURLWithPath: to).appendingPathComponent(key)

    let input = GetObjectInput(
        bucket: bucket,
        key: key
    )
    let output = try await client.getObject(input: input)

    // Get the data stream object. Return immediately if there isn't one.
    guard let body = output.body else {
        return
    }
    let data = body.toBytes().toData()
    try data.write(to: fileUrl)
}
```

Leia um objeto em um objeto Swift Data.

```
public func readFile(bucket: String, key: String) async throws -> Data {
    let input = GetObjectInput(
        bucket: bucket,
        key: key
    )
    let output = try await client.getObject(input: input)

    // Get the stream and return its contents in a `Data` object. If
    // there is no stream, return an empty `Data` object instead.
    guard let body = output.body else {
        return "".data(using: .utf8)!
    }
    let data = body.toBytes().toData()
    return data
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API do AWS SDK para Swift.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Obter a ACL de um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como obter a lista de controle de acesso (ACL) de um bucket do S3.

Go

SDK for Go V2

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
package main
```

```
import (
    "context"
    "flag"
    "fmt"

    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/s3"
)

// S3GetBucketAclAPI defines the interface for the GetBucketAcl function.
// We use this interface to test the function using a mocked service.
type S3GetBucketAclAPI interface {
    GetBucketAcl(ctx context.Context,
        params *s3.GetBucketAclInput,
        optFns ...func(*s3.Options)) (*s3.GetBucketAclOutput, error)
}

// FindBucketAcl retrieves the access control list (ACL) for an Amazon Simple
// Storage Service (Amazon S3) bucket.
// Inputs:
//     c is the context of the method call, which includes the AWS Region
//     api is the interface that defines the method call
//     input defines the input arguments to the service call.
// Output:
//     If success, a GetBucketAclOutput object containing the result of the service
//     call and nil
//     Otherwise, nil and an error from the call to GetBucketAcl
func FindBucketAcl(c context.Context, api S3GetBucketAclAPI, input
    *s3.GetBucketAclInput) (*s3.GetBucketAclOutput, error) {
    return api.GetBucketAcl(c, input)
}

func main() {
    bucket := flag.String("b", "", "The bucket for which the ACL is returned")
    flag.Parse()

    if *bucket == "" {
        fmt.Println("You must supply a bucket name (-b BUCKET)")
        return
    }

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        panic("configuration error, " + err.Error())
    }

    client := s3.NewFromConfig(cfg)

    input := &s3.GetBucketAclInput{
        Bucket: bucket,
    }

    result, err := FindBucketAcl(context.TODO(), client, input)
    if err != nil {
        fmt.Println("Got an error retrieving ACL for " + *bucket)
        return
    }

    fmt.Println("Owner:", *result.Owner.DisplayName)
    fmt.Println("")
    fmt.Println("Grants")

    for _, g := range result.Grants {
        // If we add a canned ACL, the name is nil
        if g.Grantee.DisplayName == nil {

```

```
        fmt.Println("  Grantee:      EVERYONE")
    } else {
        fmt.Println("  Grantee:      ", *g.Grantee.DisplayName)
    }

    fmt.Println("  Type:        ", string(g.Grantee.Type))
    fmt.Println("  Permission:", string(g.Permission))
    fmt.Println("")
}
}
```

- Para obter detalhes da API, consulte [GetBucketAcl](#) na Referência da API do AWS SDK for Go.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static String getBucketACL(S3Client s3, String objectKey, String
bucketName) {

    try {
        GetObjectAclRequest aclReq = GetObjectAclRequest.builder()
            .bucket(bucketName)
            .key(objectKey)
            .build();

        GetObjectAclResponse aclRes = s3.getObjectAcl(aclReq);
        List<Grant> grants = aclRes.grants();
        String grantee = "";
        for (Grant grant : grants) {
            System.out.format("  %s: %s\n", grant.grantee().id(),
grant.permission());
            grantee = grant.grantee().id();
        }

        return grantee;
    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }

    return "";
}
```

- Para obter detalhes da API, consulte [GetBucketAcl](#) na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.  
import { S3Client } from "@aws-sdk/client-s3";  
// Set the AWS Region.  
const REGION = "us-east-1";  
// Create an Amazon S3 service client object.  
const s3Client = new S3Client({ region: REGION });  
export { s3Client };
```

Obtenha as permissões de ACL.

```
// Import required AWS SDK clients and commands for Node.js.  
import { GetBucketAclCommand } from "@aws-sdk/client-s3";  
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an  
Amazon S3 service client module.  
  
// Create the parameters.  
export const bucketParams = { Bucket: "BUCKET_NAME" };  
  
export const run = async () => {  
    try {  
        const data = await s3Client.send(new GetBucketAclCommand(bucketParams));  
        console.log("Success", data.Grants);  
        return data; // For unit tests.  
    } catch (err) {  
        console.log("Error", err);  
    }  
};  
run();
```

- Para obter mais informações, consulte o [AWS SDK for JavaScript Guia do desenvolvedor](#).
- Para obter detalhes da API, consulte [GetBucketAcl](#) na Referência da API do AWS SDK for JavaScript.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class BucketWrapper:  
    """Encapsulates S3 bucket actions."""  
    def __init__(self, bucket):  
        """  
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in  
        Boto3  
                    that wraps bucket actions in a class-like structure.  
        """  
        self.bucket = bucket  
        self.name = bucket.name  
  
    def get_acl(self):  
        """  
        Get the ACL of the bucket.  
        :return: The ACL of the bucket.
```

```
"""
try:
    acl = self.bucket.Acl()
    logger.info(
        "Got ACL for bucket %s. Owner is %s.", self.bucket.name, acl.owner)
except ClientError:
    logger.exception("Couldn't get ACL for bucket %s.", self.bucket.name)
    raise
else:
    return acl
```

- Para obter detalhes da API, consulte [GetBucketAcl](#) na Referência da API do AWS SDK para Python (Boto3).

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Obter a ACL de um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como obter a lista de controle de acesso (ACL) de um objeto do S3.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
bool AwsDoc::S3::GetBucketAcl(const Aws::String& bucketName, const Aws::String&
region)
{
    Aws::Client::ClientConfiguration config;
    config.region = region;

    Aws::S3::S3Client s3_client(config);

    Aws::S3::Model::GetBucketAclRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::GetBucketAclOutcome outcome =
        s3_client.GetBucketAcl(request);

    if (outcome.IsSuccess())
    {
        Aws::Vector<Aws::S3::Model::Grant> grants =
            outcome.GetResult().GetGrants();

        for (auto it = grants.begin(); it != grants.end(); it++)
        {
            Aws::S3::Model::Grant grant = *it;
            Aws::S3::Model::Grantee grantee = grant.GetGrantee();

            std::cout << "For bucket " << bucketName << ":" "
                << std::endl << std::endl;
```

```
if (grantee.TypeHasBeenSet())
{
    std::cout << "Type:           "
        << GetGranteeTypeString(grantee.GetType()) << std::endl;
}

if (grantee.DisplayNameHasBeenSet())
{
    std::cout << "Display name:  "
        << grantee.GetDisplayName() << std::endl;
}

if (grantee.EmailAddressHasBeenSet())
{
    std::cout << "Email address: "
        << grantee.GetEmailAddress() << std::endl;
}

if (grantee.IDHasBeenSet())
{
    std::cout << "ID:           "
        << grantee.GetID() << std::endl;
}

if (grantee.URIHasBeenSet())
{
    std::cout << "URI:           "
        << grantee.GetURI() << std::endl;
}

std::cout << "Permission:    " <<
    GetPermissionString("bucket", grant.GetPermission()) <<
    std::endl << std::endl;
}

else
{
    auto err = outcome.GetError();
    std::cout << "Error: GetBucketAcl: "
        << err.GetExceptionName() << ":" << err.GetMessage() << std::endl;

    return false;
}

return true;
}
```

- Para obter detalhes da API, consulte [GetObjectAcl](#) na Referência da API do AWS SDK for C++.

Go

SDK for Go V2

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
package main

import (
    "context"
    "flag"
```

```
"fmt"

"github.com/aws/aws-sdk-go-v2/config"
"github.com/aws/aws-sdk-go-v2/service/s3"
)

// S3GetObjectAclAPI defines the interface for the GetObjectAcl function.
// We use this interface to test the function using a mocked service.
type S3GetObjectAclAPI interface {
    GetObjectAcl(ctx context.Context,
        params *s3.GetObjectAclInput,
        optFns ...func(*s3.Options)) (*s3.GetObjectAclOutput, error)
}

// FindObjectAcl gets the access control list (ACL) for an Amazon Simple Storage
// Service (Amazon S3) bucket object
// Inputs:
//     c is the context of the method call, which includes the AWS Region
//     api is the interface that defines the method call
//     input defines the input arguments to the service call.
// Output:
//     If success, a GetObjectAclOutput object containing the result of the service
//     call and nil
//     Otherwise, nil and an error from the call to GetObjectAcl
func FindObjectAcl(c context.Context, api S3GetObjectAclAPI, input
    *s3.GetObjectAclInput) (*s3.GetObjectAclOutput, error) {
    return api.GetObjectAcl(c, input)
}

func main() {
    bucket := flag.String("b", "", "The bucket containing the object")
    objectName := flag.String("o", "", "The bucket object to get ACL from")
    flag.Parse()

    if *bucket == "" || *objectName == "" {
        fmt.Println("You must supply a bucket (-b BUCKET) and object (-o OBJECT)")
        return
    }

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        panic("configuration error, " + err.Error())
    }

    client := s3.NewFromConfig(cfg)

    input := &s3.GetObjectAclInput{
        Bucket: bucket,
        Key:    objectName,
    }

    result, err := FindObjectAcl(context.TODO(), client, input)
    if err != nil {
        fmt.Println("Got an error getting ACL for " + *objectName)
        return
    }

    fmt.Println("Owner:", *result.Owner.DisplayName)
    fmt.Println("")
    fmt.Println("Grants")

    for _, g := range result.Grants {
        fmt.Println("  Grantee:  ", *g.Grantee.DisplayName)
        fmt.Println("  Type:      ", string(g.Grantee.Type))
        fmt.Println("  Permission:", string(g.Permission))
        fmt.Println("")
    }
}
```

```
}
```

- Para obter detalhes da API, consulte [GetObjectAcl](#) na Referência da API do AWS SDK for Go.

Kotlin

SDK para Kotlin

Note

Essa documentação é de pré-lançamento para um recurso em versão de pré-visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
suspend fun getBucketACL(objectKey: String, bucketName: String) {  
  
    val request = GetObjectAclRequest {  
        bucket = bucketName  
        key = objectKey  
    }  
  
    S3Client { region = "us-east-1" }.use { s3 ->  
        val response = s3.getObjectAcl(request)  
        response.grants?.forEach { grant ->  
            println("Grant permission is ${grant.permission}")  
        }  
    }  
}
```

- Para obter detalhes da API, consulte [GetObjectAcl](#) na Referência da API do AWS SDK para Kotlin.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class ObjectWrapper:  
    """Encapsulates S3 object actions."""  
    def __init__(self, s3_object):  
        """  
        :param s3_object: A Boto3 Object resource. This is a high-level resource in  
        Boto3  
                    that wraps object actions in a class-like structure.  
        """  
        self.object = s3_object  
        self.key = self.object.key  
  
    def get_acl(self):  
        """  
        Gets the ACL of the object.  
        """
```

```
:return: The ACL of the object.  
"""  
try:  
    acl = self.object.Acl()  
    logger.info(  
        "Got ACL for object %s owned by %s.",  
        self.object.key, acl.owner['DisplayName'])  
except ClientError:  
    logger.exception("Couldn't get ACL for object %s.", self.object.key)  
else:  
    return acl
```

- Para obter detalhes da API, consulte [GetObjectAcl](#) na Referência da API do AWS SDK para Python (Boto3).

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Obter a região em que reside o bucket do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como obter o local da região de um bucket do S3.

Rust

SDK for Rust

Note

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
async fn show_buckets(strict: bool, client: &Client, region: &str) -> Result<(),  
Error> {  
    let resp = client.list_buckets().send().await?;  
    let buckets = resp.buckets().unwrap_or_default();  
    let num_buckets = buckets.len();  
  
    let mut in_region = 0;  
  
    for bucket in buckets {  
        if strict {  
            let r = client  
                .get_bucket_location()  
                .bucket(bucket.name().unwrap_or_default())  
                .send()  
                .await?;  
  
            if r.location_constraint().unwrap().as_ref() == region {  
                println!("{} {}", bucket.name().unwrap_or_default());  
                in_region += 1;  
            }  
        } else {  
    }
```

```
        println!("{}", bucket.name().unwrap_or_default());
    }

    println!();
    if strict {
        println!(
            "Found {} buckets in the {} region out of a total of {} buckets.",
            in_region, region, num_buckets
        );
    } else {
        println!("Found {} buckets in all regions.", num_buckets);
    }

    Ok(())
}
```

- Para obter detalhes da API, consulte [GetBucketLocation](#) na Referência da API do AWS SDK para Rust.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Obter a configuração de ciclo de vida de um bucket do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como obter a configuração de ciclo de vida de um bucket do S3.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""
    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                    that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def get_lifecycle_configuration(self):
        """
        Get the lifecycle configuration of the bucket.

        :return: The lifecycle rules of the specified bucket.
        """
        try:
            config = self.bucket.LifecycleConfiguration()
            logger.info(
                "Got lifecycle rules %s for bucket '%s'.", config.rules,
                self.bucket.name)
        except ClientError as error:
            logger.error("Error getting lifecycle rules for bucket %s: %s",
                        self.bucket.name, error)

    def set_lifecycle_configuration(self, rules):
        """
        Set the lifecycle configuration of the bucket.

        :param rules: The lifecycle rules to apply to the bucket.
        """
        self.bucket.LifecycleConfiguration(config=rules)
```

```
        except:
            logger.exception(
                "Couldn't get lifecycle rules for bucket '%s'.", self.bucket.name)
            raise
    else:
        return config.rules
```

- Para obter detalhes da API, consulte [GetBucketLifecycleConfiguration](#) na Referência da API do AWS SDK para Python (Boto3).

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Obter a política de um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como obter a política de um bucket do S3.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
bool AwsDoc::S3::GetBucketPolicy(const Aws::String& bucketName,
                                  const Aws::String& region)
{
    Aws::Client::ClientConfiguration config;
    config.region = region;

    Aws::S3::S3Client s3_client(config);

    Aws::S3::Model::GetBucketPolicyRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::GetBucketPolicyOutcome outcome =
        s3_client.GetBucketPolicy(request);

    if (outcome.IsSuccess())
    {
        Aws::StringStream policy_stream;
        Aws::String line;

        outcome.GetResult().GetPolicy() >> line;
        policy_stream << line;

        std::cout << "Policy:" << std::endl << std::endl <<
        policy_stream.str() << std::endl;

        return true;
    }
    else
    {
        auto err = outcome.GetError();
        std::cout << "Error: GetBucketPolicy: "

```

```
        << err.GetExceptionName() << ":" << err.GetMessage() << std::endl;
    return false;
}

int main()
{
    //TODO: Change bucket_name to the name of a bucket in your account.
    const Aws::String bucket_name = "<Enter bucket name>";
    //TODO: Set to the AWS Region in which the bucket was created.
    const Aws::String region = "us-east-1";

    Aws::SDKOptions options;
    Aws::InitAPI(options);
    {
        if (!AwsDoc::S3::GetBucketPolicy(bucket_name, region))
        {
            return 1;
        }
    }
    Aws::ShutdownAPI(options);

    return 0;
}
```

- Para obter detalhes da API, consulte [GetBucketPolicy](#) na Referência da API do AWS SDK for C++.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static String getPolicy(S3Client s3, String bucketName) {

    String policyText;
    System.out.format("Getting policy for bucket: \"%s\"\n\n", bucketName);
    GetBucketPolicyRequest policyReq = GetBucketPolicyRequest.builder()
        .bucket(bucketName)
        .build();

    try {
        GetBucketPolicyResponse policyRes = s3.getBucketPolicy(policyReq);
        policyText = policyRes.policy();
        return policyText;
    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }

    return "";
}
```

- Para obter detalhes da API, consulte [GetBucketPolicy](#) na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.  
import { S3Client } from "@aws-sdk/client-s3";  
// Set the AWS Region.  
const REGION = "us-east-1";  
// Create an Amazon S3 service client object.  
const s3Client = new S3Client({ region: REGION });  
export { s3Client };
```

Obtenha a política de bucket.

```
// Import required AWS SDK clients and commands for Node.js.  
import { GetBucketPolicyCommand } from "@aws-sdk/client-s3";  
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an  
Amazon S3 service client module.  
  
// Create the parameters for calling  
export const bucketParams = { Bucket: "BUCKET_NAME" };  
  
export const run = async () => {  
    try {  
        const data = await s3Client.send(new GetBucketPolicyCommand(bucketParams));  
        console.log("Success", data);  
        return data; // For unit tests.  
    } catch (err) {  
        console.log("Error", err);  
    }  
};  
run();
```

- Para obter mais informações, consulte o [AWS SDK for JavaScript Guia do desenvolvedor](#).
- Para obter detalhes da API, consulte [GetBucketPolicy](#) na Referência da API do AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

Note

Essa documentação é de pré-lançamento para um recurso em versão de pré-visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
suspend fun getPolicy(bucketName: String): String? {  
    println("Getting policy for bucket $bucketName")
```

```
val request = GetBucketPolicyRequest {  
    bucket = bucketName  
}  
  
S3Client { region = "us-east-1" }.use { s3 ->  
    val policyRes = s3.getBucketPolicy(request)  
    return policyRes.policy  
}  
}
```

- Para obter detalhes da API, consulte [GetBucketPolicy](#) na Referência da API do AWS SDK para Kotlin.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class BucketWrapper:  
    """Encapsulates S3 bucket actions."""  
    def __init__(self, bucket):  
        """  
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in  
        Boto3  
                    that wraps bucket actions in a class-like structure.  
        """  
        self.bucket = bucket  
        self.name = bucket.name  
  
    def get_policy(self):  
        """  
        Get the security policy of the bucket.  
  
        :return: The security policy of the specified bucket, in JSON format.  
        """  
        try:  
            policy = self.bucket.Policy()  
            logger.info("Got policy %s for bucket '%s'.", policy.policy,  
self.bucket.name)  
        except ClientError:  
            logger.exception("Couldn't get policy for bucket '%s'.",  
self.bucket.name)  
            raise  
        else:  
            return json.loads(policy.policy)
```

- Para obter detalhes da API, consulte [GetBucketPolicy](#) na Referência da API do AWS SDK para Python (Boto3).

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
# Wraps an Amazon S3 bucket policy.
class BucketPolicyWrapper
  attr_reader :bucket_policy

  # @param bucket_policy [Aws::S3::BucketPolicy] A bucket policy object configured
  # with an existing bucket.
  def initialize(bucket_policy)
    @bucket_policy = bucket_policy
  end

  # Gets the policy of a bucket.
  #
  # @return [Aws::S3::GetBucketPolicyOutput, nil] The current bucket policy.
  def get_policy
    policy = @bucket_policy.data.policy
    policy.respond_to?(:read) ? policy.read : policy
    rescue Aws::Errors::ServiceError => e
      puts "Couldn't get the policy for #{@bucket_policy.bucket.name}. Here's why:
#{e.message}"
      nil
    end
  end
end
```

- Para obter detalhes da API, consulte [GetBucketPolicy](#) na Referência da API do AWS SDK for Ruby.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Obter a configuração de site de um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como obter a configuração de site de um bucket do S3.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
bool AwsDoc::S3::GetWebsiteConfig(const Aws::String& bucketName,
                                    const Aws::String& region)
{
    Aws::Client::ClientConfiguration config;

    if (!region.empty())
    {
        config.region = region;
    }

    Aws::S3::S3Client s3_client(config);

    Aws::S3::Model::GetBucketWebsiteRequest request;
    request.SetBucket(bucketName);
```

```
Aws::S3::Model::GetBucketWebsiteOutcome outcome =
    s3_client.GetBucketWebsite(request);

if (outcome.IsSuccess())
{
    Aws::S3::Model::GetBucketWebsiteResult result = outcome.GetResult();

    std::cout << "Success: GetBucketWebsite: "
        << std::endl << std::endl
        << "For bucket '" << bucketName << "' :"
        << std::endl
        << "Index page :"
        << result.GetIndexDocument().GetSuffix()
        << std::endl
        << "Error page: "
        << result.GetErrorDocument().GetKey()
        << std::endl;

    return true;
}
else
{
    auto err = outcome.GetError();

    std::cout << "Error: GetBucketWebsite: "
        << err.GetMessage() << std::endl;

    return false;
}

int main()
{
    Aws::SDKOptions options;
    Aws::InitAPI(options);
    {
        //TODO: Change bucket_name to the name of a bucket in your account.
        const Aws::String bucket_name = "DOC-EXAMPLE-BUCKET";
        //TODO: Set to the AWS Region in which the bucket was created.
        const Aws::String region = "us-east-1";

        if (!AwsDoc::S3::GetWebsiteConfig(bucket_name, region))
        {
            return 1;
        }
    }
    Aws::ShutdownAPI(options);

    return 0;
}
```

- Para obter detalhes da API, consulte [GetBucketWebsite](#) na Referência da APIs do AWS SDK for C++.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.  
import { S3Client } from "@aws-sdk/client-s3";  
// Set the AWS Region.  
const REGION = "us-east-1";  
// Create an Amazon S3 service client object.  
const s3Client = new S3Client({ region: REGION });  
export { s3Client };
```

Obtenha a configuração do site.

```
// Import required AWS SDK clients and commands for Node.js.  
import { GetBucketWebsiteCommand } from "@aws-sdk/client-s3";  
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an  
Amazon S3 service client module.  
  
// Create the parameters for calling  
export const bucketParams = { Bucket: "BUCKET_NAME" };  
  
export const run = async () => {  
    try {  
        const data = await s3Client.send(new GetBucketWebsiteCommand(bucketParams));  
        console.log("Success", data);  
        return data; // For unit tests.  
    } catch (err) {  
        console.log("Error", err);  
    }  
};  
run();
```

- Para obter detalhes da API, consulte [GetBucketWebsite](#) na Referência da APIs do AWS SDK for JavaScript.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Listar buckets do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como listar buckets do S3.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
bool AwsDoc::S3::ListBuckets() {  
    Aws::S3::S3Client client;  
  
    auto outcome = client.ListBuckets();  
    if (outcome.IsSuccess()) {  
        std::cout << "Found " << outcome.GetResult().GetBuckets().size() << "  
buckets\n";
```

```
        for (auto&& b : outcome.GetResult().GetBuckets()) {
            std::cout << b.GetName() << std::endl;
        }
        return true;
    }
    else {
        std::cout << "Failed with error: " << outcome.GetError() << std::endl;
        return false;
    }
}

int main()
{
    //The Aws::SDKOptions struct contains SDK configuration options.
    //An instance of Aws::SDKOptions is passed to the Aws::InitAPI and
    //Aws::ShutdownAPI methods. The same instance should be sent to both methods.
    Aws::SDKOptions options;
    options.loggingOptions.logLevel = Aws::Utils::Logging::LogLevel::Debug;

    //The AWS SDK for C++ must be initialized by calling Aws::InitAPI.
    InitAPI(options);

    AwsDoc::S3::ListBuckets();

    //Before the application terminates, the SDK must be shut down.
    ShutdownAPI(options);
    return 0;
}
```

- Para obter detalhes da API, consulte [ListBuckets](#) na Referência da API do AWS SDK for C++.

Go

SDK for Go V2

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
listBucketsResult, err := client.ListBuckets(context.TODO(),
&s3.ListBucketsInput{})

if err != nil {
    panic("Couldn't list buckets")
}

for _, bucket := range listBucketsResult.Buckets {
    fmt.Printf("Bucket name: %s\t\tcreated at: %v\n", *bucket.Name,
bucket.CreationDate)
}
```

- Para obter detalhes da API, consulte [ListBuckets](#) na Referência da API do AWS SDK for Go.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.  
import { S3Client } from "@aws-sdk/client-s3";  
// Set the AWS Region.  
const REGION = "us-east-1";  
// Create an Amazon S3 service client object.  
const s3Client = new S3Client({ region: REGION });  
export { s3Client };
```

Liste os buckets.

```
// Import required AWS SDK clients and commands for Node.js.  
import { ListBucketsCommand } from "@aws-sdk/client-s3";  
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an  
Amazon S3 service client module.  
  
export const run = async () => {  
    try {  
        const data = await s3Client.send(new ListBucketsCommand({}));  
        console.log("Success", data.Buckets);  
        return data; // For unit tests.  
    } catch (err) {  
        console.log("Error", err);  
    }  
};  
run();
```

- Para obter mais informações, consulte o [AWS SDK for JavaScript Guia do desenvolvedor](#).
- Para obter detalhes da API, consulte [ListBuckets](#) na Referência da API do AWS SDK for JavaScript.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class BucketWrapper:  
    """Encapsulates S3 bucket actions."""  
    def __init__(self, bucket):  
        """  
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in  
        Boto3  
                    that wraps bucket actions in a class-like structure.  
        """  
        self.bucket = bucket  
        self.name = bucket.name  
  
    @staticmethod  
    def list(s3_resource):  
        """  
        Get the buckets in all Regions for the current account.  
        :param s3_resource: A Boto3 S3 resource. This is a high-level resource in  
        Boto3
```

```
        that contains collections and factory methods to create
        other high-level S3 sub-resources.
    :return: The list of buckets.
    """
    try:
        buckets = list(s3_resource.buckets.all())
        logger.info("Got buckets: %s.", buckets)
    except ClientError:
        logger.exception("Couldn't get buckets.")
        raise
    else:
        return buckets
```

- Para obter detalhes da API, consulte [ListBuckets](#) na Referência da API do AWS SDK para Python (Boto3).

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
require "aws-sdk-s3"

# Wraps Amazon S3 resource actions.
class BucketListWrapper
  attr_reader :s3_resource

  # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
  def initialize(s3_resource)
    @s3_resource = s3_resource
  end

  # Lists buckets for the current account.
  #
  # @param count [Integer] The maximum number of buckets to list.
  def list_buckets(count)
    puts "Found these buckets:"
    @s3_resource.buckets.each do |bucket|
      puts "\t#{bucket.name}"
      count -= 1
      break if count.zero?
    end
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't list buckets. Here's why: #{e.message}"
    false
  end
end

def run_demo
  wrapper = BucketListWrapper.new(Aws::S3::Resource.new)
  wrapper.list_buckets(25)
end

run_demo if $PROGRAM_NAME == __FILE__
```

- Para obter detalhes da API, consulte [ListBuckets](#) na Referência da API do AWS SDK for Ruby.

Rust

SDK for Rust

Note

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
async fn show_buckets(strict: bool, client: &Client, region: &str) -> Result<(), Error> {
    let resp = client.list_buckets().send().await?;
    let buckets = resp.buckets().unwrap_or_default();
    let num_buckets = buckets.len();

    let mut in_region = 0;

    for bucket in buckets {
        if strict {
            let r = client
                .get_bucket_location()
                .bucket(bucket.name().unwrap_or_default())
                .send()
                .await?;

            if r.location_constraint().unwrap().as_ref() == region {
                println!("{}", bucket.name().unwrap_or_default());
                in_region += 1;
            }
        } else {
            println!("{}", bucket.name().unwrap_or_default());
        }
    }

    println!();
    if strict {
        println!(
            "Found {} buckets in the {} region out of a total of {} buckets.",
            in_region, region, num_buckets
        );
    } else {
        println!("Found {} buckets in all regions.", num_buckets);
    }

    Ok(())
}
```

- Para obter detalhes da API, consulte [ListBuckets](#) na Referência da API do AWS SDK para Rust.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Listar carregamentos fracionados em andamento para um bucket do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como listar carregamentos fracionados em andamento em um bucket do S3.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static void listUploads( S3Client s3, String bucketName ) {  
  
    try {  
        ListMultipartUploadsRequest listMultipartUploadsRequest =  
ListMultipartUploadsRequest.builder()  
            .bucket(bucketName)  
            .build();  
  
        ListMultipartUploadsResponse response =  
s3.listMultipartUploads(listMultipartUploadsRequest);  
        List<MultipartUpload> uploads = response.uploads();  
        for (MultipartUpload upload: uploads) {  
            System.out.println("Upload in progress: Key = " + upload.key() +  
"\\", id = " + upload.uploadId());  
        }  
  
    } catch (S3Exception e) {  
        System.err.println(e.getMessage());  
        System.exit(1);  
    }  
}
```

- Para obter detalhes da API, consulte [ListMultipartUploads](#) na Referência da API do AWS SDK for Java 2.x.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Listar a versão de objetos em um bucket do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como listar versões de objeto em um bucket do S3.

Rust

SDK for Rust

Note

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
async fn show_versions(client: &Client, bucket: &str) -> Result<(), Error> {
    let resp = client.list_object_versions().bucket(bucket).send().await?;

    for version in resp.versions().unwrap_or_default() {
        println!("{} ", version.key().unwrap_or_default());
        println!(" version ID: {}", version.version_id().unwrap_or_default());
        println!();
    }

    Ok(())
}
```

- Para obter detalhes da API, consulte [ListObjectVersions](#) na Referência da API do AWS SDK para Rust.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Listar objetos em um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como listar objetos em um bucket do S3.

.NET

AWS SDK for .NET

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
/// <summary>
/// Shows how to list the objects in an Amazon S3 bucket.
/// </summary>
/// <param name="client">An initialized Amazon S3 client object.</param>
/// <param name="bucketName">The name of the bucket for which to list
/// the contents.</param>
/// <returns>A boolean value indicating the success or failure of the
/// copy operation.</returns>
public static async Task<bool> ListBucketContentsAsync(IAmazonS3 client,
string bucketName)
{
    try
    {
        var request = new ListObjectsV2Request
        {
            BucketName = bucketName,
            MaxKeys = 5,
        };

        Console.WriteLine("-----");
        Console.WriteLine($"Listing the contents of {bucketName}:");
```

```
Console.WriteLine("-----");
var response = new ListObjectsV2Response();

do
{
    response = await client.ListObjectsV2Async(request);

    response.S3Objects
        .ForEach(obj => Console.WriteLine($"{obj.Key,-35}{obj.LastModified.ToShortDateString(),10}{obj.Size,10}"));

    // If the response is truncated, set the request
ContinuationToken
    // from the NextContinuationToken property of the response.
    request.ContinuationToken = response.NextContinuationToken;
}
while (response.IsTruncated);

return true;
}
catch (AmazonS3Exception ex)
{
    Console.WriteLine($"Error encountered on server.
Message:{ex.Message}' getting list of objects.");
    return false;
}
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK for .NET.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
bool AwsDoc::S3::ListObjects(const Aws::String& bucketName,
    const Aws::String& region)
{
    Aws::Client::ClientConfiguration config;

    if (!region.empty())
    {
        config.region = region;
    }

    Aws::S3::S3Client s3_client(config);

    Aws::S3::Model::ListObjectsRequest request;
    request.WithBucket(bucketName);

    auto outcome = s3_client.ListObjects(request);

    if (outcome.IsSuccess())
    {
        std::cout << "Objects in bucket '" << bucketName << "':
        << std::endl << std::endl;
```

```
Aws::Vector<Aws::S3::Model::Object> objects =
    outcome.GetResult().GetContents();

for (Aws::S3::Model::Object& object : objects)
{
    std::cout << object.GetKey() << std::endl;
}

return true;
}
else
{
    std::cout << "Error: ListObjects: " <<
        outcome.GetError().GetMessage() << std::endl;

    return false;
}

int main()
{
    Aws::SDKOptions options;
    Aws::InitAPI(options);
    {
        //TODO: Name of a bucket in your account.
        //The bucket must have at least one object in it. One way to achieve
        //this is to configure and run put_object.cpp's executable first.
        const Aws::String bucket_name = "DOC-EXAMPLE-BUCKET";

        //TODO: Set to the AWS Region in which the bucket was created.
        Aws::String region = "us-east-1";

        if (!AwsDoc::S3::ListObjects(bucket_name, region))
        {
            return 1;
        }

    }
    Aws::ShutdownAPI(options);

    return 0;
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK for C++.

Go

SDK for Go V2

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
// List objects in the bucket.
// n.b. object keys in Amazon S3 do not begin with '/'. You do not need to lead
// your
// prefix with it.
fmt.Println("Listing the objects in the bucket:")
listObjsResponse, err := client.ListObjectsV2(context.TODO(),
&s3.ListObjectsV2Input{
    Bucket: aws.String(name),
    Prefix: aws.String("")},
```

```
    })  
  
    if err != nil {  
        panic("Couldn't list bucket contents")  
    }  
  
    for _, object := range listObjsResponse.Contents {  
        fmt.Printf("%s (%d bytes, class %v) \n", *object.Key, object.Size,  
        object.StorageClass)  
    }  
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK for Go.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static void listBucketObjects(S3Client s3, String bucketName ) {  
  
    try {  
        ListObjectsRequest listObjects = ListObjectsRequest  
            .builder()  
            .bucket(bucketName)  
            .build();  
  
        ListObjectsResponse res = s3.listObjects(listObjects);  
        List<S3Object> objects = res.contents();  
        for (S3Object myValue : objects) {  
            System.out.print("\n The name of the key is " + myValue.key());  
            System.out.print("\n The object is " + calKb(myValue.size()) + "  
KBs");  
            System.out.print("\n The owner is " + myValue.owner());  
        }  
  
    } catch (S3Exception e) {  
        System.err.println(e.awsErrorDetails().errorMessage());  
        System.exit(1);  
    }  
  
    //convert bytes to kbs.  
    private static long calKb(Long val) {  
        return val/1024;  
    }  
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.  
import { S3Client } from "@aws-sdk/client-s3";  
// Set the AWS Region.  
const REGION = "us-east-1";  
// Create an Amazon S3 service client object.  
const s3Client = new S3Client({ region: REGION });  
export { s3Client };
```

Liste os objetos.

```
// Import required AWS SDK clients and commands for Node.js.  
import { ListObjectsCommand } from "@aws-sdk/client-s3";  
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an  
Amazon S3 service client module.  
  
// Create the parameters for the bucket  
export const bucketParams = { Bucket: "BUCKET_NAME" };  
  
export const run = async () => {  
    try {  
        const data = await s3Client.send(new ListObjectsCommand(bucketParams));  
        console.log("Success", data);  
        return data; // For unit tests.  
    } catch (err) {  
        console.log("Error", err);  
    }  
};  
run();
```

Liste mil ou mais objetos.

```
// Import required AWS SDK clients and commands for Node.js.  
import { ListObjectsCommand } from "@aws-sdk/client-s3";  
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an  
Amazon S3 service client module.  
  
// Create the parameters for the bucket  
export const bucketParams = { Bucket: "BUCKET_NAME" };  
  
export async function run() {  
    // Declare truncated as a flag that the while loop is based on.  
    let truncated = true;  
    // Declare a variable to which the key of the last element is assigned to in the  
    response.  
    let pageMarker;  
    // while loop that runs until 'response.truncated' is false.  
    while (truncated) {  
        try {  
            const response = await s3Client.send(new ListObjectsCommand(bucketParams));  
            // return response; //For unit tests  
            response.Contents.forEach((item) => {  
                console.log(item.Key);  
            });  
            // Log the key of every item in the response to standard output.  
            truncated = response.IsTruncated;  
            // If truncated is true, assign the key of the last element in the response  
            to the pageMarker variable.  
            if (truncated) {  
                pageMarker = response.Contents[response.Contents.length - 1].Key;  
            }  
        } catch (err) {  
            console.error("Error listing objects: ", err.message);  
        }  
    }  
}
```

```
    pageMarker = response.Contents.slice(-1)[0].Key;
    // Assign the pageMarker value to bucketParams so that the next iteration
    starts from the new pageMarker.
    bucketParams.Marker = pageMarker;
}
// At end of the list, response.truncated is false, and the function exits
the while loop.
} catch (err) {
    console.log("Error", err);
    truncated = false;
}
}
run();
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

Note

Essa documentação é de pré-lançamento para um recurso em versão de pré-visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
suspend fun listBucketObjects(bucketName: String) {

    val request = ListObjectsRequest {
        bucket = bucketName
    }

    S3Client { region = "us-east-1" }.use { s3 ->

        val response = s3.listObjects(request)
        response.contents?.forEach { myObject ->
            println("The name of the key is ${myObject.key}")
            println("The object is ${calKb(myObject.size)} KBs")
            println("The owner is ${myObject.owner}")
        }
    }
}

private fun calKb(intValue: Long): Long {
    return intValue / 1024
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK para Kotlin.

PHP

SDK for PHP

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Liste objetos em um bucket.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2', 'version' => 'latest']);

try {
    $contents = $s3client->listObjects([
        'Bucket' => $bucket_name,
    ]);
    echo "The contents of your bucket are: \n";
    foreach ($contents['Contents'] as $content) {
        echo $content['Key'] . "\n";
    }
} catch (Exception $exception) {
    echo "Failed to list objects in $bucket_name with error: " . $exception-
>getMessage();
    exit("Please fix error with listing objects before continuing.");
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK for PHP.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""
    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource in
Boto3
                           that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

    @staticmethod
    def list(bucket, prefix=None):
        """
        Lists the objects in a bucket, optionally filtered by a prefix.

        :param bucket: The bucket to query. This is a Boto3 Bucket resource.
        :param prefix: When specified, only objects that start with this prefix are
listed.
        :return: The list of objects.
        """
        try:
            if not prefix:
                objects = list(bucket.objects.all())

```

```
        else:
            objects = list(bucket.objects.filter(Prefix=prefix))
            logger.info("Got objects %s from bucket '%s'",
                        [o.key for o in objects], bucket.name)
        except ClientError:
            logger.exception("Couldn't get objects for bucket '%s'.", bucket.name)
            raise
    else:
        return objects
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK para Python (Boto3).

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
require "aws-sdk-s3"

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

  # @param bucket [Aws::S3::Bucket] An existing Amazon S3 bucket.
  def initialize(bucket)
    @bucket = bucket
  end

  # Lists object in a bucket.
  #
  # @param max_objects [Integer] The maximum number of objects to list.
  # @return [Integer] The number of objects listed.
  def list_objects(max_objects)
    count = 0
    puts "The objects in #{@bucket.name} are:"
    @bucket.objects.each do |obj|
      puts "\t#{obj.key}"
      count += 1
      break if count == max_objects
    end
    count
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't list objects in bucket #{@bucket.name}. Here's why: #{e.message}"
    0
  end

  def run_demo
    bucket_name = "doc-example-bucket"

    wrapper = BucketListObjectsWrapper.new(Aws::S3::Bucket.new(bucket_name))
    count = wrapper.list_objects(25)
    puts "Listed #{count} objects."
  end

run_demo if $PROGRAM_NAME == __FILE__
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK for Ruby.

Rust

SDK for Rust

Note

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
pub async fn list_objects(client: &Client, bucket_name: &str) -> Result<(), Error> {
    let objects = client.list_objects_v2().bucket(bucket_name).send().await?;
    println!("Objects in bucket:");
    for obj in objects.contents().unwrap_or_default() {
        println!("{}:{}?", obj.key().unwrap());
    }
    Ok(())
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK para Rust.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public func listBucketFiles(bucket: String) async throws -> [String] {
    let input = ListObjectsV2Input(
        bucket: bucket
    )
    let output = try await client.listObjectsV2(input: input)
    var names: [String] = []

    guard let objList = output.contents else {
        return []
    }

    for obj in objList {
        if let objName = obj.key {
            names.append(objName)
        }
    }
}
```

```
    return names
}
```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API do AWS SDK para Swift.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Restaurar uma cópia arquivada de um objeto em um bucket do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como restaurar uma cópia arquivada de um objeto em um bucket do S3.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static void restoreS3Object(S3Client s3, String bucketName, String
keyName, String expectedBucketOwner) {

    try {
        RestoreRequest restoreRequest = RestoreRequest.builder()
            .days(10)

        .glacierJobParameters(GlacierJobParameters.builder().tier(Tier.STANDARD).build())
            .build();

        RestoreObjectRequest objectRequest = RestoreObjectRequest.builder()
            .expectedBucketOwner(expectedBucketOwner)
            .bucket(bucketName)
            .key(keyName)
            .restoreRequest(restoreRequest)
            .build();

        s3.restoreObject(objectRequest);

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Para obter detalhes da API, consulte [RestoreObject](#) em Referência da API do AWS SDK for Java 2.x.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Definir uma nova ACL para um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como definir uma nova lista de controle de acesso (ACL) para um bucket do S3.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static void setBucketAcl(S3Client s3, String bucketName, String id) {  
  
    try {  
        Grant ownerGrant = Grant.builder()  
            .grantee(builder -> builder.id(id))  
            .type(Type.CANONICAL_USER)  
            .permission(Permission.FULL_CONTROL)  
            .build();  
  
        List<Grant> grantList2 = new ArrayList<>();  
        grantList2.add(ownerGrant);  
  
        AccessControlPolicy acl = AccessControlPolicy.builder()  
            .owner(builder -> builder.id(id))  
            .grants(grantList2)  
            .build();  
  
        PutBucketAclRequest putAclReq = PutBucketAclRequest.builder()  
            .bucket(bucketName)  
            .accessControlPolicy(acl)  
            .build();  
  
        s3.putBucketAcl(putAclReq);  
  
    } catch (S3Exception e) {  
        e.printStackTrace();  
        System.exit(1);  
    }  
}
```

- Para obter detalhes da API, consulte [PutBucketAcl](#) na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.
```

```
import { S3Client } from "@aws-sdk/client-s3";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an Amazon S3 service client object.
const s3Client = new S3Client({ region: REGION });
export { s3Client };
```

Coloque a ACL do bucket.

```
// Import required AWS SDK clients and commands for Node.js.
import { PutBucketAclCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
Amazon S3 service client module.

// Set the parameters. For more information,
// see https://docs.aws.amazon.com/AWSJavaScriptSDK/latest/AWS/
S3.html#putBucketAcl-property.
export const bucketParams = {
  Bucket: "BUCKET_NAME",
  // 'GrantFullControl' allows grantee the read, write, read ACP, and write ACL
  permissions on the bucket.
  // Use a canonical user ID for an AWS account, formatted as follows:
  // id=002160194XXXXXXXXXXXXXXXXXXXXXXa7a49125274
  GrantFullControl: "GRANTEE_1",
  // 'GrantWrite' allows grantee to create, overwrite, and delete any object in the
  bucket.
  // For example, 'uri=http://acs.amazonaws.com/groups/s3/LogDelivery'
  GrantWrite: "GRANTEE_2",
};

export const run = async () => {
  try {
    const data = await s3Client.send(new PutBucketAclCommand(bucketParams));
    console.log("Success, permissions added to bucket", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- Para obter mais informações, consulte o [AWS SDK for JavaScript Guia do desenvolvedor](#).
- Para obter detalhes da API, consulte [PutBucketAcl](#) na Referência da API do AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

Note

Essa documentação é de pré-lançamento para um recurso em versão de pré-visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
suspend fun setBucketAcl(bucketName: String, idVal: String) {
```

```
val myGrant = Grantee {
    id = idVal
    type = Type.CanonicalUser
}

val ownerGrant = Grant {
    grantee = myGrant
    permission = Permission.FullControl
}

val grantList = mutableListOf<Grant>()
grantList.add(ownerGrant)

val ownerOb = Owner {
    id = idVal
}

val acl = AccessControlPolicy {
    owner = ownerOb
    grants = grantList
}

val request = PutBucketAclRequest {
    bucket = bucketName
    accessControlPolicy = acl
}

S3Client { region = "us-east-1" }.use { s3 ->
    s3.putBucketAcl(request)
    println("An ACL was successfully set on $bucketName")
}
}
```

- Para obter detalhes da API, consulte [PutBucketAcl](#) na Referência da API do AWS SDK para Kotlin.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""
    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                    that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def grant_log_delivery_access(self):
        """
        Grant the AWS Log Delivery group write access to the bucket so that
        Amazon S3 can deliver access logs to the bucket. This is the only
        recommended
        """


```

```
use of an S3 bucket ACL.  
"""  
try:  
    acl = self.bucket.Acl()  
    # Putting an ACL overwrites the existing ACL. If you want to preserve  
    # existing grants, append new grants to the list of existing grants.  
    grants = acl.grants if acl.grants else []  
    grants.append({  
        'Grantee': {  
            'Type': 'Group',  
            'URI': 'http://acs.amazonaws.com/groups/s3/LogDelivery'  
        },  
        'Permission': 'WRITE'  
    })  
    acl.put(  
        AccessControlPolicy={  
            'Grants': grants,  
            'Owner': acl.owner  
        }  
    )  
    logger.info("Granted log delivery access to bucket '%s',  
self.bucket.name)  
except ClientError:  
    logger.exception("Couldn't add ACL to bucket '%s'.", self.bucket.name)  
    raise
```

- Para obter detalhes da API, consulte [PutBucketAcl](#) na Referência da API do AWS SDK para Python (Boto3).

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Definir a ACL de um objeto do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como definir a lista de controle de acesso (ACL) de um objeto do S3.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class ObjectWrapper:  
    """Encapsulates S3 object actions."""  
    def __init__(self, s3_object):  
        """  
        :param s3_object: A Boto3 Object resource. This is a high-level resource in  
        Boto3  
                    that wraps object actions in a class-like structure.  
        """  
        self.object = s3_object  
        self.key = self.object.key  
  
    def put_acl(self, email):
```

```
"""
    Applies an ACL to the object that grants read access to an AWS user
identified
    by email address.

:param email: The email address of the user to grant access.
"""
try:
    acl = self.object.Acl()
    # Putting an ACL overwrites the existing ACL, so append new grants
    # if you want to preserve existing grants.
    grants = acl.grants if acl.grants else []
    grants.append({
        'Grantee': {
            'Type': 'AmazonCustomerByEmail',
            'EmailAddress': email
        },
        'Permission': 'READ'
    })
    acl.put(
        AccessControlPolicy={
            'Grants': grants,
            'Owner': acl.owner
        }
    )
    logger.info("Granted read access to %s.", email)
except ClientError:
    logger.exception("Couldn't add ACL to object '%s'.", self.object.key)
    raise
```

- Para obter detalhes da API, consulte [PutObjectAcl](#) na Referência da API do AWS SDK para Python (Boto3).

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Definir a configuração de site de um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como definir a configuração de site de um bucket do S3.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
bool AwsDoc::S3::PutWebsiteConfig(const Aws::String& bucketName,
                                    const Aws::String& indexPage, const Aws::String& errorPage,
                                    const Aws::String& region)
{
    Aws::Client::ClientConfiguration config;

    if (!region.empty())
    {
```

```
        config.region = region;
    }

    Aws::S3::S3Client s3_client(config);

    Aws::S3::Model::IndexDocument index_doc;
    index_doc.SetSuffix(indexPage);

    Aws::S3::Model::ErrorDocument error_doc;
    error_doc.SetKey(errorPage);

    Aws::S3::Model::WebsiteConfiguration website_config;
    website_config.SetIndexDocument(index_doc);
    website_config.SetErrorDocument(error_doc);

    Aws::S3::Model::PutBucketWebsiteRequest request;
    request.SetBucket(bucketName);
    request.SetWebsiteConfiguration(website_config);

    Aws::S3::Model::PutBucketWebsiteOutcome outcome =
        s3_client.PutBucketWebsite(request);

    if (outcome.IsSuccess())
    {
        std::cout << "Success: Set website configuration for bucket '" 
        << bucketName << "." << std::endl;

        return true;
    }
    else
    {
        std::cout << "Error: PutBucketWebsite: "
        << outcome.GetError().GetMessage() << std::endl;

        return false;
    }

    return 1;
}

int main()
{
    Aws::SDKOptions options;
    Aws::InitAPI(options);
    {

        //TODO: Change bucket_name to the name of a bucket in your account.
        const Aws::String bucket_name = "DOC-EXAMPLE-BUCKET";
        //TODO: Set to the AWS Region in which the bucket was created.
        const Aws::String region = "us-east-1";
        //TODO: Create these two files to serve as your website
        const Aws::String index_page = "index.html";
        const Aws::String error_page = "404.html";


        if (!AwsDoc::S3::PutWebsiteConfig(bucket_name, index_page, error_page,
region))
        {
            return 1;
        }

    }
    Aws::ShutdownAPI(options);

    return 0;
}
```

- Para obter detalhes da API, consulte [PutBucketWebsite](#) na Referência da API do AWS SDK for C++.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static void setWebsiteConfig( S3Client s3, String bucketName, String
indexDoc) {

    try {
        WebsiteConfiguration websiteConfig = WebsiteConfiguration.builder()
            .indexDocument(IndexDocument.builder().suffix(indexDoc).build())
            .build();

        PutBucketWebsiteRequest pubWebsiteReq =
PutBucketWebsiteRequest.builder()
            .bucket(bucketName)
            .websiteConfiguration(websiteConfig)
            .build();

        s3.putBucketWebsite(pubWebsiteReq);
        System.out.println("The call was successful");

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Para obter detalhes da API, consulte [PutBucketWebsite](#) na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.
import { S3Client } from "@aws-sdk/client-s3";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an Amazon S3 service client object.
const s3Client = new S3Client({ region: REGION });
export { s3Client };
```

Defina a configuração do site.

```
// Import required AWS SDK clients and commands for Node.js.
import { PutBucketWebsiteCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
// Amazon S3 service client module.

// Create the parameters for the bucket
export const bucketParams = { Bucket: "BUCKET_NAME" };
export const staticHostParams = {
    Bucket: bucketParams,
    WebsiteConfiguration: {
        ErrorDocument: {
            Key: "",
        },
        IndexDocument: {
            Suffix: "",
        },
    },
};

export const run = async () => {
    // Insert specified bucket name and index and error documents into parameters
    // JSON
    // from command line arguments
    staticHostParams.Bucket = bucketParams;
    staticHostParams.WebsiteConfiguration.IndexDocument.Suffix = "INDEX_PAGE"; // The
    // index document inserted into parameters JSON.
    staticHostParams.WebsiteConfiguration.ErrorDocument.Key = "ERROR_PAGE"; // The
    // error document inserted into parameters JSON.
    // Set the new website configuration on the selected bucket.
    try {
        const data = await s3Client.send(new
PutBucketWebsiteCommand(staticHostParams));
        console.log("Success", data);
        return data;
    } catch (err) {
        console.log("Error", err);
    }
};
run();
```

- Para obter mais informações, consulte o [AWS SDK for JavaScript Guia do desenvolvedor](#).
- Para obter detalhes da API, consulte [PutBucketWebsite](#) na Referência da API do AWS SDK for JavaScript.

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket website actions.
class BucketWebsiteWrapper
    attr_reader :bucket_website

    # @param bucket_website [Aws::S3::BucketWebsite] A bucket website object
    # configured with an existing bucket.
    def initialize(bucket_website)
```

```
    @bucket_website = bucket_website
end

# Sets a bucket as a static website.
#
# @param index_document [String] The name of the index document for the website.
# @param error_document [String] The name of the error document to show for 4XX
errors.
# @return [Boolean] True when the bucket is configured as a website; otherwise,
false.
def set_website(index_document, error_document)
  @bucket_website.put(
    website_configuration: {
      index_document: { suffix: index_document },
      error_document: { key: error_document }
    }
  )
  true
rescue Aws::Errors::ServiceError => e
  puts "Couldn't configure #{@bucket_website.bucket.name} as a website. Here's
why: #{e.message}"
  false
end
end

def run_demo
  bucket_name = "doc-example-bucket"
  index_document = "index.html"
  error_document = "404.html"

  wrapper = BucketWebsiteWrapper.new(Aws::S3::BucketWebsite.new(bucket_name))
  return unless wrapper.set_website(index_document, error_document)

  puts "Successfully configured bucket #{bucket_name} as a static website."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- Para obter detalhes da API, consulte [PutBucketWebsite](#) na Referência da API do AWS SDK for Ruby.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Carregar um objeto em um bucket do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como fazer upload de um objeto em um bucket do S3.

.NET

AWS SDK for .NET

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
/// <summary>
/// Shows how to upload a file from the local computer to an Amazon S3
/// bucket.
/// </summary>
/// <param name="client">An initialized Amazon S3 client object.</param>
/// <param name="bucketName">The Amazon S3 bucket to which the object
/// will be uploaded.</param>
/// <param name="objectName">The object to upload.</param>
/// <param name="filePath">The path, including file name, of the object
/// on the local computer to upload.</param>
/// <returns>A boolean value indicating the success or failure of the
/// upload procedure.</returns>
public static async Task<bool> UploadFileAsync(
    IAmazonS3 client,
    string bucketName,
    string objectName,
    string filePath)
{
    var request = new PutObjectRequest
    {
        BucketName = bucketName,
        Key = objectName,
        FilePath = filePath,
    };

    var response = await client.PutObjectAsync(request);
    if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
    {
        Console.WriteLine($"Successfully uploaded {objectName} to
{bucketName}.");
        return true;
    }
    else
    {
        Console.WriteLine($"Could not upload {objectName} to
{bucketName}.");
        return false;
    }
}
```

- Para obter detalhes da API, consulte [PutObject](#) na Referência da API do AWS SDK for .NET.

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
bool AwsDoc::S3::PutObject(const Aws::String& bucketName,
    const Aws::String& objectName,
    const Aws::String& region)
{
    // Verify that the file exists.
    struct stat buffer;

    if (stat(objectName.c_str(), &buffer) == -1)
    {
        std::cout << "Error: PutObject: File '" <<
            objectName << "' does not exist." << std::endl;
```

```
        return false;
    }

    Aws::Client::ClientConfiguration config;

    if (!region.empty())
    {
        config.region = region;
    }

    Aws::S3::S3Client s3_client(config);

    Aws::S3::Model::PutObjectRequest request;
    request.SetBucket(bucketName);
    //We are using the name of the file as the key for the object in the bucket.
    //However, this is just a string and can set according to your retrieval needs.
    request.SetKey(objectName);

    std::shared_ptr<Aws::IOStream> input_data =
        Aws::MakeShared<Aws::FStream>("SampleAllocationTag",
            objectName.c_str(),
            std::ios_base::in | std::ios_base::binary);

    request.SetBody(input_data);

    Aws::S3::Model::PutObjectOutcome outcome =
        s3_client.PutObject(request);

    if (outcome.IsSuccess()) {

        std::cout << "Added object '" << objectName << "' to bucket ''"
            << bucketName << "'";
        return true;
    }
    else
    {
        std::cout << "Error: PutObject: " <<
            outcome.GetError().GetMessage() << std::endl;

        return false;
    }
}

int main()
{
    Aws::SDKOptions options;
    Aws::InitAPI(options);
    {
        //TODO: Change bucket_name to the name of a bucket in your account.
        const Aws::String bucket_name = "<Enter bucket name>";
        //TODO: Create a file called "my-file.txt" in the local folder where your
        executables are built to.
        const Aws::String object_name = "<Enter file>";
        //TODO: Set to the AWS Region in which the bucket was created.
        const Aws::String region = "us-east-1";

        if (!AwsDoc::S3::PutObject(bucket_name, object_name, region)) {

            return 1;
        }
    }
    Aws::ShutdownAPI(options);

    return 0;
}
```

- Para obter detalhes da API, consulte [PutObject](#) na Referência da API do AWS SDK for C++.

Go

SDK for Go V2

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
// Place an object in a bucket.
fmt.Println("Upload an object to the bucket")
// Get the object body to upload.
// Image credit: https://unsplash.com/photos/iz58d89q3ss
stat, err := os.Stat("image.jpg")
if err != nil {
    panic("Couldn't stat image: " + err.Error())
}
file, err := os.Open("image.jpg")

if err != nil {
    panic("Couldn't open local file")
}

_, err = client.PutObject(context.TODO(), &s3.PutObjectInput{
    Bucket:      aws.String(name),
    Key:         aws.String("path/myfile.jpg"),
    Body:         file,
    ContentLength: stat.Size(),
})

file.Close()

if err != nil {
    panic("Couldn't upload file: " + err.Error())
}
```

- Para obter detalhes da API, consulte [PutObject](#) na Referência da API do AWS SDK for Go.

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Carregue um objeto em um bucket.

```
public static String putS3Object(S3Client s3, String bucketName, String
objectKey, String objectPath) {

    try {
        Map<String, String> metadata = new HashMap<>();
        metadata.put("x-amz-meta-myVal", "test");
        PutObjectRequest putOb = PutObjectRequest.builder()
            .bucket(bucketName)
            .key(objectKey)
```

```
        .metadata(metadata)
        .build();

    PutObjectResponse response = s3.putObject(putOb,
RequestBody.fromBytes(getObjectFile(objectPath)));
    return response.eTag();

} catch (S3Exception e) {
    System.err.println(e.getMessage());
    System.exit(1);
}

return "";
}

// Return a byte array.
private static byte[] getObjectFile(String filePath) {

    FileInputStream fileInputStream = null;
    byte[] bytesArray = null;

    try {
        File file = new File(filePath);
        bytesArray = new byte[(int) file.length()];
        fileInputStream = new FileInputStream(file);
        fileInputStream.read(bytesArray);

    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (fileInputStream != null) {
            try {
                fileInputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

return bytesArray;
}
```

Carregue um objeto em um bucket e defina etiquetas.

```
public static void putS3ObjectTags(S3Client s3, String bucketName, String
objectKey, String objectPath) {

try {

    Tag tag1 = Tag.builder()
        .key("Tag 1")
        .value("This is tag 1")
        .build();

    Tag tag2 = Tag.builder()
        .key("Tag 2")
        .value("This is tag 2")
        .build();

    List<Tag> tags = new ArrayList<>();
    tags.add(tag1);
    tags.add(tag2);

    Tagging allTags = Tagging.builder()
```

```
.tagSet(tags)
.build();

PutObjectRequest putOb = PutObjectRequest.builder()
.bucket(bucketName)
.key(objectKey)
.tagging(allTags)
.build();

s3.putObject(putOb, RequestBody.fromBytes(getObjectFile(objectPath)));

} catch (S3Exception e) {
    System.err.println(e.getMessage());
    System.exit(1);
}
}

public static void updateObjectTags(S3Client s3, String bucketName, String
objectKey) {

    try {
        GetObjectTaggingRequest taggingRequest =
GetObjectTaggingRequest.builder()
        .bucket(bucketName)
        .key(objectKey)
        .build();

        GetObjectTaggingResponse getTaggingRes =
s3.getObjectTagging(taggingRequest);
        List<Tag> obTags = getTaggingRes.tagSet();
        for (Tag sinTag: obTags) {
            System.out.println("The tag key is: "+sinTag.key());
            System.out.println("The tag value is: "+sinTag.value());
        }

        // Replace the object's tags with two new tags.
        Tag tag3 = Tag.builder()
        .key("Tag 3")
        .value("This is tag 3")
        .build();

        Tag tag4 = Tag.builder()
        .key("Tag 4")
        .value("This is tag 4")
        .build();

        List<Tag> tags = new ArrayList<>();
        tags.add(tag3);
        tags.add(tag4);

        Tagging updatedTags = Tagging.builder()
        .tagSet(tags)
        .build();

        PutObjectTaggingRequest taggingRequest1 =
PutObjectTaggingRequest.builder()
        .bucket(bucketName)
        .key(objectKey)
        .tagging(updatedTags)
        .build();

        s3.putObjectTagging(taggingRequest1);
        GetObjectTaggingResponse getTaggingRes2 =
s3.getObjectTagging(taggingRequest);
        List<Tag> modTags = getTaggingRes2.tagSet();
        for (Tag sinTag: modTags) {
```

```
        System.out.println("The tag key is: "+sinTag.key());
        System.out.println("The tag value is: "+sinTag.value());
    }

} catch (S3Exception e) {
    System.err.println(e.getMessage());
    System.exit(1);
}
}
```

Carregue um objeto em um bucket e defina metadados.

```
public static String putS3Object(S3Client s3, String bucketName, String
objectKey, String objectPath) {

    try {
        Map<String, String> metadata = new HashMap<>();
        metadata.put("author", "Mary Doe");
        metadata.put("version", "1.0.0.0");

        PutObjectRequest putOb = PutObjectRequest.builder()
            .bucket(bucketName)
            .key(objectKey)
            .metadata(metadata)
            .build();

        PutObjectResponse response = s3.putObject(putOb,
RequestBody.fromBytes(getObjectFile(objectPath)));
        return response.eTag();

    } catch (S3Exception e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }

    return "";
}

// Return a byte array.
private static byte[] getObjectFile(String filePath) {

    FileInputStream fileInputStream = null;
    byte[] bytesArray = null;

    try {
        File file = new File(filePath);
        bytesArray = new byte[(int) file.length()];
        fileInputStream = new FileInputStream(file);
        fileInputStream.read(bytesArray);

    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (fileInputStream != null) {
            try {
                fileInputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    return bytesArray;
}
```

Carregue um objeto em um bucket e defina um valor de retenção de objetos.

```
public static void setRetentionPeriod(S3Client s3, String key, String bucket) {  
  
    try{  
        LocalDate localDate = LocalDate.parse("2020-07-17");  
        LocalDateTime localDateTime = localDate.atStartOfDay();  
        Instant instant = localDateTime.toInstant(ZoneOffset.UTC);  
  
        ObjectLockRetention lockRetention = ObjectLockRetention.builder()  
            .mode("COMPLIANCE")  
            .retainUntilDate(instant)  
            .build();  
  
        PutObjectRetentionRequest retentionRequest =  
PutObjectRetentionRequest.builder()  
            .bucket(bucket)  
            .key(key)  
            .bypassGovernanceRetention(true)  
            .retention(lockRetention)  
            .build();  
  
        // To set Retention on an object, the Amazon S3 bucket must support  
        // object locking, otherwise an exception is thrown.  
        s3.putObjectRetention(retentionRequest);  
        System.out.print("An object retention configuration was successfully  
placed on the object");  
  
    } catch (S3Exception e) {  
        System.err.println(e.awsErrorDetails().errorMessage());  
        System.exit(1);  
    }  
}
```

- Para obter detalhes da API, consulte [PutObject](#) na Referência da API do AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.  
import { S3Client } from "@aws-sdk/client-s3";  
// Set the AWS Region.  
const REGION = "us-east-1";  
// Create an Amazon S3 service client object.  
const s3Client = new S3Client({ region: REGION });  
export { s3Client };
```

Crie e carregue o objeto.

```
// Import required AWS SDK clients and commands for Node.js.
import { PutObjectCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
// Amazon S3 service client module.

// Set the parameters.
export const bucketParams = {
  Bucket: "BUCKET_NAME",
  // Specify the name of the new object. For example, 'index.html'.
  // To create a directory for the object, use '/'. For example, 'myApp/
  package.json'.
  Key: "OBJECT_NAME",
  // Content of the new object.
  Body: "BODY",
};

// Create and upload the object to the S3 bucket.
export const run = async () => {
  try {
    const data = await s3Client.send(new PutObjectCommand(bucketParams));
    return data; // For unit tests.
    console.log(
      "Successfully uploaded object: " +
      bucketParams.Bucket +
      "/" +
      bucketParams.Key
    );
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Carregue o objeto.

```
// Import required AWS SDK clients and commands for Node.js.
import { PutObjectCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
// Amazon S3 service client module.
import {path} from "path";
import {fs} from "fs";

const file = "OBJECT_PATH_AND_NAME"; // Path to and name of object. For example
'..../myFiles/index.js'.
const fileStream = fs.createReadStream(file);

// Set the parameters
export const uploadParams = {
  Bucket: "BUCKET_NAME",
  // Add the required 'Key' parameter using the 'path' module.
  Key: path.basename(file),
  // Add the required 'Body' parameter
  Body: fileStream,
};

// Upload file to specified bucket.
export const run = async () => {
  try {
    const data = await s3Client.send(new PutObjectCommand(uploadParams));
    console.log("Success", data);
    return data; // For unit tests.
  } catch (err) {
```

```
        console.log("Error", err);
    }
};

run();
```

- Para obter mais informações, consulte o [AWS SDK for JavaScript Guia do desenvolvedor](#).
- Para obter detalhes da API, consulte [PutObject](#) na Referência da API do AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

Note

Essa documentação é de pré-lançamento para um recurso em versão de pré-visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
suspend fun putS3Object(bucketName: String, objectKey: String, objectPath: String)
{
    val metadataVal = mutableMapOf<String, String>()
    metadataVal["myVal"] = "test"

    val request = PutObjectRequest {
        bucket = bucketName
        key = objectKey
        metadata = metadataVal
        body = File(objectPath).asByteStream()
    }

    S3Client { region = "us-east-1" }.use { s3 ->
        val response = s3.putObject(request)
        println("Tag information is ${response.eTag}")
    }
}
```

- Para obter detalhes da API, consulte [PutObject](#) na Referência da API do AWS SDK para Kotlin.

PHP

SDK for PHP

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Carregue um objeto em um bucket.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2', 'version' => 'latest']);

$file_name = "local-file-" . uniqid();
try {
    $s3client->putObject([
        'Bucket' => $bucket_name,
```

```
        'Key' => $file_name,
        'SourceFile' => 'testfile.txt'
    ]);
    echo "Uploaded $file_name to $bucket_name.\n";
} catch (Exception $exception) {
    echo "Failed to upload $file_name with error: " . $exception->getMessage();
    exit("Please fix error with file upload before continuing.");
}
```

- Para obter detalhes da API, consulte [PutObject](#) na Referência da API do AWS SDK for PHP.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""
    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource in
        Boto3
                           that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

    def put(self, data):
        """
        Upload data to the object.

        :param data: The data to upload. This can either be bytes or a string. When
        this
                           argument is a string, it is interpreted as a file name, which
        is
                           opened in read bytes mode.
        """
        put_data = data
        if isinstance(data, str):
            try:
                put_data = open(data, 'rb')
            except IOError:
                logger.exception("Expected file name or binary data, got '%s'.",
data)
                raise

            try:
                self.object.put(Body=put_data)
                self.object.wait_until_exists()
                logger.info(
                    "Put object '%s' to bucket '%s'.", self.object.key,
                    self.object.bucket_name)
            except ClientError:
                logger.exception(
                    "Couldn't put object '%s' to bucket '%s'.", self.object.key,
                    self.object.bucket_name)
                raise
            finally:
                if getattr(put_data, 'close', None):

```

```
put_data.close()
```

- Para obter detalhes da API, consulte [PutObject](#) na Referência da API do AWS SDK para Python (Boto3).

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Carregue um arquivo usando um carregador gerenciado (`Object.upload_file`).

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectUploadFileWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Uploads a file to an Amazon S3 object by using a managed uploader.
  #
  # @param file_path [String] The path to the file to upload.
  # @return [Boolean] True when the file is uploaded; otherwise false.
  def upload_file(file_path)
    @object.upload_file(file_path)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't upload file #{file_path} to #{@object.key}. Here's why: #{e.message}"
    false
  end
end

def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-uploaded-file"
  file_path = "object_upload_file.rb"

  wrapper = ObjectUploadFileWrapper.new(Aws::S3::Object.new(bucket_name,
    object_key))
  return unless wrapper.upload_file(file_path)

  puts "File #{file_path} successfully uploaded to #{bucket_name}:#{object_key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Carregue um arquivo usando `Object.put`.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectPutWrapper
  attr_reader :object
```

```
# @param object [Aws::S3::Object] An existing Amazon S3 object.
def initialize(object)
  @object = object
end

def put_object(source_file_path)
  File.open(source_file_path, "rb") do |file|
    @object.put(body: file)
  end
  true
rescue Aws::Errors::ServiceError => e
  puts "Couldn't put #{source_file_path} to #{object.key}. Here's why:
#{e.message}"
  false
end
end

def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object-key"
  file_path = "my-local-file.txt"

  wrapper = ObjectPutWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  success = wrapper.put_object(file_path)
  return unless success

  puts "Put file #{file_path} into #{object_key} in #{bucket_name}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Carregue um arquivo usando Object.put e adicione criptografia no lado do servidor.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectPutSseWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object_encrypted(object_content, encryption)
    @object.put(body: object_content, server_side_encryption: encryption)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put your content to #{object.key}. Here's why: #{e.message}"
    false
  end
end

def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-encrypted-content"
  object_content = "This is my super-secret content."
  encryption = "AES256"

  wrapper = ObjectPutSseWrapper.new(Aws::S3::Object.new(bucket_name,
  object_content))
  return unless wrapper.put_object_encrypted(object_content, encryption)
```

```
    puts "Put your content into #{bucket_name}:#{object_key} and encrypted it with
#{encryption}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- Para obter detalhes da API, consulte [PutObject](#) na Referência da API do AWS SDK for Ruby.

Rust

SDK for Rust

Note

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
pub async fn upload_object(
    client: &Client,
    bucket_name: &str,
    file_name: &str,
    key: &str,
) -> Result<(), Error> {
    let body = ByteStream::from_path(Path::new(file_name)).await;
    client
        .put_object()
        .bucket(bucket_name)
        .key(key)
        .body(body.unwrap())
        .send()
        .await?;

    println!("Uploaded file: {}", file_name);
    Ok(())
}
```

- Para obter detalhes da API, consulte [PutObject](#) na Referência da API do AWS SDK para Rust.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Carregue um arquivo do armazenamento local para um bucket.

```
public func uploadFile(bucket: String, key: String, file: String) async throws
{
    let fileUrl = URL(fileURLWithPath: file)
```

```
let fileData = try Data(contentsOf: fileUrl)
let dataStream = ByteStream.from(data: fileData)

let input = PutObjectInput(
    body: dataStream,
    bucket: bucket,
    key: key
)
_ = try await client.putObject(input: input)
}
```

Carregue o conteúdo de um objeto Swift Data para um bucket.

```
public func createFile(bucket: String, key: String, withData data: Data) async
throws {
    let dataStream = ByteStream.from(data: data)

    let input = PutObjectInput(
        body: dataStream,
        bucket: bucket,
        key: key
    )
    _ = try await client.putObject(input: input)
}
```

- Para obter detalhes da API, consulte [PutObject](#) na Referência da API do AWS SDK para Swift.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Cenários do Amazon S3 usando AWS SDKs

Os exemplos de código a seguir mostram como implementar cenários comuns no Amazon S3 com AWS SDKs. Esses cenários mostram como realizar tarefas específicas chamando várias funções no Amazon S3. Cada exemplo inclui um link para o GitHub, onde você pode encontrar instruções sobre como configurar e executar o código.

Exemplos

- [Criar um URL pré-assinado para o Amazon S3 usando um AWS SDK \(p. 1537\)](#)
- [Começar a usar buckets e objetos do Amazon S3 usando um AWS SDK \(p. 1545\)](#)
- [Gerenciar objetos do Amazon S3 versionados em lotes com uma função do Lambda usando um AWS SDK \(p. 1580\)](#)
- [Usar um gerenciador de transferências para carregar e baixar arquivos de e para o Amazon S3 usando um AWS SDK \(p. 1581\)](#)
- [Trabalhar com objetos versionados do Amazon S3 usando um AWS SDK \(p. 1594\)](#)

Criar um URL pré-assinado para o Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como criar um URL pré-assinado para o S3 e carregar um objeto.

Go

SDK for Go V2

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
// Get a presigned URL for the object.  
// In order to get a presigned URL for an object, you must  
// create a Presignclient  
fmt.Println("Create Presign client")  
presignClient := s3.NewPresignClient(&client)  
  
presignResult, err := presignClient.PresignGetObject(context.TODO(),  
&s3.GetObjectInput{  
    Bucket: aws.String(name),  
    Key:    aws.String("path/myfile.jpg"),  
})  
  
if err != nil {  
    panic("Couldn't get presigned URL for GetObject")  
}  
  
fmt.Printf("Presigned URL For object: %s\n", presignResult.URL)
```

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
public static void signBucket(S3Presigner presigner, String bucketName, String  
keyName) {  
  
    try {  
        PutObjectRequest objectRequest = PutObjectRequest.builder()  
            .bucket(bucketName)  
            .key(keyName)  
            .contentType("text/plain")  
            .build();  
  
        PutObjectPresignRequest presignRequest =  
PutObjectPresignRequest.builder()  
            .signatureDuration(Duration.ofMinutes(10))  
            .putObjectRequest(objectRequest)  
            .build();  
  
        PresignedPutObjectRequest presignedRequest =  
presigner.presignPutObject(presignRequest);  
        String myURL = presignedRequest.url().toString();  
        System.out.println("Presigned URL to upload a file to: " +myURL);  
        System.out.println("Which HTTP method needs to be used when uploading a  
file: " + presignedRequest.httpRequest().method());  
  
        // Upload content to the Amazon S3 bucket by using this URL.  
    } catch (AmazonServiceException e) {  
        System.out.println("Caught an AmazonServiceException, which means your request failed to make it to the server." + e.getMessage());  
    } catch (AmazonClientException e) {  
        System.out.println("Caught an AmazonClientException, which means the client encountered an error while interacting with Amazon S3." + e.getMessage());  
    }  
}
```

```
URL url = presignedRequest.url();

// Create the connection and use it to upload the new object by using
// the presigned URL.
HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
connection.setDoOutput(true);
connection.setRequestProperty("Content-Type", "text/plain");
connection.setRequestMethod("PUT");
OutputStreamWriter out = new
OutputStreamWriter(connection.getOutputStream());
out.write("This text was uploaded as an object by using a presigned
URL.");
out.close();

connection.getResponseCode();
System.out.println("HTTP response code is " +
connection.getResponseCode());

} catch (S3Exception | IOException e) {
    e.printStackTrace();
}
}
```

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie o cliente.

```
// Create service client module using ES6 syntax.
import { S3Client } from "@aws-sdk/client-s3";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an Amazon S3 service client object.
const s3Client = new S3Client({ region: REGION });
export { s3Client };
```

Crie um URL pré-assinado para carregar um objeto em um bucket.

```
// Import the required AWS SDK clients and commands for Node.js
import {
    CreateBucketCommand,
    DeleteObjectCommand,
    PutObjectCommand,
    DeleteBucketCommand
} from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
// Amazon S3 service client module.
import { getSignedUrl } from "@aws-sdk/s3-request-presigner";
import fetch from "node-fetch";

// Set parameters
// Create a random name for the Amazon Simple Storage Service (Amazon S3) bucket
// and key
```

```
export const bucketParams = {
  Bucket: `test-bucket-${Math.ceil(Math.random() * 10 ** 10)}`,
  Key: `test-object-${Math.ceil(Math.random() * 10 ** 10)}`,
  Body: "BODY"
};
export const run = async () => {
  try {
    // Create an S3 bucket.
    console.log(`Creating bucket ${bucketParams.Bucket}`);
    await s3Client.send(new CreateBucketCommand({ Bucket: bucketParams.Bucket }));
    console.log(`Waiting for "${bucketParams.Bucket}" bucket creation...`);
  } catch (err) {
    console.log("Error creating bucket", err);
  }
  try {
    // Create a command to put the object in the S3 bucket.
    const command = new PutObjectCommand(bucketParams);
    // Create the presigned URL.
    const signedUrl = await getSignedUrl(s3Client, command, {
      expiresIn: 3600,
    });
    console.log(
      `\nPutting "${bucketParams.Key}" using signedUrl with body
      "${bucketParams.Body}" in v3`
    );
    console.log(signedUrl);
    const response = await fetch(signedUrl, {method: 'PUT', body:
      bucketParams.Body});
    console.log(
      `\nResponse returned by signed URL: ${await response.text()}\n`
    );
  } catch (err) {
    console.log("Error creating presigned URL", err);
  }
  try {
    // Delete the object.
    console.log(`\nDeleting object "${bucketParams.Key}" from bucket`);
    await s3Client.send(
      new DeleteObjectCommand({ Bucket: bucketParams.Bucket, Key:
        bucketParams.Key })
    );
  } catch (err) {
    console.log("Error deleting object", err);
  }
  try {
    // Delete the S3 bucket.
    console.log(`\nDeleting bucket ${bucketParams.Bucket}`);
    await s3Client.send(
      new DeleteBucketCommand({ Bucket: bucketParams.Bucket })
    );
  } catch (err) {
    console.log("Error deleting bucket", err);
  }
};
run();
```

Crie um URL pré-assinado para baixar um objeto de um bucket.

```
// Import the required AWS SDK clients and commands for Node.js
import {
  CreateBucketCommand,
  PutObjectCommand,
  GetObjectCommand,
  DeleteObjectCommand,
```

```
    DeleteBucketCommand }
from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js"; // Helper function that creates an
// Amazon S3 service client module.
import { getSignedUrl } from "@aws-sdk/s3-request-presigner";
const fetch = require("node-fetch");

// Set parameters
// Create a random names for the S3 bucket and key.
export const bucketParams = {
  Bucket: `test-bucket-${Math.ceil(Math.random() * 10 ** 10)}`,
  Key: `test-object-${Math.ceil(Math.random() * 10 ** 10)}`,
  Body: "BODY"
};

export const run = async () => {
  // Create an S3 bucket.
  try {
    console.log(`Creating bucket ${bucketParams.Bucket}`);
    const data = await s3Client.send(
      new CreateBucketCommand({ Bucket: bucketParams.Bucket })
    );
    return data; // For unit tests.
    console.log(`Waiting for "${bucketParams.Bucket}" bucket creation...\n`);
  } catch (err) {
    console.log("Error creating bucket", err);
  }
  // Put the object in the S3 bucket.
  try {
    console.log(`Putting object "${bucketParams.Key}" in bucket`);
    const data = await s3Client.send(
      new PutObjectCommand({
        Bucket: bucketParams.Bucket,
        Key: bucketParams.Key,
        Body: bucketParams.Body,
      })
    );
    return data; // For unit tests.
  } catch (err) {
    console.log("Error putting object", err);
  }
  // Create a presigned URL.
  try {
    // Create the command.
    const command = new GetObjectCommand(bucketParams);

    // Create the presigned URL.
    const signedUrl = await getSignedUrl(s3Client, command, {
      expiresIn: 3600,
    });
    console.log(
      `\nGetting "${bucketParams.Key}" using signedUrl with body
      "${bucketParams.Body}" in v3`
    );
    console.log(signedUrl);
    const response = await fetch(signedUrl);
    console.log(
      `\nResponse returned by signed URL: ${await response.text()}\n`
    );
  } catch (err) {
    console.log("Error creating presigned URL", err);
  }
  // Delete the object.
  try {
    console.log(`\nDeleting object "${bucketParams.Key}" from bucket`);
    const data = await s3Client.send(
```

```
        new DeleteObjectCommand({ Bucket: bucketParams.Bucket, Key:  
bucketParams.Key })  
    );  
    return data; // For unit tests.  
} catch (err) {  
    console.log("Error deleting object", err);  
}  
// Delete the S3 bucket.  
try {  
    console.log(`\nDeleting bucket ${bucketParams.Bucket}`);  
    const data = await s3Client.send(  
        new DeleteBucketCommand({ Bucket: bucketParams.Bucket, Key:  
bucketParams.Key })  
    );  
    return data; // For unit tests.  
} catch (err) {  
    console.log("Error deleting object", err);  
}  
};  
run();
```

- Para obter mais informações, consulte o [AWS SDK for JavaScript Guia do desenvolvedor](#).

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Gere um URL pré-assinado que possa realizar uma ação do S3 por tempo limitado. Use o pacote Requests (Solicitações) para fazer uma solicitação com o URL.

```
import argparse  
import logging  
import boto3  
from botocore.exceptions import ClientError  
import requests  
  
logger = logging.getLogger(__name__)  
  
def generate_presigned_url(s3_client, client_method, method_parameters,  
    expires_in):  
    """  
    Generate a presigned Amazon S3 URL that can be used to perform an action.  
  
    :param s3_client: A Boto3 Amazon S3 client.  
    :param client_method: The name of the client method that the URL performs.  
    :param method_parameters: The parameters of the specified client method.  
    :param expires_in: The number of seconds the presigned URL is valid for.  
    :return: The presigned URL.  
    """  
    try:  
        url = s3_client.generate_presigned_url(  
            ClientMethod=client_method,  
            Params=method_parameters,  
            ExpiresIn=expires_in  
        )  
        logger.info("Got presigned URL: %s", url)  
    except ClientError:
```

```
logger.exception(
    "Couldn't get a presigned URL for client method '%s'.", client_method)
raise
return url

def usage_demo():
    logging.basicConfig(level=logging.INFO, format='%(levelname)s: %(message)s')

    print('*'*88)
    print("Welcome to the Amazon S3 presigned URL demo.")
    print('*'*88)

    parser = argparse.ArgumentParser()
    parser.add_argument('bucket', help="The name of the bucket.")
    parser.add_argument(
        'key', help="For a GET operation, the key of the object in Amazon S3. For a "
        "PUT operation, the name of a file to upload.")
    parser.add_argument(
        'action', choices=('get', 'put'), help="The action to perform.")
    args = parser.parse_args()

    s3_client = boto3.client('s3')
    client_action = 'get_object' if args.action == 'get' else 'put_object'
    url = generate_presigned_url(
        s3_client, client_action, {'Bucket': args.bucket, 'Key': args.key}, 1000)

    print("Using the Requests package to send a request to the URL.")
    response = None
    if args.action == 'get':
        response = requests.get(url)
    elif args.action == 'put':
        print("Putting data to the URL.")
        try:
            with open(args.key, 'r') as object_file:
                object_text = object_file.read()
            response = requests.put(url, data=object_text)
        except FileNotFoundError:
            print(f"Couldn't find {args.key}. For a PUT operation, the key must be "
            "the "
            f"name of a file that exists on your computer.")

    if response is not None:
        print("Got response:")
        print(f"Status: {response.status_code}")
        print(response.text)

    print('*'*88)

if __name__ == '__main__':
    usage_demo()
```

Gere uma solicitação POST pré-assinada para carregar um arquivo.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""
    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                    that wraps bucket actions in a class-like structure.
        """

```

```
        self.bucket = bucket
        self.name = bucket.name

    def generate_presigned_post(self, object_key, expires_in):
        """
        Generate a presigned Amazon S3 POST request to upload a file.
        A presigned POST can be used for a limited time to let someone without an
AWS
        account upload a file to a bucket.

        :param object_key: The object key to identify the uploaded object.
        :param expires_in: The number of seconds the presigned POST is valid.
        :return: A dictionary that contains the URL and form fields that contain
                required access data.
        """
        try:
            response = self.bucket.meta.client.generate_presigned_post(
                Bucket=self.bucket.name, Key=object_key, ExpiresIn=expires_in)
            logger.info("Got presigned POST URL: %s", response['url'])
        except ClientError:
            logger.exception(
                "Couldn't get a presigned POST URL for bucket '%s' and object
'%s'", self.bucket.name, object_key)
            raise
        return response
```

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
require "aws-sdk-s3"
require "net/http"

# Creates a presigned URL that can be used to upload content to an object.
#
# @param bucket [Aws::S3::Bucket] An existing Amazon S3 bucket.
# @param object_key [String] The key to give the uploaded object.
# @return [URI, nil] The parsed URI if successful; otherwise nil.
def get_presigned_url(bucket, object_key)
    url = bucket.object(object_key).presigned_url(:put)
    puts "Created presigned URL: #{url}."
    URI(url)
rescue Aws::Errors::ServiceError => e
    puts "Couldn't create presigned URL for #{bucket.name}:#{object_key}. Here's why:
#{e.message}"
end

def run_demo
    bucket_name = "doc-example-bucket"
    object_key = "my-file.txt"
    object_content = "This is the content of my-file.txt."

    bucket = Aws::S3::Bucket.new(bucket_name)
    presigned_url = get_presigned_url(bucket, object_key)
    return unless presigned_url

    response = Net::HTTP.start(presigned_url.host) do |http|
```

```
    http.send_request("PUT", presigned_url.request_uri, object_content,
"content_type" => "")
end

case response
when Net::HTTPSuccess
  puts "Content uploaded!"
else
  puts response.value
end
end

run_demo if $PROGRAM_NAME == __FILE__
```

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Começar a usar buckets e objetos do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como:

- Criar um bucket.
- Carregar um arquivo no bucket.
- Baixar um objeto de um bucket.
- Copiar um objeto em uma subpasta em um bucket.
- Listar os objetos em um bucket.
- Excluir os objetos em um bucket.
- Excluir um bucket.

.NET

AWS SDK for .NET

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
using System;
using System.IO;
using System.Threading.Tasks;
using Amazon.S3;

public class S3_Basics
{
    public static async Task Main()
    {
        // Create an Amazon S3 client object. The constructor uses the
        // default user installed on the system. To work with Amazon S3
        // features in a different AWS Region, pass the AWS Region as a
        // parameter to the client constructor.
        IAmazonS3 client = new AmazonS3Client();
        string bucketName = string.Empty;
```

```
string filePath = string.Empty;
string keyName = string.Empty;

Console.WriteLine("Amazon Simple Storage Service (Amazon S3) basic");
Console.WriteLine("procedures. This application will:");
Console.WriteLine("\n\t1. Create a bucket");
Console.WriteLine("\n\t2. Upload an object to the new bucket");
Console.WriteLine("\n\t3. Copy the uploaded object to a folder in the
bucket");
Console.WriteLine("\n\t4. List the items in the new bucket");
Console.WriteLine("\n\t5. Delete all the items in the bucket");
Console.WriteLine("\n\t6. Delete the bucket");

Console.WriteLine("-----");

// Create a bucket.
Console.WriteLine("\nCreate a new Amazon S3 bucket.\n");

Console.Write("Please enter a name for the new bucket: ");
bucketName = Console.ReadLine();

var success = await S3Bucket.CreateBucketAsync(client, bucketName);
if (success)
{
    Console.WriteLine($"Successfully created bucket: {bucketName}.\n");
}
else
{
    Console.WriteLine($"Could not create bucket: {bucketName}.\n");
}

Console.WriteLine("Upload a file to the new bucket.");

// Get the local path and filename for the file to upload.
while (string.IsNullOrEmpty(filePath))
{
    Console.Write("Please enter the path and filename of the file to
upload: ");
    filePath = Console.ReadLine();

    // Confirm that the file exists on the local computer.
    if (!File.Exists(filePath))
    {
        Console.WriteLine($"Couldn't find {filePath}. Try again.\n");
        filePath = string.Empty;
    }
}

// Get the file name from the full path.
keyName = Path.GetFileName(filePath);

success = await S3Bucket.UploadFileAsync(client, bucketName, keyName,
filePath);

if (success)
{
    Console.WriteLine($"Successfully uploaded {keyName} from {filePath}
to {bucketName}.\n");
}
else
{
    Console.WriteLine($"Could not upload {keyName}.\n");
}

// Set the file path to an empty string to avoid overwriting the
// file we just uploaded to the bucket.
```

```
filePath = string.Empty;

// Now get a new location where we can save the file.
while (string.IsNullOrEmpty(filePath))
{
    // First get the path to which the file will be downloaded.
    Console.WriteLine("Please enter the path where the file will be
downloaded: ");
    filePath = Console.ReadLine();

    // Confirm that the file exists on the local computer.
    if (File.Exists($"{filePath}\\{keyName}"))
    {
        Console.WriteLine($"Sorry, the file already exists in that
location.\n");
        filePath = string.Empty;
    }
}

// Download an object from a bucket.
success = await S3Bucket.DownloadObjectFromBucketAsync(client,
bucketName, keyName, filePath);

if (success)
{
    Console.WriteLine($"Successfully downloaded {keyName}.\n");
}
else
{
    Console.WriteLine($"Sorry, could not download {keyName}.\n");
}

// Copy the object to a different folder in the bucket.
string folderName = string.Empty;

while (string.IsNullOrEmpty(folderName))
{
    Console.WriteLine("Please enter the name of the folder to copy your
object to: ");
    folderName = Console.ReadLine();
}

while (string.IsNullOrEmpty(keyName))
{
    // Get the name to give to the object once uploaded.
    Console.WriteLine("Enter the name of the object to copy: ");
    keyName = Console.ReadLine();
}

await S3Bucket.CopyObjectInBucketAsync(client, bucketName, keyName,
folderName);

// List the objects in the bucket.
await S3Bucket.ListBucketContentsAsync(client, bucketName);

// Delete the contents of the bucket.
await S3Bucket.DeleteBucketContentsAsync(client, bucketName);

// Deleting the bucket too quickly after deleting its contents will
// cause an error that the bucket isn't empty. So...
Console.WriteLine("Press <Enter> when you are ready to delete the
bucket.");
_ = Console.ReadLine();

// Delete the bucket.
await S3Bucket.DeleteBucketAsync(client, bucketName);
```

```
    }  
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API do AWS SDK for .NET.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjects](#)
 - [PutObject](#)

C++

SDK para C++

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
#include <iostream>  
#include <aws/core/Aws.h>  
#include <aws/s3/S3Client.h>  
#include <aws/s3/model/CopyObjectRequest.h>  
#include <aws/s3/model/CreateBucketRequest.h>  
#include <aws/s3/model/DeleteBucketRequest.h>  
#include <aws/s3/model/DeleteObjectRequest.h>  
#include <aws/s3/model/GetObjectRequest.h>  
#include <aws/s3/model/ListObjectsRequest.h>  
#include <aws/s3/model/PutObjectRequest.h>  
#include <aws/s3/model/BucketLocationConstraint.h>  
#include <aws/s3/model/CreateBucketConfiguration.h>  
#include <aws/core/utils/UUID.h>  
#include <aws/core/utils/StringUtils.h>  
#include <aws/core/utils/memory/stl/AWSAllocator.h>  
#include <aws/core/utils/memory/stl/AWSStreamFwd.h>  
#include <fstream>  
#include "awsdoc/s3/s3_examples.h"  
  
namespace AwsDoc {  
    namespace S3 {  
  
        //! Delete an S3 bucket.  
        /*!  
         \sa DeleteBucket()  
         \param bucketName the S3 bucket's name.  
         \param client an S3 client.  
         \param logProgress enables verbose logging.  
        */  
        static bool DeleteBucket(const Aws::String &bucketName, Aws::S3::S3Client &client, bool logProgress);  
  
        //! Delete an object in an S3 bucket.  
        /*!  
         \sa DeleteObjectFromBucket()  
         \param bucketName the S3 bucket's name.
```

```
\param key the key for the object in the S3 bucket.
\param client an S3 client.
\param logProgress enables verbose logging.
*/
static bool
DeleteObjectFromBucket(const Aws::String &bucketName, const Aws::String
&key, Aws::S3::S3Client &client,
                      bool logProgress);
}

//! Scenario to create, copy, and delete S3 buckets and objects.
/*!
\sa S3_GettingStartedScenario()
\param uploadFilePath path to file to upload to an S3 bucket.
\param saveFilePath path for saving a downloaded S3 object.
\param clientConfig Aws client configuration.
\param logProgress enables verbose logging.
*/
bool AwsDoc::S3::S3_GettingStartedScenario(const Aws::String &uploadFilePath, const
                                            Aws::String &saveFilePath,
                                            const Aws::Client::ClientConfiguration
                                            &clientConfig,
                                            bool logProgress) {

    Aws::S3::S3Client client(clientConfig);

    // Create a unique bucket name which is only temporary and will be deleted.
    // Format: "doc-example-bucket-" + lowercase UUID.
    Aws::String uuid = Aws::Utils::UUID::RandomUUID();
    Aws::String bucketName = "doc-example-bucket-" +
                           Aws::Utils::StringUtils::ToLower(uuid.c_str());

    // 1. Create a bucket.
    {
        Aws::S3::Model::CreateBucketRequest request;
        request.SetBucket(bucketName);

        if (clientConfig.region != Aws::Region::US_EAST_1) {
            Aws::S3::Model::CreateBucketConfiguration createBucketConfiguration;
            createBucketConfiguration.WithLocationConstraint(
                Aws::S3::Model::BucketLocationConstraintMapper::GetBucketLocationConstraintForName(
                    clientConfig.region));
            request.WithCreateBucketConfiguration(createBucketConfiguration);
        }

        Aws::S3::Model::CreateBucketOutcome outcome = client.CreateBucket(request);

        if (!outcome.IsSuccess()) {
            const Aws::S3::S3Error &err = outcome.GetError();
            std::cerr << "Error: CreateBucket: " <<
            err.GetExceptionName() << ":" << err.GetMessage() <<
            std::endl;
            return false;
        }
        else if (logProgress) {
            std::cout << "Created the bucket, '" << bucketName <<
                         "', in the region, '" << clientConfig.region << "." <<
            std::endl;
        }
    }

    // 2. Upload a local file to the bucket.
    Aws::String key = "key-for-test";
```

```
{  
    Aws::S3::Model::PutObjectRequest request;  
    request.SetBucket(bucketName);  
    request.SetKey(key);  
  
    std::shared_ptr<Aws::FStream> input_data =  
        Aws::MakeShared<Aws::FStream>("SampleAllocationTag",  
                                         uploadFilePath,  
                                         std::ios_base::in |  
                                         std::ios_base::binary);  
  
    if (!input_data->is_open()) {  
        std::cout << "Error: unable to open file, '" << uploadFilePath << "'."  
        << std::endl;  
        AwsDoc::S3::DeleteBucket(bucketName, client, logProgress);  
        return false;  
    }  
  
    request.SetBody(input_data);  
  
    Aws::S3::Model::PutObjectOutcome outcome =  
        client.PutObject(request);  
  
    if (!outcome.IsSuccess()) {  
        std::cout << "Error: PutObject: " <<  
            outcome.GetError().GetMessage() << std::endl;  
        AwsDoc::S3::DeleteObjectFromBucket(bucketName, key, client,  
logProgress);  
        AwsDoc::S3::DeleteBucket(bucketName, client, logProgress);  
        return false;  
    }  
    else if (logProgress) {  
        std::cout << "Added the object with the key, '" << key << "', to the  
bucket, '"  
                << bucketName << "." << std::endl;  
    }  
}  
  
// 3. Download the object to a local file.  
{  
    Aws::S3::Model::GetObjectRequest request;  
    request.SetBucket(bucketName);  
    request.SetKey(key);  
  
    Aws::S3::Model::GetObjectOutcome outcome =  
        client.GetObject(request);  
  
    if (!outcome.IsSuccess()) {  
        const Aws::S3::S3Error &err = outcome.GetError();  
        std::cout << "Error: GetObject: " <<  
            err.GetExceptionName() << ":" << err.GetMessage() <<  
std::endl;  
    }  
    else {  
        if (logProgress) {  
            std::cout << "Downloaded the object with the key, '" << key << "',  
in the bucket, '"  
                    << bucketName << "." << std::endl;  
        }  
  
        Aws::Iostream &ioStream = outcome.GetResultWithOwnership().  
            GetBody();  
        Aws::OFStream outStream(saveFilePath);  
        if (!outStream.is_open()) {  
            std::cout << "Error: unable to open file, '" << saveFilePath <<  
".." << std::endl;  
    }  
}
```

```
        }
        else if (logProgress) {
            outStream << ioStream.rdbuf();
            std::cout << "Wrote the downloaded object to the file ''"
            << saveFilePath << "." << std::endl;
        }
    }

// 4. Copy the object to a different "folder" in the bucket.
Aws::String copiedToKey = "test-folder/" + key;
{
    Aws::S3::Model::CopyObjectRequest request;
    request.WithBucket(bucketName)
        .WithKey(copiedToKey)
        .WithCopySource(bucketName + "/" + key);

    Aws::S3::Model::CopyObjectOutcome outcome =
        client.CopyObject(request);
    if (!outcome.IsSuccess()) {
        std::cout << "Error: CopyObject: " <<
            outcome.GetError().GetMessage() << std::endl;
    }
    else if (logProgress) {
        std::cout << "Copied the object with the key, '" << key << "', to the
key, '" << copiedToKey
                << ", in the bucket, '" << bucketName << "'." << std::endl;
    }
}

// 5. List objects in the bucket.
{
    Aws::S3::Model::ListObjectsRequest request;
    request.WithBucket(bucketName);

    Aws::S3::Model::ListObjectsOutcome outcome = client.ListObjects(request);

    if (!outcome.IsSuccess()) {
        std::cout << "Error: ListObjects: " <<
            outcome.GetError().GetMessage() << std::endl;
    }
    else if (logProgress) {
        Aws::Vector<Aws::S3::Model::Object> objects =
            outcome.GetResult().GetContents();

        std::cout << objects.size() << " objects in the bucket, '" <<
bucketName << ":" << std::endl;

        for (Aws::S3::Model::Object &object: objects) {
            std::cout << "      '" << object.GetKey() << "'"
            << std::endl;
        }
    }
}

// 6. Delete all objects in the bucket.
// All objects in the bucket must be deleted before deleting the bucket.
AwsDoc::S3::DeleteObjectFromBucket(bucketName, copiedToKey, client,
logProgress);
AwsDoc::S3::DeleteObjectFromBucket(bucketName, key, client, logProgress);

// 7. Delete the bucket.
return AwsDoc::S3::DeleteBucket(bucketName, client, logProgress);
}

bool AwsDoc::S3::DeleteObjectFromBucket(const Aws::String &bucketName, const
Aws::String &key,
```

```
Aws::S3::S3Client &client, bool logProgress) {
    Aws::S3::Model::DeleteObjectRequest request;
    request.SetBucket(bucketName);
    request.SetKey(key);

    Aws::S3::Model::DeleteObjectOutcome outcome =
        client.DeleteObject(request);

    if (!outcome.IsSuccess()) {
        std::cout << "Error: DeleteObject: " <<
            outcome.GetError().GetMessage() << std::endl;
        return false;
    }
    else if (logProgress) {
        std::cout << "Deleted the object with the key, '" << key << "', from the
        bucket, ''"
                    << bucketName << "." << std::endl;
    }

    return true;
}

bool AwsDoc::S3::DeleteBucket(const Aws::String &bucketName, Aws::S3::S3Client
&client, bool logProgress) {
    Aws::S3::Model::DeleteBucketRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::DeleteBucketOutcome outcome =
        client.DeleteBucket(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: DeleteBucket: " <<
            err.GetExceptionName() << ":" << err.GetMessage() << std::endl;
        return false;
    }
    else if (logProgress) {
        std::cout << "Deleted the bucket, '" << bucketName << "." << std::endl;
    }
    return true;
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API do AWS SDK for C++.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjects](#)
 - [PutObject](#)

Go

SDK for Go V2

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
// This bucket name is 100% unique.  
// Remember that bucket names must be globally unique among all buckets.  
  
myBucketName := "mybucket-" + (xid.New().String())  
fmt.Printf("Bucket name: %v\n", myBucketName)  
  
cfg, err := config.LoadDefaultConfig(context.TODO())  
  
if err != nil {  
    panic("Failed to load configuration")  
}  
  
s3client := s3.NewFromConfig(cfg)  
  
MakeBucket(*s3client, myBucketName)  
BucketOps(*s3client, myBucketName)  
AccountBucketOps(*s3client, myBucketName)  
BucketDelOps(*s3client, myBucketName)
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API do AWS SDK for Go.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjects](#)
 - [PutObject](#)

Java

SDK para Java 2.x

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
/**  
 * Before running this Java V2 code example, set up your development environment,  
 * including your credentials.  
 *  
 * For more information, see the following documentation topic:  
 *  
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html  
 */  
  
public class S3Scenario {  
  
    public static void main(String[] args) throws IOException {  
  
        final String usage = "\n" +  
            "Usage:\n" +  
            "  <bucketName> <key> <objectPath> <savePath> <toBucket>\n\n" +  
            "Where:\n" +  
            "  <bucketName> - The Amazon S3 bucket to create.\n\n" +
```

```
"      key - The key to use.\n\n" +
"      objectPath - The path where the file is located (for example, C:/AWS/book2.pdf). +
"      savePath - The path where the file is saved after it's downloaded
(for example, C:/AWS/book2.pdf). " +
"      toBucket - An Amazon S3 bucket to where an object is copied to
(for example, C:/AWS/book2.pdf). ";

if (args.length != 5) {
    System.out.println(usage);
    System.exit(1);
}

String bucketName = args[0];
String key = args[1];
String objectPath = args[2];
String savePath = args[3];
String toBucket = args[4] ;

ProfileCredentialsProvider credentialsProvider =
ProfileCredentialsProvider.create();
Region region = Region.US_EAST_1;
S3Client s3 = S3Client.builder()
.region(region)
.credentialsProvider(credentialsProvider)
.build();

// Create an Amazon S3 bucket.
createBucket(s3, bucketName);

// Update a local file to the Amazon S3 bucket.
uploadLocalFile(s3, bucketName, key, objectPath);

// Download the object to another local file.
getBytes(s3, bucketName, key, savePath);

// Perform a multipart upload.
String multipartKey = "multiPartKey";
multipartUpload(s3, toBucket, multipartKey);

// List all objects located in the Amazon S3 bucket.
// Show 2 ways
listAllObjects(s3, bucketName);
anotherListExample(s3, bucketName) ;

// Copy the object to another Amazon S3 bucket
copyBucketObject (s3, bucketName, key, toBucket);

// Delete the object from the Amazon S3 bucket.
deleteObjectFromBucket(s3, bucketName, key);

// Delete the Amazon S3 bucket
deleteBucket(s3, bucketName);
System.out.println("All Amazon S3 operations were successfully performed");
s3.close();
}

// Create a bucket by using a S3Waiter object
public static void createBucket( S3Client s3Client, String bucketName) {

try {
    S3Waiter s3Waiter = s3Client.waiter();
    CreateBucketRequest bucketRequest = CreateBucketRequest.builder()
        .bucket(bucketName)
        .build();
}
```

```
s3Client.createBucket(bucketRequest);
HeadBucketRequest bucketRequestWait = HeadBucketRequest.builder()
    .bucket(bucketName)
    .build();

        // Wait until the bucket is created and print out the response.
        WaiterResponse<HeadBucketResponse> waiterResponse =
s3Waiter.waitUntilBucketExists(bucketRequestWait);
        waiterResponse.matched().response().ifPresent(System.out::println);
        System.out.println(bucketName + " is ready");

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void deleteBucket(S3Client client, String bucket) {
    DeleteBucketRequest deleteBucketRequest = DeleteBucketRequest.builder()
        .bucket(bucket)
        .build();

    client.deleteBucket(deleteBucketRequest);
    System.out.println(bucket + " was deleted.");
}

/**
 * Upload an object in parts
 */
private static void multipartUpload(S3Client s3, String bucketName, String key)
{

    int mB = 1024 * 1024;
    // First create a multipart upload and get the upload id
    CreateMultipartUploadRequest createMultipartUploadRequest =
CreateMultipartUploadRequest.builder()
    .bucket(bucketName)
    .key(key)
    .build();

    CreateMultipartUploadResponse response =
s3.createMultipartUpload(createMultipartUploadRequest);
    String uploadId = response.uploadId();
    System.out.println(uploadId);

    // Upload all the different parts of the object
    UploadPartRequest uploadPartRequest1 = UploadPartRequest.builder()
        .bucket(bucketName)
        .key(key)
        .uploadId(uploadId)
        .partNumber(1).build();

    String etag1 = s3.uploadPart(uploadPartRequest1,
RequestBody.fromByteBuffer(getRandomByteBuffer(5 * mB))).eTag();
    CompletedPart part1 =
CompletedPart.builder().partNumber(1).eTag(etag1).build();

    UploadPartRequest uploadPartRequest2 =
UploadPartRequest.builder().bucket(bucketName).key(key)
        .uploadId(uploadId)
        .partNumber(2).build();
    String etag2 = s3.uploadPart(uploadPartRequest2,
RequestBody.fromByteBuffer(getRandomByteBuffer(3 * mB))).eTag();
    CompletedPart part2 =
CompletedPart.builder().partNumber(2).eTag(etag2).build();
```

```
// Call completeMultipartUpload operation to tell S3 to merge all uploaded
// parts and finish the multipart operation.
CompletedMultipartUpload completedMultipartUpload =
CompletedMultipartUpload.builder()
    .parts(part1, part2)
    .build();

CompleteMultipartUploadRequest completeMultipartUploadRequest =
CompleteMultipartUploadRequest.builder()
    .bucket(bucketName)
    .key(key)
    .uploadId(uploadId)
    .multipartUpload(completedMultipartUpload)
    .build();

s3.completeMultipartUpload(completeMultipartUploadRequest);
}

private static ByteBuffer getRandomByteBuffer(int size) {
    byte[] b = new byte[size];
    new Random().nextBytes(b);
    return ByteBuffer.wrap(b);
}

// Return a byte array
private static byte[] getObjectFile(String filePath) {

    FileInputStream fileInputStream = null;
    byte[] bytesArray = null;

    try {
        File file = new File(filePath);
        bytesArray = new byte[(int) file.length()];
        fileInputStream = new FileInputStream(file);
        fileInputStream.read(bytesArray);

    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (fileInputStream != null) {
            try {
                fileInputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    return bytesArray;
}

public static void getObjectBytes (S3Client s3, String bucketName, String
keyName, String path ) {

    try {
        GetObjectRequest objectRequest = GetObjectRequest
            .builder()
            .key(keyName)
            .bucket(bucketName)
            .build();

        ResponseBytes<GetObjectResponse> objectBytes =
s3.getObjectAsBytes(objectRequest);
        byte[] data = objectBytes.asByteArray();

        // Write the data to a local file.
        File myFile = new File(path );
    }
}
```

```
OutputStream os = new FileOutputStream(myFile);
os.write(data);
System.out.println("Successfully obtained bytes from an S3 object");
os.close();

} catch (IOException ex) {
    ex.printStackTrace();
} catch (S3Exception e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}

}

public static void uploadLocalFile(S3Client s3, String bucketName, String key,
String objectPath) {

    PutObjectRequest objectRequest = PutObjectRequest.builder()
        .bucket(bucketName)
        .key(key)
        .build();

    s3.putObject(objectRequest,
RequestBody.fromBytes(getObjectFile(objectPath)));
}

public static void listAllObjects(S3Client s3, String bucketName) {

    ListObjectsV2Request listObjectsReqManual = ListObjectsV2Request.builder()
        .bucket(bucketName)
        .maxKeys(1)
        .build();

    boolean done = false;
    while (!done) {
        ListObjectsV2Response listObjResponse =
s3.listObjectsV2(listObjectsReqManual);
        for (S3Object content : listObjResponse.contents()) {
            System.out.println(content.key());
        }

        if (listObjResponse.nextContinuationToken() == null) {
            done = true;
        }

        listObjectsReqManual = listObjectsReqManual.toBuilder()
            .continuationToken(listObjResponse.nextContinuationToken())
            .build();
    }
}

public static void anotherListExample(S3Client s3, String bucketName) {

    ListObjectsV2Request listReq = ListObjectsV2Request.builder()
        .bucket(bucketName)
        .maxKeys(1)
        .build();

    ListObjectsV2Iterable listRes = s3.listObjectsV2Paginator(listReq);

    // Process response pages
    listRes.stream()
        .flatMap(r -> r.contents().stream())
        .forEach(content -> System.out.println(" Key: " + content.key() + " size
= " + content.size()));
}
```

```
// Helper method to work with paginated collection of items directly
listRes.contents().stream()
    .forEach(content -> System.out.println(" Key: " + content.key() + " size = " + content.size()));

    for (S3Object content : listRes.contents()) {
        System.out.println(" Key: " + content.key() + " size = " + content.size());
    }
}

public static void deleteObjectFromBucket(S3Client s3, String bucketName,
String key) {

    DeleteObjectRequest deleteObjectRequest = DeleteObjectRequest.builder()
        .bucket(bucketName)
        .key(key)
        .build();

    s3.deleteObject(deleteObjectRequest);
    System.out.println(key +" was deleted");
}

public static String copyBucketObject (S3Client s3, String fromBucket, String objectKey, String toBucket) {

    String encodedUrl = null;
    try {
        encodedUrl = URLEncoder.encode(fromBucket + "/" + objectKey,
StandardCharsets.UTF_8.toString());
    } catch (UnsupportedEncodingException e) {
        System.out.println("URL could not be encoded: " + e.getMessage());
    }
    CopyObjectRequest copyReq = CopyObjectRequest.builder()
        .copySource(encodedUrl)
        .destinationBucket(toBucket)
        .destinationKey(objectKey)
        .build();

    try {
        CopyObjectResponse copyRes = s3.copyObject(copyReq);
        System.out.println("The "+ objectKey +" was copied to "+toBucket);
        return copyRes.copyObjectResult().toString();
    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }

    return "";
}
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API do AWS SDK for Java 2.x.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)

- [ListObjects](#)
- [PutObject](#)

JavaScript

SDK para JavaScript V3

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
import {
  CreateBucketCommand,
  PutObjectCommand,
  CopyObjectCommand,
  DeleteObjectCommand,
  DeleteBucketCommand,
  GetObjectCommand
} from "@aws-sdk/client-s3";
import { s3Client, REGION } from "../libs/s3Client.js"; // Helper function that
creates an Amazon S3 service client module.

if (process.argv.length < 5) {
  console.log(
    "Usage: node s3_basics.js <the bucket name> <the AWS Region to use> <object
name> <object content>\n"
    "Example: node s3_basics_full.js test-bucket 'test.txt' 'Test Content'"
  );
}
const bucket_name = process.argv[2];
const object_key = process.argv[3];
const object_content = process.argv[4];

export const run = async (bucket_name, object_key, object_content) => {
  try {
    const create_bucket_params = {
      Bucket: bucket_name
    };
    console.log("\nCreating the bucket, named " + bucket_name + "...\\n");
    console.log("about to create");
    const data = await s3Client.send(
      new CreateBucketCommand(create_bucket_params)
    );
    console.log("Bucket created at ", data.Location);
    try {
      console.log(
        "\nCreated and uploaded an object named " +
        object_key +
        " to first bucket " +
        bucket_name +
        "...\\n"
      );
      // Set the parameters for the object to upload.
      const object_upload_params = {
        Bucket: bucket_name,
        // Specify the name of the new object. For example, 'test.html'.
        // To create a directory for the object, use '/'. For example, 'myApp/
package.json'.
        Key: object_key,
        // Content of the new object.
        Body: object_content,
      };
    }
  }
}
```

```
// Create and upload the object to the first S3 bucket.
await s3Client.send(new PutObjectCommand(object_upload_params));
console.log(
    "Successfully uploaded object: " +
    object_upload_params.Bucket +
    "/" +
    object_upload_params.Key
);
try {
    const download_bucket_params = {
        Bucket: bucket_name,
        Key: object_key
    };
    console.log(
        "\nDownloading " +
        object_key +
        " from" +
        bucket_name +
        "...\\n"
    );
    // Create a helper function to convert a ReadableStream into a string.
    const streamToString = (stream) =>
        new Promise((resolve, reject) => {
            const chunks = [];
            stream.on("data", (chunk) => chunks.push(chunk));
            stream.on("error", reject);
            stream.on("end", () =>
                resolve(Buffer.concat(chunks).toString("utf8"))
            );
        });
    // Get the object from the Amazon S3 bucket. It is returned as a
    // ReadableStream.
    const data = await s3Client.send(new
GetObjectCommand(download_bucket_params));
    // Convert the ReadableStream to a string.
    const bodyContents = await streamToString(data.Body);
    console.log(bodyContents);
    try {
        // Copy the object from the first bucket to the second bucket.
        const copy_object_params = {
            Bucket: bucket_name,
            CopySource: "/" + bucket_name + "/" + object_key,
            Key: "copy-destination/" + object_key,
        };
        console.log(
            "\nCopying " +
            object_key +
            " from" +
            bucket_name +
            " to " +
            bucket_name +
            "/" +
            copy_object_params.Key +
            "...\\n"
        );
        await s3Client.send(new CopyObjectCommand(copy_object_params));
        console.log("Success, object copied to folder.");
        try {
            console.log("\nDeleting " + object_key + " from" + bucket_name);
            const delete_object_from_bucket_params = {
                Bucket: bucket_name,
                Key: object_key,
            };
            await s3Client.send(
                new DeleteObjectCommand(delete_object_from_bucket_params)
            );
        }
    }
}
```

```
};

console.log("Success. Object deleted from bucket.");
try {
    console.log(
        "\nDeleting " +
        object_key +
        " from " +
        bucket_name +
        "/copy-destination folder"
    );
const delete_object_from_folder_params = {
    Bucket: bucket_name,
    Key: "copy-destination/" + object_key,
};

await s3Client.send(
    new DeleteObjectCommand(delete_object_from_folder_params)
);
console.log("Success. Object deleted from folder.");
try {
    console.log(
        "\nDeleting the bucket named " + bucket_name + "...\\n"
    );
const delete_bucket_params = {Bucket: bucket_name};
await s3Client.send(
    new DeleteBucketCommand(delete_bucket_params)
);
console.log("Success. First bucket deleted.");
return "Run successfully"; // For unit tests.
} catch (err) {
    console.log("Error deleting object from folder.", err);
    process.exit(1);
}
} catch (err) {
    console.log("Error deleting bucket.", err);
    process.exit(1);
}
} catch (err) {
    console.log("Error deleting object from bucket.", err);
    process.exit(1);
}
} catch (err) {
    console.log("Error copying object from to folder", err);
    process.exit(1);
}
} catch (err) {
    console.log("Error downloading object", err);
    process.exit(1);
}
}catch (err) {
    console.log("Error creating and upload object to bucket", err);
    process.exit(1);
}
console.log("works");
} catch (err) {
    console.log("Error creating bucket", err);
}
};

run(bucket_name, object_key, object_content);
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API do AWS SDK for JavaScript.

- [CopyObject](#)
- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteObjects](#)
- [GetObject](#)
- [ListObjects](#)
- [PutObject](#)

Kotlin

SDK para Kotlin

Note

Essa documentação é de pré-lançamento para um recurso em versão de pré-visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
suspend fun main(args: Array<String>) {  
  
    val usage = """  
Usage:  
    <bucketName> <key> <objectPath> <savePath> <toBucket>  
  
Where:  
    bucketName - The Amazon S3 bucket to create.  
    key - The key to use.  
    objectPath - The path where the file is located (for example, C:/AWS/  
book2.pdf).  
    savePath - The path where the file is saved after it's downloaded (for  
example, C:/AWS/book2.pdf).  
    toBucket - An Amazon S3 bucket to where an object is copied to (for  
example, C:/AWS/book2.pdf).  
    """  
  
    if (args.size != 4) {  
        println(usage)  
        exitProcess(1)  
    }  
  
    val bucketName = args[0]  
    val key = args[1]  
    val objectPath = args[2]  
    val savePath = args[3]  
    val toBucket = args[4]  
  
    // Create an Amazon S3 bucket.  
    createBucket(bucketName)  
  
    // Update a local file to the Amazon S3 bucket.  
    putObject(bucketName, key, objectPath)  
  
    // Download the object to another local file.  
    getObject(bucketName, key, savePath)  
  
    // List all objects located in the Amazon S3 bucket.  
    listBucketObs(bucketName)
```

```
// Copy the object to another Amazon S3 bucket
copyBucketOb(bucketName, key, toBucket)

// Delete the object from the Amazon S3 bucket.
deleteBucketObs(bucketName, key)

// Delete the Amazon S3 bucket.
deleteBucket(bucketName)
println("All Amazon S3 operations were successfully performed")
}

suspend fun createBucket(bucketName: String) {

    val request = CreateBucketRequest {
        bucket = bucketName
    }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.createBucket(request)
        println("$bucketName is ready")
    }
}

suspend fun putObject(bucketName: String, objectKey: String, objectPath: String) {

    val metadataVal = mutableMapOf<String, String>()
    metadataVal["myVal"] = "test"

    val request = PutObjectRequest {
        bucket = bucketName
        key = objectKey
        metadata = metadataVal
        this.body = Paths.get(objectPath).asByteStream()
    }

    S3Client { region = "us-east-1" }.use { s3 ->
        val response = s3.putObject(request)
        println("Tag information is ${response.eTag}")
    }
}

suspend fun getObject(bucketName: String, keyName: String, path: String) {

    val request = GetObjectRequest {
        key = keyName
        bucket = bucketName
    }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.getObject(request) { resp ->
            val myFile = File(path)
            resp.body?.writeToFile(myFile)
            println("Successfully read $keyName from $bucketName")
        }
    }
}

suspend fun listBucketObs(bucketName: String) {

    val request = ListObjectsRequest {
        bucket = bucketName
    }

    S3Client { region = "us-east-1" }.use { s3 ->
        val response = s3.listObjects(request)
```

```
        response.contents?.forEach { myObject ->
            println("The name of the key is ${myObject.key}")
            println("The owner is ${myObject.owner}")
        }
    }

suspend fun copyBucketOb(fromBucket: String, objectKey: String, toBucket: String) {

    var encodedUrl = ""
    try {
        encodedUrl = URLEncoder.encode("$fromBucket/$objectKey",
StandardCharsets.UTF_8.toString())
    } catch (e: UnsupportedEncodingException) {
        println("URL could not be encoded: " + e.message)
    }

    val request = CopyObjectRequest {
        copySource = encodedUrl
        bucket = toBucket
        key = objectKey
    }
    S3Client { region = "us-east-1" }.use { s3 ->
        s3.copyObject(request)
    }
}

suspend fun deleteBucketObs(bucketName: String, objectName: String) {

    val objectId = ObjectIdentifier {
        key = objectName
    }

    val delOb = Delete {
        objects = listOf(objectId)
    }

    val request = DeleteObjectsRequest {
        bucket = bucketName
        delete = delOb
    }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.deleteObjects(request)
        println("$objectName was deleted from $bucketName")
    }
}

suspend fun deleteBucket(bucketName: String?) {

    val request = DeleteBucketRequest {
        bucket = bucketName
    }
    S3Client { region = "us-east-1" }.use { s3 ->
        s3.deleteBucket(request)
        println("The $bucketName was successfully deleted!")
    }
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API do AWS SDK para Kotlin.
 - [CopyObject](#)
 - [CreateBucket](#)

- [DeleteBucket](#)
- [DeleteObjects](#)
- [GetObject](#)
- [ListObjects](#)
- [PutObject](#)

PHP

SDK for PHP

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

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

use Aws\S3\S3Client;

echo("-----\n");
print("Welcome to the Amazon S3 getting started demo using PHP!\n");
echo("-----\n");

$region = 'us-west-2';
$version = 'latest';

$s3client = new S3Client([
    'region' => $region,
    'version' => $version
]);
/* Inline declaration example
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2', 'version' => 'latest']);
*/

$bucket_name = "doc-example-bucket-" . uniqid();

try {
    $s3client->createBucket([
        'Bucket' => $bucket_name,
        'CreateBucketConfiguration' => ['LocationConstraint' => $region],
    ]);
    echo "Created bucket named: $bucket_name \n";
} catch (Exception $exception) {
    echo "Failed to create bucket $bucket_name with error: " . $exception-
>getMessage();
    exit("Please fix error with bucket creation before continuing.");
}

$file_name = "local-file-" . uniqid();
try {
    $s3client->putObject([
        'Bucket' => $bucket_name,
        'Key' => $file_name,
        'SourceFile' => 'testfile.txt'
    ]);
    echo "Uploaded $file_name to $bucket_name.\n";
} catch (Exception $exception) {
    echo "Failed to upload $file_name with error: " . $exception->getMessage();
    exit("Please fix error with file upload before continuing.");
}

try {
```

```
$file = $s3client->GetObject([
    'Bucket' => $bucket_name,
    'Key' => $file_name,
]);
$body = $file->get('Body');
$body->rewind();
echo "Downloaded the file and it begins with: {$body->read(26)}.\n";
} catch (Exception $exception) {
    echo "Failed to download $file_name from $bucket_name with error: " .
$exception->getMessage();
    exit("Please fix error with file downloading before continuing.");
}

try {
    $folder = "copied-folder";
    $s3client->copyObject([
        'Bucket' => $bucket_name,
        'CopySource' => "$bucket_name/$file_name",
        'Key' => "$folder/$file_name-copy",
    ]);
    echo "Copied $file_name to $folder/$file_name-copy.\n";
} catch (Exception $exception) {
    echo "Failed to copy $file_name with error: " . $exception->getMessage();
    exit("Please fix error with object copying before continuing.");
}

try {
    $contents = $s3client->listObjects([
        'Bucket' => $bucket_name,
    ]);
    echo "The contents of your bucket are: \n";
    foreach ($contents['Contents'] as $content) {
        echo $content['Key'] . "\n";
    }
} catch (Exception $exception) {
    echo "Failed to list objects in $bucket_name with error: " . $exception-
>getMessage();
    exit("Please fix error with listing objects before continuing.");
}

try {
    $objects = [];
    foreach ($contents['Contents'] as $content) {
        $objects[] = [
            'Key' => $content['Key'],
        ];
    }
    $s3client->deleteObjects([
        'Bucket' => $bucket_name,
        'Key' => $file_name,
        'Delete' => [
            'Objects' => $objects,
        ],
    ]);
    $check = $s3client->listObjects([
        'Bucket' => $bucket_name,
    ]);
    if (count($check) <= 0) {
        throw new Exception("Bucket wasn't empty.");
    }
    echo "Deleted all objects and folders from $bucket_name.\n";
} catch (Exception $exception) {
    echo "Failed to delete $file_name from $bucket_name with error: " . $exception-
>getMessage();
    exit("Please fix error with object deletion before continuing.");
}
```

```
try {
    $s3client->deleteBucket([
        'Bucket' => $bucket_name,
    ]);
    echo "Deleted bucket $bucket_name.\n";
} catch (Exception $exception) {
    echo "Failed to delete $bucket_name with error: " . $exception->getMessage();
    exit("Please fix error with bucket deletion before continuing.");
}

echo "Successfully ran the Amazon S3 with PHP demo.\n";
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API do AWS SDK for PHP.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjects](#)
 - [PutObject](#)

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
import io
import os
import uuid

import boto3
from boto3.s3.transfer import S3UploadFailedError
from botocore.exceptions import ClientError


def do_scenario(s3_resource):
    print('*'*88)
    print("Welcome to the Amazon S3 getting started demo!")
    print('*'*88)

    bucket_name = f'doc-example-bucket-{uuid.uuid4()}' 
    bucket = s3_resource.Bucket(bucket_name)
    try:
        bucket.create(
            CreateBucketConfiguration={
                'LocationConstraint': s3_resource.meta.client.meta.region_name})
        print(f"Created demo bucket named {bucket.name}.")
    except ClientError as err:
        print(f" Tried and failed to create demo bucket {bucket_name}.")
        print(f"\t{err.response['Error']['Code']}:{err.response['Error']['Message']}")
        print(f"\nCan't continue the demo without a bucket!")
        return
```

```
file_name = None
while file_name is None:
    file_name = input("\nEnter a file you want to upload to your bucket: ")
    if not os.path.exists(file_name):
        print(f"Couldn't find file {file_name}. Are you sure it exists?")
        file_name = None

obj = bucket.Object(os.path.basename(file_name))
try:
    obj.upload_file(file_name)
    print(f"Uploaded file {file_name} into bucket {bucket.name} with key {obj.key}.")
except S3UploadFailedError as err:
    print(f"Couldn't upload file {file_name} to {bucket.name}.")
    print(f"\t{err}")

answer = input(f"\nDo you want to download {obj.key} into memory (y/n)? ")
if answer.lower() == 'y':
    data = io.BytesIO()
    try:
        obj.download_fileobj(data)
        data.seek(0)
        print(f"Got your object. Here are the first 20 bytes:\n")
        print(f"\t{data.read(20)}")
    except ClientError as err:
        print(f"Couldn't download {obj.key}.")
        print(f"\t{err.response['Error']['Code']}:{err.response['Error']['Message']}")

    answer = input(
        f"\nDo you want to copy {obj.key} to a subfolder in your bucket (y/n)? ")
    if answer.lower() == 'y':
        dest_obj = bucket.Object(f'demo-folder/{obj.key}')
        try:
            dest_obj.copy({'Bucket': bucket.name, 'Key': obj.key})
            print(f"Copied {obj.key} to {dest_obj.key}.")
        except ClientError as err:
            print(f"Couldn't copy {obj.key} to {dest_obj.key}.")
            print(f"\t{err.response['Error']['Code']}:{err.response['Error']['Message']}")

print("\nYour bucket contains the following objects:")
try:
    for o in bucket.objects.all():
        print(f"\t{o.key}")
except ClientError as err:
    print(f"Couldn't list the objects in bucket {bucket.name}.")
    print(f"\t{err.response['Error']['Code']}:{err.response['Error']['Message']}")

answer = input(
    "\nDo you want to delete all of the objects as well as the bucket (y/n)? ")
if answer.lower() == 'y':
    try:
        bucket.objects.delete()
        bucket.delete()
        print(f"Emptied and deleted bucket {bucket.name}.\n")
    except ClientError as err:
        print(f"Couldn't empty and delete bucket {bucket.name}.")
        print(f"\t{err.response['Error']['Code']}:{err.response['Error']['Message']}")

print("Thanks for watching!")
print('*'*88)
```

```
if __name__ == '__main__':
    do_scenario(boto3.resource('s3'))
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API do AWS SDK para Python (Boto3).
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjects](#)
 - [PutObject](#)

Ruby

SDK for Ruby

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

```
require "aws-sdk-s3"

# Wraps the getting started scenario actions.
class ScenarioGettingStarted
    attr_reader :s3_resource

    # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
    def initialize(s3_resource)
        @s3_resource = s3_resource
    end

    # Creates a bucket with a random name in the currently configured account and
    # AWS Region.
    #
    # @return [Aws::S3::Bucket] The newly created bucket.
    def create_bucket
        bucket = @s3_resource.create_bucket(
            bucket: "doc-example-bucket-#{Random.uuid}",
            create_bucket_configuration: {
                location_constraint: s3_resource.client.config.region
            }
        )
        puts("Created demo bucket named #{bucket.name}.")
    rescue Aws::Errors::ServiceError => e
        puts("Tried and failed to create demo bucket.")
        puts("\t#{e.code}: #{e.message}")
        puts("\nCan't continue the demo without a bucket!")
        raise
    else
        bucket
    end

    # Requests a file name from the user.
    #
    # @return The name of the file.
    def request_file_name
        file_name = "
```

```
while file_name.empty?
  puts("\nEnter a file you want to upload to your bucket: ")
  file_name = gets.chomp
  unless File.file?(file_name)
    puts("Couldn't find file #{file_name}. Are you sure it exists?")
    file_name = ""
  end
end
file_name
end

# Uploads a file from your computer to an Amazon S3 bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket where the file is uploaded.
# @return [Aws::S3::Object] The Amazon S3 object that contains the uploaded file.
def upload_file(bucket)
  file_name = request_file_name
  s3_object = bucket.object(File.basename(file_name))
  s3_object.upload_file(file_name)
  puts("Uploaded file #{file_name} into bucket #{bucket.name} with key
#{s3_object.key}.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't upload file #{file_name} to #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  s3_object
end

# Downloads an Amazon S3 object to a file.
#
# @param s3_object [Aws::S3::Object] The object to download.
def download_file(s3_object)
  puts("\nDo you want to download #{s3_object.key} to a local file (y/n)? ")
  answer = gets.chomp.downcase
  if answer == "y"
    puts("Enter a name for the downloaded file: ")
    file_name = gets.chomp
    s3_object.download_file(file_name)
    puts("Object #{s3_object.key} successfully downloaded to #{file_name}.")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't download #{s3_object.key}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end

# Copies an Amazon S3 object to a subfolder within the same bucket.
#
# @param source_object [Aws::S3::Object] The source object to copy.
# @return [Aws::S3::Object, nil] The destination object.
def copy_object(source_object)
  dest_object = nil
  puts("\nDo you want to copy #{source_object.key} to a subfolder in your bucket
(y/n)? ")
  answer = gets.chomp.downcase
  if answer == "y"
    dest_object = source_object.bucket.object("demo-folder/#{source_object.key}")
    dest_object.copy_from(source_object)
    puts("Copied #{source_object.key} to #{dest_object.key}.")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't copy #{source_object.key}.")
  puts("\t#{e.code}: #{e.message}")
  raise
else
```

```
    dest_object
end

# Lists the objects in an Amazon S3 bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket to query.
def list_objects(bucket)
  puts("\nYour bucket contains the following objects:")
  bucket.objects.each do |obj|
    puts("\t#{obj.key}")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't list the objects in bucket #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end

# Deletes the objects in an Amazon S3 bucket and deletes the bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket to empty and delete.
def delete_bucket(bucket)
  puts("\nDo you want to delete all of the objects as well as the bucket (y/n)?")
  answer = gets.chomp.downcase
  if answer == "y"
    bucket.objects.batch_delete!
    bucket.delete
    puts("Emptied and deleted bucket #{bucket.name}.\n")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't empty and delete bucket #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end
end

# Runs the Amazon S3 getting started scenario.
def run_scenario(scenario)
  puts("-" * 88)
  puts("Welcome to the Amazon S3 getting started demo!")
  puts("-" * 88)

  bucket = scenario.create_bucket
  s3_object = scenario.upload_file(bucket)
  scenario.download_file(s3_object)
  scenario.copy_object(s3_object)
  scenario.list_objects(bucket)
  scenario.delete_bucket(bucket)

  puts("Thanks for watching!")
  puts("-" * 88)
rescue Aws::Errors::ServiceError
  puts("Something went wrong with the demo!")
end

run_scenario(ScenarioGettingStarted.new(Aws::S3::Resource.new)) if $PROGRAM_NAME ==
__FILE__
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API do AWS SDK for Ruby.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)

- [DeleteObjects](#)
- [GetObject](#)
- [ListObjects](#)
- [PutObject](#)

Rust

SDK for Rust

Note

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Código para a caixa binária que executa o cenário.

```
use aws_config::meta::region::RegionProviderChain;
use aws_sdk_s3::{Client, Error, Region};
use uuid::Uuid;

#[tokio::main]
async fn main() -> Result<(), Error> {
    let (region, client, bucket_name, file_name, key, target_key) =
        initialize_variables().await;

    if let Err(e) = run_s3_operations(region, client, bucket_name, file_name, key,
target_key).await
    {
        println!("{}{:?}", e);
    };

    Ok(())
}

async fn initialize_variables() -> (Region, Client, String, String, String, String)
{
    let region_provider = RegionProviderChain::first_try(Region::new("us-west-2"));
    let region = region_provider.region().await.unwrap();

    let shared_config =
aws_config::from_env().region(region_provider).load().await;
    let client = Client::new(&shared_config);

    let bucket_name = format!("{}{}", "doc-example-bucket-", Uuid::new_v4());
    let file_name = "s3/testfile.txt".to_string();
    let key = "test file key name".to_string();
    let target_key = "target_key".to_string();

    (region, client, bucket_name, file_name, key, target_key)
}

async fn run_s3_operations(
    region: Region,
    client: Client,
```

```
    bucket_name: String,
    file_name: String,
    key: String,
    target_key: String,
) -> Result<(), Error> {
    s3_service::create_bucket(&client, &bucket_name, region.as_ref()).await?;
    s3_service::upload_object(&client, &bucket_name, &file_name, &key).await?;
    s3_service::download_object(&client, &bucket_name, &key).await?;
    s3_service::copy_object(&client, &bucket_name, &key, &target_key).await?;
    s3_service::list_objects(&client, &bucket_name).await?;
    s3_service::delete_objects(&client, &bucket_name).await?;
    s3_service::delete_bucket(&client, &bucket_name).await?;

    Ok(())
}
```

Uma caixa de biblioteca com ações comuns chamadas pelo binário.

```
use aws_sdk_s3::model::{
    BucketLocationConstraint, CreateBucketConfiguration, Delete, ObjectIdentifier,
};
use aws_sdk_s3::output::ListObjectsV2Output;
use aws_sdk_s3::types::ByteStream;
use aws_sdk_s3::{Client, Error};
use std::path::Path;
use std::str;

pub async fn delete_bucket(client: &Client, bucket_name: &str) -> Result<(), Error> {
    client.delete_bucket().bucket(bucket_name).send().await?;
    println!("bucket deleted");
    Ok(())
}

pub async fn delete_objects(client: &Client, bucket_name: &str) -> Result<(), Error> {
    let objects = client.list_objects_v2().bucket(bucket_name).send().await?;

    let mut delete_objects: Vec<ObjectIdentifier> = vec![];
    for obj in objects.contents().unwrap_or_default() {
        let obj_id = ObjectIdentifier::builder()
            .set_key(Some(obj.key().unwrap().to_string()))
            .build();
        delete_objects.push(obj_id);
    }
    client
        .delete_objects()
        .bucket(bucket_name)
        .delete(Delete::builder().set_objects(Some(delete_objects)).build())
        .send()
        .await?;

    let objects: ListObjectsV2Output =
        client.list_objects_v2().bucket(bucket_name).send().await?;
    match objects.key_count {
        0 => Ok(()),
        _ => Err(Error::Unhandled(Box::from(
            "There were still objects left in the bucket.",
        ))),
    }
}
```

```
pub async fn list_objects(client: &Client, bucket_name: &str) -> Result<(), Error> {
    let objects = client.list_objects_v2().bucket(bucket_name).send().await?;
    println!("Objects in bucket:");
    for obj in objects.contents().unwrap_or_default() {
        println!("{}:{}", obj.key().unwrap());
    }
}

Ok(())

pub async fn copy_object(
    client: &Client,
    bucket_name: &str,
    object_key: &str,
    target_key: &str,
) -> Result<(), Error> {
    let mut source_bucket_and_object: String = "".to_owned();
    source_bucket_and_object.push_str(bucket_name);
    source_bucket_and_object.push('/');
    source_bucket_and_object.push_str(object_key);

    client
        .copy_object()
        .copy_source(source_bucket_and_object)
        .bucket(bucket_name)
        .key(target_key)
        .send()
        .await?;

    Ok(())
}

pub async fn download_object(client: &Client, bucket_name: &str, key: &str) ->
Result<(), Error> {
    let resp = client
        .get_object()
        .bucket(bucket_name)
        .key(key)
        .send()
        .await?;

    let data = resp.body.collect().await;
    println!(
        "Data from downloaded object: {}:{}",
        data.unwrap().into_bytes().slice(0..20)
    );

    Ok(())
}

pub async fn upload_object(
    client: &Client,
    bucket_name: &str,
    file_name: &str,
    key: &str,
) -> Result<(), Error> {
    let body = ByteStream::from_path(Path::new(file_name)).await;
    client
        .put_object()
        .bucket(bucket_name)
        .key(key)
        .body(body.unwrap())
        .send()
        .await?;

    println!("Uploaded file: {}", file_name);
}
```

```
        Ok(())
    }

pub async fn create_bucket(client: &Client, bucket_name: &str, region: &str) ->
    Result<(), Error> {
    let constraint = BucketLocationConstraint::from(region);
    let cfg = CreateBucketConfiguration::builder()
        .location_constraint(constraint)
        .build();
    client
        .create_bucket()
        .create_bucket_configuration(cfg)
        .bucket(bucket_name)
        .send()
        .await?;
    println!("{}", bucket_name);
    Ok(())
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API do AWS SDK para Rust.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjects](#)
 - [PutObject](#)

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Uma classe Swift que gerencia chamadas para o SDK para Swift.

```
import Foundation
import AWSS3
import ClientRuntime
import AWSClientRuntime

/// A class containing all the code that interacts with the AWS SDK for Swift.
public class ServiceHandler {
    let client: S3Client

    /// Initialize and return a new ``ServiceHandler`` object, which is used to
    /// drive the AWS calls
    /// used for the example.
    ///
    /// - Returns: A new ``ServiceHandler`` object, ready to be called to
    /// execute AWS operations.
```

```
public init() async {
    do {
        client = try await S3Client()
    } catch {
        print("ERROR: ", dump(error, name: "Initializing s3 client"))
        exit(1)
    }
}

/// Create a new user given the specified name.
///
/// - Parameters:
///   - name: Name of the bucket to create.
/// Throws an exception if an error occurs.
public func createBucket(name: String) async throws {
    let config = S3ClientTypes.CreateBucketConfiguration(
        locationConstraint: .usEast2
    )
    let input = CreateBucketInput(
        bucket: name,
        createBucketConfiguration: config
    )
    _ = try await client.createBucket(input: input)
}

/// Delete a bucket.
/// - Parameter name: Name of the bucket to delete.
public func deleteBucket(name: String) async throws {
    let input = DeleteBucketInput(
        bucket: name
    )
    _ = try await client.deleteBucket(input: input)
}

/// Upload a file from local storage to the bucket.
/// - Parameters:
///   - bucket: Name of the bucket to upload the file to.
///   - key: Name of the file to create.
///   - file: Path name of the file to upload.
public func uploadFile(bucket: String, key: String, file: String) async throws {
    let urlString = URL(fileURLWithPath: file)
    let fileData = try Data(contentsOf: urlString)
    let dataStream = ByteStream.from(data: fileData)

    let input = PutObjectInput(
        body: dataStream,
        bucket: bucket,
        key: key
    )
    _ = try await client.putObject(input: input)
}

/// Create a file in the specified bucket with the given name. The new
/// file's contents are uploaded from a `Data` object.
///
/// - Parameters:
///   - bucket: Name of the bucket to create a file in.
///   - key: Name of the file to create.
///   - data: A `Data` object to write into the new file.
public func createFile(bucket: String, key: String, withData data: Data) async
throws {
    let dataStream = ByteStream.from(data: data)

    let input = PutObjectInput(
        body: dataStream,
```

```
        bucket: bucket,
        key: key
    )
    _ = try await client.putObject(input: input)
}

/// Download the named file to the given directory on the local device.
///
/// - Parameters:
///   - bucket: Name of the bucket that contains the file to be copied.
///   - key: The name of the file to copy from the bucket.
///   - to: The path of the directory on the local device where you want to
///         download the file.
public func downloadFile(bucket: String, key: String, to: String) async throws
{
    let fileUrl = URL(fileURLWithPath: to).appendingPathComponent(key)

    let input = GetObjectInput(
        bucket: bucket,
        key: key
    )
    let output = try await client.getObject(input: input)

    // Get the data stream object. Return immediately if there isn't one.
    guard let body = output.body else {
        return
    }
    let data = body.toBytes().toData()
    try data.write(to: fileUrl)
}

/// Read the specified file from the given S3 bucket into a Swift
/// `Data` object.
///
/// - Parameters:
///   - bucket: Name of the bucket containing the file to read.
///   - key: Name of the file within the bucket to read.
///
/// - Returns: A `Data` object containing the complete file data.
public func readFile(bucket: String, key: String) async throws -> Data {
    let input = GetObjectInput(
        bucket: bucket,
        key: key
    )
    let output = try await client.getObject(input: input)

    // Get the stream and return its contents in a `Data` object. If
    // there is no stream, return an empty `Data` object instead.
    guard let body = output.body else {
        return "".data(using: .utf8)!
    }
    let data = body.toBytes().toData()
    return data
}

/// Copy a file from one bucket to another.
///
/// - Parameters:
///   - sourceBucket: Name of the bucket containing the source file.
///   - name: Name of the source file.
///   - destBucket: Name of the bucket to copy the file into.
public func copyFile(from sourceBucket: String, name: String, to destBucket:
String) async throws {
    let srcUrl = ("\"(sourceBucket)/
\(name)").addingPercentEncoding(withAllowedCharacters: .urlPathAllowed)
```

```
let input = CopyObjectInput(  
    bucket: destBucket,  
    copySource: srcUrl,  
    key: name  
)  
_ = try await client.copyObject(input: input)  
  
/// Deletes the specified file from Amazon S3.  
///  
/// - Parameters:  
///   - bucket: Name of the bucket containing the file to delete.  
///   - key: Name of the file to delete.  
///  
public func deleteFile(bucket: String, key: String) async throws {  
    let input = DeleteObjectInput(  
        bucket: bucket,  
        key: key  
)  
  
    do {  
        _ = try await client.deleteObject(input: input)  
    } catch {  
        throw error  
    }  
}  
  
/// Returns an array of strings, each naming one file in the  
/// specified bucket.  
///  
/// - Parameter bucket: Name of the bucket to get a file listing for.  
/// - Returns: An array of `String` objects, each giving the name of  
///           one file contained in the bucket.  
public func listBucketFiles(bucket: String) async throws -> [String] {  
    let input = ListObjectsV2Input(  
        bucket: bucket  
)  
    let output = try await client.listObjectsV2(input: input)  
    var names: [String] = []  
  
    guard let objList = output.contents else {  
        return []  
    }  
  
    for obj in objList {  
        if let objName = obj.key {  
            names.append(objName)  
        }  
    }  
  
    return names  
}
```

Um programa de linha de comando Swift para gerenciar as chamadas do SDK.

```
import Foundation  
import ServiceHandler  
import ArgumentParser  
  
/// The command-line arguments and options available for this  
/// example command.  
struct ExampleCommand: ParsableCommand {  
    @Argument(help: "Name of the S3 bucket to create")
```

```
var bucketName: String

@Argument(help: "Pathname of the file to upload to the S3 bucket")
var uploadSource: String

@Argument(help: "The name (key) to give the file in the S3 bucket")
var objName: String

@Argument(help: "S3 bucket to copy the object to")
var destBucket: String

@Argument(help: "Directory where you want to download the file from the S3
bucket")
var downloadDir: String

static var configuration = CommandConfiguration(
    commandName: "s3-basics",
    abstract: "Demonstrates a series of basic AWS S3 functions.",
    discussion: """
    Performs the following Amazon S3 commands:

    * `CreateBucket`
    * `PutObject`
    * `GetObject`
    * `CopyObject`
    * `ListObjects`
    * `DeleteObjects`
    * `DeleteBucket`
"""

)
/// Called by ``main()`` to do the actual running of the AWS
/// example.
func runAsync() async throws {
    let serviceHandler = await ServiceHandler()

    // 1. Create the bucket.
    print("Creating the bucket \(bucketName)...")
    try await serviceHandler.createBucket(name: bucketName)

    // 2. Upload a file to the bucket.
    print("Uploading the file \(uploadSource)...")
    try await serviceHandler.uploadFile(bucket: bucketName, key: objName, file:
uploadSource)

    // 3. Download the file.
    print("Downloading the file \(objName) to \(downloadDir)...")
    try await serviceHandler.downloadFile(bucket: bucketName, key: objName, to:
downloadDir)

    // 4. Copy the file to another bucket.
    print("Copying the file to the bucket \(destBucket)...")
    try await serviceHandler.copyFile(from: bucketName, name: objName, to:
destBucket)

    // 5. List the contents of the bucket.

    print("Getting a list of the files in the bucket \(bucketName)")
    let fileList = try await serviceHandler.listBucketFiles(bucket: bucketName)
    let numFiles = fileList.count
    if numFiles != 0 {
        print("\(numFiles) file\(((numFiles > 1) ? "s" : "") in bucket
\(bucketName):")
        for name in fileList {
            print("  \(name)")
        }
    }
}
```

```
        } else {
            print("No files found in bucket \$(bucketName)")
        }

        // 6. Delete the objects from the bucket.

        print("Deleting the file \$(objName) from the bucket \$(bucketName)...")
        try await serviceHandler.deleteFile(bucket: bucketName, key: objName)
        print("Deleting the file \$(objName) from the bucket \$(destBucket)...")
        try await serviceHandler.deleteFile(bucket: destBucket, key: objName)

        // 7. Delete the bucket.
        print("Deleting the bucket \$(bucketName)...")
        try await serviceHandler.deleteBucket(name: bucketName)

        print("Done.")
    }
}

// Main program entry point.
//
@main
struct Main {
    static func main() async {
        let args = Array(CommandLine.arguments.dropFirst())

        do {
            let command = try ExampleCommand.parse(args)
            try await command.runAsync()
        } catch {
            ExampleCommand.exit(withError: error)
        }
    }
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API do AWS SDK para Swift.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjects](#)
 - [PutObject](#)

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Gerenciar objetos do Amazon S3 versionados em lotes com uma função do Lambda usando um AWS SDK

O exemplo de código a seguir mostra como gerenciar objetos do S3 versionados em lotes com uma função do Lambda.

Python

SDK para Python (Boto3).

Mostra como manipular objetos do Amazon Simple Storage Service (Amazon S3) versionados em lotes criando trabalhos que chamam funções do AWS Lambda para executar o processamento. Este exemplo cria um bucket habilitado para versão, carrega as estrofes do poema Você é velho, padre William, de Lewis Carroll, e usa trabalhos em lote do Amazon S3 para distorcer o poema de várias maneiras.

Aprenda como:

- Criar funções do Lambda que operam em objetos versionados.
- Criar um manifesto de objetos para atualizar.
- Cria trabalhos em lote que invocam funções do Lambda para atualizar objetos.
- Excluir funções do Lambda
- Esvaziar e excluir um bucket versionado.

Este exemplo é mais bem visualizado no GitHub. Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- Amazon S3

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar um gerenciador de transferências para carregar e baixar arquivos de e para o Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como usar um gerenciador de transferências para carregar e baixar arquivos de e para o S3.

Para obter mais informações, consulte [Carregar um objeto usando carregamento fracionado](#).

.NET

AWS SDK for .NET

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Chame funções que transferem arquivos de e para um bucket do S3 usando o Amazon S3 TransferUtility.

```
global using System.Text;
global using Amazon;
global using Amazon.S3;
global using Amazon.S3.Model;
global using Amazon.S3.Transfer;
global using TransferUtilityBasics;
```

```
// This Amazon S3 client uses the default user credentials
// defined for this computer.
IAmazonS3 client = new AmazonS3Client();
var transferUtil = new TransferUtility(client);

// Change the following values to an Amazon S3 bucket that
// exists in your AWS account.
var bucketName = "doc-example-bucket1";
var localPath =
    $"{Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData)}\TransferFolder";

DisplayInstructions();

PressEnter();

// Upload a single file to an S3 bucket.
var fileToUpload = "UploadTest.docx";

Console.WriteLine($"Uploading {fileToUpload} to the S3 bucket, {bucketName}.");

var success = await TransferMethods.UploadSingleFileAsync(transferUtil, bucketName,
    fileToUpload, localPath);
if (success)
{
    Console.WriteLine($"Successfully uploaded the file, {fileToUpload} to
{bucketName}.");
}

PressEnter();

// Upload a local directory to an S3 bucket.
var keyPrefix = "UploadFolder";
var uploadPath = $"{localPath}\UploadFolder";

Console.WriteLine($"Uploading the files in {uploadPath} to {bucketName}");
Console.WriteLine($"{uploadPath} contains the following files:");
DisplayLocalFiles(uploadPath);
Console.WriteLine();

success = await TransferMethods.UploadFullDirectoryAsync(transferUtil, bucketName,
    keyPrefix, uploadPath);
if (success)
{
    Console.WriteLine($"Successfully uploaded the files in {uploadPath} to
{bucketName}");
    Console.WriteLine($"{bucketName} currently contains the following files:");
    await DisplayBucketFiles(client, bucketName, keyPrefix);
    Console.WriteLine();
}

PressEnter();

// Download a single file from an S3 bucket.
var keyName = "FileToDelete.docx";

Console.WriteLine($"Downloading {keyName} from {bucketName}.");

success = await TransferMethods.DownloadSingleFileAsync(transferUtil, bucketName,
    keyName, localPath);
if (success)
{
    Console.WriteLine($"Successfully downloaded the file, {keyName} from
{bucketName}");
}
```

```
PressEnter();

// Download the contents of a directory from an S3 bucket.
var s3Path = "DownloadFolder";
var downloadPath = $"{localPath}\\\DownloadFolder";

Console.WriteLine($"Downloading the contents of {bucketName}\\\{s3Path}");
Console.WriteLine($"{bucketName}\\\{s3Path} contains the following files:");
await DisplayBucketFiles(client, bucketName, s3Path);
Console.WriteLine();

success = await TransferMethods.DownloadS3DirectoryAsync(transferUtil, bucketName,
    s3Path, downloadPath);
if (success)
{
    Console.WriteLine($"Downloaded the files in {bucketName} to {downloadPath}.");
    Console.WriteLine($"{downloadPath} now contains the following files:");
    DisplayLocalFiles(downloadPath);
}

Console.WriteLine("\nThe TransferUtility Basics application has completed.");
PressEnter();

static void DisplayInstructions()
{
    var sepBar = new string('-', 80);

    Console.Clear();
    Console.WriteLine(sepBar);
    Console.WriteLine(CenterText("Amazon S3 Transfer Utility Basics"));
    Console.WriteLine(sepBar);
    Console.WriteLine("This program shows how to use the Amazon S3 Transfer
Utility.");
    Console.WriteLine("It performs the following actions:");
    Console.WriteLine("\t1. Upload a single object to an S3 bucket.");
    Console.WriteLine("\t2. Upload all an entire directory from the local computer
to an\n\t S3 bucket.");
    Console.WriteLine("\t3. Download a single object from an S3 bucket.");
    Console.WriteLine("\t4. Download the objects in an S3 bucket to a local
directory.");
    Console.WriteLine($"{sepBar}");
}

static void PressEnter()
{
    Console.WriteLine("Press <Enter> to continue.");
    _ = Console.ReadLine();
    Console.WriteLine("\n");
}

static string CenterText(string textToCenter)
{
    var centeredText = new StringBuilder();
    centeredText.Append(new string(' ', (int)(80 - textToCenter.Length) / 2));
    centeredText.Append(textToCenter);
    return centeredText.ToString();
}

static void DisplayLocalFiles(string localPath)
{
    var fileList = Directory.GetFiles(localPath);
    if (fileList is not null)
    {
        foreach (var fileName in fileList)
        {
```

```
        Console.WriteLine(fileName);
    }
}

static async Task DisplayBucketFiles(IAmazonS3 client, string bucketName, string
s3Path)
{
    ListObjectsV2Request request = new()
    {
        BucketName = bucketName,
        Prefix = s3Path,
        MaxKeys = 5,
    };

    var response = new ListObjectsV2Response();

    do
    {
        response = await client.ListObjectsV2Async(request);

        response.S3Objects
            .ForEach(obj => Console.WriteLine($"{obj.Key}"));

        // If the response is truncated, set the request ContinuationToken
        // from the NextContinuationToken property of the response.
        request.ContinuationToken = response.NextContinuationToken;
    } while (response.IsTruncated);
}
```

Carregue um único arquivo.

```
public static async Task<bool> UploadSingleFileAsync(
    TransferUtility transferUtil,
    string bucketName,
    string fileName,
    string localPath)
{
    if (File.Exists($"{localPath}\\{fileName}"))
    {
        try
        {
            await transferUtil.UploadAsync(new TransferUtilityUploadRequest
            {
                BucketName = bucketName,
                Key = fileName,
                FilePath = $"{localPath}\\{fileName}",
            });

            return true;
        }
        catch (AmazonS3Exception s3Ex)
        {
            Console.WriteLine($"Could not upload {fileName} from
{localPath} because:");
            Console.WriteLine(s3Ex.Message);
            return false;
        }
    }
    else
    {
        Console.WriteLine($"{fileName} does not exist in {localPath}");
        return false;
    }
}
```

```
    }  
}
```

Carregue um diretório local inteiro.

```
public static async Task<bool> UploadFullDirectoryAsync(  
    TransferUtility transferUtil,  
    string bucketName,  
    string keyPrefix,  
    string localPath)  
{  
    if (Directory.Exists(localPath))  
    {  
        try  
        {  
            await transferUtil.UploadDirectoryAsync(new  
TransferUtilityUploadDirectoryRequest  
            {  
                BucketName = bucketName,  
                KeyPrefix = keyPrefix,  
                Directory = localPath,  
            });  
  
            return true;  
        }  
        catch (AmazonS3Exception s3Ex)  
        {  
            Console.WriteLine($"Can't upload the contents of {localPath}  
because:");  
            Console.WriteLine(s3Ex?.Message);  
            return false;  
        }  
    }  
    else  
    {  
        Console.WriteLine($"The directory {localPath} does not exist.");  
        return false;  
    }  
}
```

Baixe um único arquivo.

```
public static async Task<bool> DownloadSingleFileAsync(  
    TransferUtility transferUtil,  
    string bucketName,  
    string keyName,  
    string localPath)  
{  
    await transferUtil.DownloadAsync(new TransferUtilityDownloadRequest  
    {  
        BucketName = bucketName,  
        Key = keyName,  
        FilePath = $"{localPath}\\{keyName}",  
    });  
  
    if (File.Exists($"{localPath}\\{keyName}"))  
    {  
        return true;  
    }  
    else
```

```
{  
    return false;  
}  
}
```

Baixe o conteúdo de um bucket do S3.

```
public static async Task<bool> DownloadS3DirectoryAsync(  
    TransferUtility transferUtil,  
    string bucketName,  
    string s3Path,  
    string localPath)  
{  
    await transferUtil.DownloadDirectoryAsync(new  
TransferUtilityDownloadDirectoryRequest  
    {  
        BucketName = bucketName,  
        LocalDirectory = localPath,  
        S3Directory = s3Path,  
    });  
  
    if (Directory.Exists(localPath))  
    {  
        return true;  
    }  
    else  
    {  
        return false;  
    }  
}
```

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie funções que transfiram arquivos usando várias configurações do gerenciador de transferências disponíveis. Use uma classe de retorno de chamada para gravar o andamento do retorno de chamada durante a transferência de arquivos.

```
import sys  
import threading  
  
import boto3  
from boto3.s3.transfer import TransferConfig  
  
MB = 1024 * 1024  
s3 = boto3.resource('s3')  
  
class TransferCallback:  
    """  
    Handle callbacks from the transfer manager.  
    """
```

```
The transfer manager periodically calls the __call__ method throughout
the upload and download process so that it can take action, such as
displaying progress to the user and collecting data about the transfer.
"""

def __init__(self, target_size):
    self._target_size = target_size
    self._total_transferred = 0
    self._lock = threading.Lock()
    self.thread_info = {}

def __call__(self, bytes_transferred):
    """
    The callback method that is called by the transfer manager.

    Display progress during file transfer and collect per-thread transfer
    data. This method can be called by multiple threads, so shared instance
    data is protected by a thread lock.
    """
    thread = threading.current_thread()
    with self._lock:
        self._total_transferred += bytes_transferred
        if thread.ident not in self.thread_info.keys():
            self.thread_info[thread.ident] = bytes_transferred
        else:
            self.thread_info[thread.ident] += bytes_transferred

        target = self._target_size * MB
        sys.stdout.write(
            f"\r{self._total_transferred} of {target} transferred "
            f"({(self._total_transferred / target) * 100:.2f}%.)")
        sys.stdout.flush()

def upload_with_default_configuration(local_file_path, bucket_name,
                                      object_key, file_size_mb):
    """
    Upload a file from a local folder to an Amazon S3 bucket, using the default
    configuration.
    """
    transfer_callback = TransferCallback(file_size_mb)
    s3.Bucket(bucket_name).upload_file(
        local_file_path,
        object_key,
        Callback=transfer_callback)
    return transfer_callback.thread_info

def upload_with_chunksize_and_meta(local_file_path, bucket_name, object_key,
                                   file_size_mb, metadata=None):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart chunk size and adding metadata to the Amazon S3 object.

    The multipart chunk size controls the size of the chunks of data that are
    sent in the request. A smaller chunk size typically results in the transfer
    manager using more threads for the upload.

    The metadata is a set of key-value pairs that are stored with the object
    in Amazon S3.
    """
    transfer_callback = TransferCallback(file_size_mb)

    config = TransferConfig(multipart_chunksize=1 * MB)
    extra_args = {'Metadata': metadata} if metadata else None
    s3.Bucket(bucket_name).upload_file(
```

```
local_file_path,
object_key,
Config=config,
ExtraArgs=extra_args,
Callback=transfer_callback)
return transfer_callback.thread_info

def upload_with_high_threshold(local_file_path, bucket_name, object_key,
                                file_size_mb):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart threshold larger than the size of the file.

    Setting a multipart threshold larger than the size of the file results
    in the transfer manager sending the file as a standard upload instead of
    a multipart upload.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
    s3.Bucket(bucket_name).upload_file(
        local_file_path,
        object_key,
        Config=config,
        Callback=transfer_callback)
    return transfer_callback.thread_info

def upload_with_sse(local_file_path, bucket_name, object_key,
                    file_size_mb, sse_key=None):
    """
    Upload a file from a local folder to an Amazon S3 bucket, adding server-side
    encryption with customer-provided encryption keys to the object.

    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is
    provided in the download request.
    """
    transfer_callback = TransferCallback(file_size_mb)
    if sse_key:
        extra_args = {
            'SSECustomerAlgorithm': 'AES256',
            'SSECustomerKey': sse_key}
    else:
        extra_args = None
    s3.Bucket(bucket_name).upload_file(
        local_file_path,
        object_key,
        ExtraArgs=extra_args,
        Callback=transfer_callback)
    return transfer_callback.thread_info

def download_with_default_configuration(bucket_name, object_key,
                                         download_file_path, file_size_mb):
    """
    Download a file from an Amazon S3 bucket to a local folder, using the
    default configuration.
    """
    transfer_callback = TransferCallback(file_size_mb)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path,
        Callback=transfer_callback)
    return transfer_callback.thread_info
```

```
def download_with_single_thread(bucket_name, object_key,
                                 download_file_path, file_size_mb):
    """
    Download a file from an Amazon S3 bucket to a local folder, using a
    single thread.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(use_threads=False)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path,
        Config=config,
        Callback=transfer_callback)
    return transfer_callback.thread_info

def download_with_high_threshold(bucket_name, object_key,
                                  download_file_path, file_size_mb):
    """
    Download a file from an Amazon S3 bucket to a local folder, setting a
    multipart threshold larger than the size of the file.

    Setting a multipart threshold larger than the size of the file results
    in the transfer manager sending the file as a standard download instead
    of a multipart download.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path,
        Config=config,
        Callback=transfer_callback)
    return transfer_callback.thread_info

def download_with_sse(bucket_name, object_key, download_file_path,
                      file_size_mb, sse_key):
    """
    Download a file from an Amazon S3 bucket to a local folder, adding a
    customer-provided encryption key to the request.

    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is
    provided in the download request.
    """
    transfer_callback = TransferCallback(file_size_mb)

    if sse_key:
        extra_args = {
            'SSECustomerAlgorithm': 'AES256',
            'SSECustomerKey': sse_key}
    else:
        extra_args = None
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path,
        ExtraArgs=extra_args,
        Callback=transfer_callback)
    return transfer_callback.thread_info
```

Demonstre as funções do gerenciador de transferências e relate resultados.

```
import hashlib
import os
import platform
import shutil
```

```
import time

import boto3
from boto3.s3.transfer import TransferConfig
from botocore.exceptions import ClientError
from botocore.exceptions import ParamValidationError
from botocore.exceptions import NoCredentialsError

import file_transfer

MB = 1024 * 1024
# These configuration attributes affect both uploads and downloads.
CONFIG_ATTRS = ('multipart_threshold', 'multipart_chunksize', 'max_concurrency',
                 'use_threads')
# These configuration attributes affect only downloads.
DOWNLOAD_CONFIG_ATTRS = ('max_io_queue', 'io_chunksize', 'num_download_attempts')

class TransferDemoManager:
    """
    Manages the demonstration. Collects user input from a command line, reports
    transfer results, maintains a list of artifacts created during the
    demonstration, and cleans them up after the demonstration is completed.
    """

    def __init__(self):
        self._s3 = boto3.resource('s3')
        self._chore_list = []
        self._create_file_cmd = None
        self._size_multiplier = 0
        self.file_size_mb = 30
        self.demo_folder = None
        self.demo_bucket = None
        self._setup_platform_specific()
        self._terminal_width = shutil.get_terminal_size(fallback=(80, 80))[0]

    def collect_user_info(self):
        """
        Collect local folder and Amazon S3 bucket name from the user. These
        locations are used to store files during the demonstration.
        """
        while not self.demo_folder:
            self.demo_folder = input(
                "Which file folder do you want to use to store "
                "demonstration files? ")
            if not os.path.isdir(self.demo_folder):
                print(f"{self.demo_folder} isn't a folder!")
                self.demo_folder = None

        while not self.demo_bucket:
            self.demo_bucket = input(
                "Which Amazon S3 bucket do you want to use to store "
                "demonstration files? ")
            try:
                self._s3.meta.client.head_bucket(Bucket=self.demo_bucket)
            except ParamValidationError as err:
                print(err)
                self.demo_bucket = None
            except ClientError as err:
                print(err)
                print(
                    f"Either {self.demo_bucket} doesn't exist or you don't "
                    f"have access to it.")
                self.demo_bucket = None

    def demo(self, question, upload_func, download_func,
```

```
        upload_args=None, download_args=None):
    """Run a demonstration.

    Ask the user if they want to run this specific demonstration.
    If they say yes, create a file on the local path, upload it
    using the specified upload function, then download it using the
    specified download function.
    """
    if download_args is None:
        download_args = {}
    if upload_args is None:
        upload_args = {}
    question = question.format(self.file_size_mb)
    answer = input(f"{question} (y/n)")
    if answer.lower() == 'y':
        local_file_path, object_key, download_file_path = \
            self._create_demo_file()

        file_transfer.TransferConfig = \
            self._config_wrapper(TransferConfig, CONFIG_ATTRS)
        self._report_transfer_params('Uploading', local_file_path,
                                      object_key, **upload_args)
        start_time = time.perf_counter()
        thread_info = upload_func(local_file_path, self.demo_bucket,
                                  object_key, self.file_size_mb,
                                  **upload_args)
        end_time = time.perf_counter()
        self._report_transfer_result(thread_info, end_time - start_time)

        file_transfer.TransferConfig = \
            self._config_wrapper(TransferConfig,
                                CONFIG_ATTRS + DOWNLOAD_CONFIG_ATTRS)
        self._report_transfer_params('Downloading', object_key,
                                     download_file_path, **download_args)
        start_time = time.perf_counter()
        thread_info = download_func(self.demo_bucket, object_key,
                                    download_file_path, self.file_size_mb,
                                    **download_args)
        end_time = time.perf_counter()
        self._report_transfer_result(thread_info, end_time - start_time)

    def last_name_set(self):
        """Get the name set used for the last demo."""
        return self._chore_list[-1]

    def cleanup(self):
        """
        Remove files from the demo folder, and uploaded objects from the
        Amazon S3 bucket.
        """
        print('-' * self._terminal_width)
        for local_file_path, s3_object_key, downloaded_file_path \
            in self._chore_list:
            print(f"Removing {local_file_path}")
            try:
                os.remove(local_file_path)
            except FileNotFoundError as err:
                print(err)

            print(f"Removing {downloaded_file_path}")
            try:
                os.remove(downloaded_file_path)
            except FileNotFoundError as err:
                print(err)

        if self.demo_bucket:
```

```
print(f"Removing {self.demo_bucket}:{s3_object_key}")
try:
    self._s3.Bucket(self.demo_bucket).Object(
        s3_object_key).delete()
except ClientError as err:
    print(err)

def _setup_platform_specific(self):
    """Set up platform-specific command used to create a large file."""
    if platform.system() == "Windows":
        self._create_file_cmd = "fsutil file createnew {} {}"
        self._size_multiplier = MB
    elif platform.system() == "Linux" or platform.system() == "Darwin":
        self._create_file_cmd = f"dd if=/dev/urandom of={{}} " \
            f"bs={MB} count={{}}"
        self._size_multiplier = 1
    else:
        raise EnvironmentError(
            f"Demo of platform {platform.system()} isn't supported.")

def _create_demo_file(self):
    """
    Create a file in the demo folder specified by the user. Store the local
    path, object name, and download path for later cleanup.

    Only the local file is created by this method. The Amazon S3 object and
    download file are created later during the demonstration.

    Returns:
    A tuple that contains the local file path, object name, and download
    file path.
    """
    file_name_template = "TestFile{}-{}.demo"
    local_suffix = "local"
    object_suffix = "s3object"
    download_suffix = "downloaded"
    file_tag = len(self._chore_list) + 1

    local_file_path = os.path.join(
        self.demo_folder,
        file_name_template.format(file_tag, local_suffix))

    s3_object_key = file_name_template.format(file_tag, object_suffix)

    downloaded_file_path = os.path.join(
        self.demo_folder,
        file_name_template.format(file_tag, download_suffix))

    filled_cmd = self._create_file_cmd.format(
        local_file_path,
        self.file_size_mb * self._size_multiplier)

    print(f"Creating file of size {self.file_size_mb} MB "
          f"in {self.demo_folder} by running:")
    print(f"'{:4}{filled_cmd}'")
    os.system(filled_cmd)

    chore = (local_file_path, s3_object_key, downloaded_file_path)
    self._chore_list.append(chore)
    return chore

def _report_transfer_params(self, verb, source_name, dest_name, **kwargs):
    """Report configuration and extra arguments used for a file transfer."""
    print('-' * self._terminal_width)
    print(f'{verb} {source_name} ({self.file_size_mb} MB) to {dest_name}')
    if kwargs:
```

```
        print('With extra args:')
        for arg, value in kwargs.items():
            print(f'{"":4}{arg:<20}: {value}')

    @staticmethod
    def ask_user(question):
        """
        Ask the user a yes or no question.

        Returns:
        True when the user answers 'y' or 'Y'; otherwise, False.
        """
        answer = input(f'{question} (y/n) ')
        return answer.lower() == 'y'

    @staticmethod
    def _config_wrapper(func, config_attrs):
        def wrapper(*args, **kwargs):
            config = func(*args, **kwargs)
            print('With configuration:')
            for attr in config_attrs:
                print(f'{"":4}{attr:<20}: {getattr(config, attr)}')
            return config

        return wrapper

    @staticmethod
    def _report_transfer_result(thread_info, elapsed):
        """Report the result of a transfer, including per-thread data."""
        print(f"\nUsed {len(thread_info)} threads.")
        for ident, byte_count in thread_info.items():
            print(f'{"":4}Thread {ident} copied {byte_count} bytes.')
        print(f"Your transfer took {elapsed:.2f} seconds.")

def main():
    """
    Run the demonstration script for s3_file_transfer.
    """
    demo_manager = TransferDemoManager()
    demo_manager.collect_user_info()

    # Upload and download with default configuration. Because the file is 30 MB
    # and the default multipart_threshold is 8 MB, both upload and download are
    # multipart transfers.
    demo_manager.demo(
        "Do you want to upload and download a {} MB file "
        "using the default configuration?",
        file_transfer.upload_with_default_configuration,
        file_transfer.download_with_default_configuration)

    # Upload and download with multipart_threshold set higher than the size of
    # the file. This causes the transfer manager to use standard transfers
    # instead of multipart transfers.
    demo_manager.demo(
        "Do you want to upload and download a {} MB file "
        "as a standard (not multipart) transfer?",
        file_transfer.upload_with_high_threshold,
        file_transfer.download_with_high_threshold)

    # Upload with specific chunk size and additional metadata.
    # Download with a single thread.
    demo_manager.demo(
        "Do you want to upload a {} MB file with a smaller chunk size and "
        "then download the same file using a single thread?",
        file_transfer.upload_with_chunksizes_and_meta,
```

```
file_transfer.download_with_single_thread,
upload_args={
    'metadata': {
        'upload_type': 'chunky',
        'favorite_color': 'aqua',
        'size': 'medium'}})

# Upload using server-side encryption with customer-provided
# encryption keys.
# Generate a 256-bit key from a passphrase.
sse_key = hashlib.sha256('demo_passphrase'.encode('utf-8')).digest()
demo_manager.demo(
    "Do you want to upload and download a {} MB file using "
    "server-side encryption?",
    file_transfer.upload_with_sse,
    file_transfer.download_with_sse,
    upload_args={'sse_key': sse_key},
    download_args={'sse_key': sse_key})

# Download without specifying an encryption key to show that the
# encryption key must be included to download an encrypted object.
if demo_manager.ask_user("Do you want to try to download the encrypted "
    "object without sending the required key?"):
    try:
        _, object_key, download_file_path = \
            demo_manager.last_name_set()
        file_transfer.download_with_default_configuration(
            demo_manager.demo_bucket, object_key, download_file_path,
            demo_manager.file_size_mb)
    except ClientError as err:
        print("Got expected error when trying to download an encrypted "
            "object without specifying encryption info:")
        print(f"'{err}'")

# Remove all created and downloaded files, remove all objects from
# S3 storage.
if demo_manager.ask_user(
    "Demonstration complete. Do you want to remove local files "
    "and S3 objects?"):
    demo_manager.cleanup()

if __name__ == '__main__':
    try:
        main()
    except NoCredentialsError as error:
        print(error)
    print("To run this example, you must have valid credentials in "
        "a shared credential file or set in environment variables.")
```

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Trabalhar com objetos versionados do Amazon S3 usando um AWS SDK

O código de exemplo a seguir mostra como:

- Crie um bucket do S3 versionado.

- Obtenha todas as versões de um objeto.
- Reverta um objeto para uma versão anterior.
- Exclua e restaure um objeto versionado.
- Exclua, permanentemente, todas as versões de um objeto.

Python

SDK para Python (Boto3).

Tip

Para saber mais sobre como configurar e executar esse exemplo, consulte o [GitHub](#).

Crie funções que envolvam ações do S3.

```
def create_versioned_bucket(bucket_name, prefix):
    """
    Creates an Amazon S3 bucket, enables it for versioning, and configures a
    lifecycle
    that expires noncurrent object versions after 7 days.

    Adding a lifecycle configuration to a versioned bucket is a best practice.
    It helps prevent objects in the bucket from accumulating a large number of
    noncurrent versions, which can slow down request performance.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket_name: The name of the bucket to create.
    :param prefix: Identifies which objects are automatically expired under the
                   configured lifecycle rules.
    :return: The newly created bucket.
    """
    try:
        bucket = s3.create_bucket(
            Bucket=bucket_name,
            CreateBucketConfiguration={
                'LocationConstraint': s3.meta.client.meta.region_name
            }
        )
        logger.info("Created bucket %s.", bucket.name)
    except ClientError as error:
        if error.response['Error']['Code'] == 'BucketAlreadyOwnedByYou':
            logger.warning("Bucket %s already exists! Using it.", bucket_name)
            bucket = s3.Bucket(bucket_name)
        else:
            logger.exception("Couldn't create bucket %s.", bucket_name)
            raise

    try:
        bucket.Versioning().enable()
        logger.info("Enabled versioning on bucket %s.", bucket.name)
    except ClientError:
        logger.exception("Couldn't enable versioning on bucket %s.", bucket.name)
        raise

    try:
        expiration = 7
        bucket.LifecycleConfiguration().put(
            LifecycleConfiguration={
                'Rules': [
                    {
                        'Status': 'Enabled',
                        'Prefix': prefix,

```

```
        'NoncurrentVersionExpiration': {'NoncurrentDays': expiration}
    }]
}
)
logger.info("Configured lifecycle to expire noncurrent versions after %s
days "
            "on bucket %s.", expiration, bucket.name)
except ClientError as error:
    logger.warning("Couldn't configure lifecycle on bucket %s because %s. "
                   "Continuing anyway.", bucket.name, error)

return bucket

def rollback_object(bucket, object_key, version_id):
    """
    Rolls back an object to an earlier version by deleting all versions that
    occurred after the specified rollback version.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket: The bucket that holds the object to roll back.
    :param object_key: The object to roll back.
    :param version_id: The version ID to roll back to.
    """
    # Versions must be sorted by last_modified date because delete markers are
    # at the end of the list even when they are interspersed in time.
    versions = sorted(bucket.object_versions.filter(Prefix=object_key),
                      key=attrgetter('last_modified'), reverse=True)

    logger.debug(
        "Got versions:\n%s",
        '\n'.join([f"\t{version.version_id}, last modified {version.last_modified}"
                  for version in versions]))

    if version_id in [ver.version_id for ver in versions]:
        print(f"Rolling back to version {version_id}")
        for version in versions:
            if version.version_id != version_id:
                version.delete()
                print(f"Deleted version {version.version_id}")
            else:
                break

        print(f"Active version is now {bucket.Object(object_key).version_id}")
    else:
        raise KeyError(f"{version_id} was not found in the list of versions for "
                      f"{object_key}.")

def revive_object(bucket, object_key):
    """
    Revives a versioned object that was deleted by removing the object's active
    delete marker.
    A versioned object presents as deleted when its latest version is a delete
    marker.
    By removing the delete marker, we make the previous version the latest version
    and the object then presents as *not* deleted.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to revive.
    """
    # Get the latest version for the object.
    response = s3.meta.client.list_object_versions(
```

```
Bucket=bucket.name, Prefix=object_key, MaxKeys=1)

if 'DeleteMarkers' in response:
    latest_version = response['DeleteMarkers'][0]
    if latest_version['IsLatest']:
        logger.info("Object %s was indeed deleted on %s. Let's revive it.",
                    object_key, latest_version['LastModified'])
        obj = bucket.Object(object_key)
        obj.Version(latest_version['VersionId']).delete()
        logger.info("Revived %s, active version is now %s with body '%s'",
                    object_key, obj.version_id, obj.get()['Body'].read())
    else:
        logger.warning("Delete marker is not the latest version for %s!",
                      object_key)
elif 'Versions' in response:
    logger.warning("Got an active version for %s, nothing to do.", object_key)
else:
    logger.error("Couldn't get any version info for %s.", object_key)

def permanently_delete_object(bucket, object_key):
    """
    Permanently deletes a versioned object by deleting all of its versions.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to delete.
    """
    try:
        bucket.object_versions.filter(Prefix=object_key).delete()
        logger.info("Permanently deleted all versions of object %s.", object_key)
    except ClientError:
        logger.exception("Couldn't delete all versions of %s.", object_key)
        raise
```

Carregue uma estrofe de poema para um objeto versionado e realize uma série de ações nela.

```
def usage_demo_single_object(obj_prefix='demo-versioning/'):
    """
    Demonstrates usage of versioned object functions. This demo uploads a stanza
    of a poem and performs a series of revisions, deletions, and revivals on it.

    :param obj_prefix: The prefix to assign to objects created by this demo.
    """
    with open('father_william.txt') as file:
        stanzas = file.read().split('\n\n')

    width = get_terminal_size((80, 20))[0]
    print('-'*width)
    print("Welcome to the usage demonstration of Amazon S3 versioning.")
    print("This demonstration uploads a single stanza of a poem to an Amazon "
          "S3 bucket and then applies various revisions to it.")
    print('-'*width)
    print("Creating a version-enabled bucket for the demo...")
    bucket = create_versioned_bucket('bucket-' + str(uuid.uuid1()), obj_prefix)

    print("\nThe initial version of our stanza:")
    print(stanzas[0])

    # Add the first stanza and revise it a few times.
    print("\nApplying some revisions to the stanza...")
    obj_stanza_1 = bucket.Object(f'{obj_prefix}stanza-1')
    obj_stanza_1.put(Body=bytes(stanzas[0], 'utf-8'))
```

```
obj_stanza_1.put(Body=bytes(stanzas[0].upper(), 'utf-8'))
obj_stanza_1.put(Body=bytes(stanzas[0].lower(), 'utf-8'))
obj_stanza_1.put(Body=bytes(stanzas[0][::-1], 'utf-8'))
print("The latest version of the stanza is now:",
      obj_stanza_1.get()['Body'].read().decode('utf-8'),
      sep='\n')

# Versions are returned in order, most recent first.
obj_stanza_1_versions = bucket.object_versions.filter(Prefix=obj_stanza_1.key)
print(
    "The version data of the stanza revisions:",
    *[f"    {version.version_id}, last modified {version.last_modified}"
      for version in obj_stanza_1_versions],
    sep='\n'
)

# Rollback two versions.
print("\nRolling back two versions...")
rollback_object(bucket, obj_stanza_1.key, list(obj_stanza_1_versions)[2].version_id)
print("The latest version of the stanza:",
      obj_stanza_1.get()['Body'].read().decode('utf-8'),
      sep='\n')

# Delete the stanza
print("\nDeleting the stanza...")
obj_stanza_1.delete()
try:
    obj_stanza_1.get()
except ClientError as error:
    if error.response['Error']['Code'] == 'NoSuchKey':
        print("The stanza is now deleted (as expected).")
    else:
        raise

# Revive the stanza
print("\nRestoring the stanza...")
revive_object(bucket, obj_stanza_1.key)
print("The stanza is restored! The latest version is again:",
      obj_stanza_1.get()['Body'].read().decode('utf-8'),
      sep='\n')

# Permanently delete all versions of the object. This cannot be undone!
print("\nPermanently deleting all versions of the stanza...")
permanently_delete_object(bucket, obj_stanza_1.key)
obj_stanza_1_versions = bucket.object_versions.filter(Prefix=obj_stanza_1.key)
if len(list(obj_stanza_1_versions)) == 0:
    print("The stanza has been permanently deleted and now has no versions.")
else:
    print("Something went wrong. The stanza still exists!")

print(f"\nRemoving {bucket.name}...")
bucket.delete()
print(f"{bucket.name} deleted.")
print("Demo done!")
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API do AWS SDK para Python (Boto3).
 - [CreateBucket](#)
 - [DeleteObject](#)
 - [ListObjectVersions](#)
 - [PutBucketLifecycleConfiguration](#)

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos do Amazon S3 entre serviços usando AWS SDKs

As amostras de aplicações a seguir usam AWS SDKs para combinar o Amazon S3 com outros Serviços da AWS. Cada exemplo inclui um link para o GitHub, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Exemplos

- [Criar uma aplicação Amazon Transcribe \(p. 1599\)](#)
- [Converter texto em fala e de volta em texto usando um AWS SDK \(p. 1600\)](#)
- [Criar um cluster do Amazon EMR de longa duração e executar várias etapas usando um AWS SDK \(p. 1600\)](#)
- [Criar um cluster do Amazon EMR de curta duração e executar uma etapa usando um AWS SDK \(p. 1601\)](#)
- [Criar uma aplicação de exploração do Amazon Textract \(p. 1602\)](#)
- [Detectar EPI em imagens com o Amazon Rekognition usando um AWS SDK \(p. 1603\)](#)
- [Detectar entidades em texto extraído de uma imagem usando um AWS SDK \(p. 1604\)](#)
- [Detectar faces em uma imagem usando um AWS SDK \(p. 1604\)](#)
- [Detectar objetos em imagens com o Amazon Rekognition usando um AWS SDK \(p. 1605\)](#)
- [Detectar pessoas e objetos em um vídeo com o Amazon Rekognition usando um AWS SDK \(p. 1607\)](#)
- [Salvar o EXIF e outras informações de imagem usando um AWS SDK \(p. 1608\)](#)

Criar uma aplicação Amazon Transcribe

O exemplo de código a seguir mostra como usar o Amazon Transcribe para transcrever e exibir gravações de voz no navegador.

JavaScript

SDK para JavaScript V3

Crie uma aplicação que use o Amazon Transcribe para transcrever e exibir gravações de voz no navegador. A aplicação usa dois buckets do Amazon Simple Storage Service (Amazon S3), um para hospedar o código da aplicação e outro para armazenar transcrições. A aplicação usa um grupo de usuários do Amazon Cognito para autenticar seus usuários. Os usuários autenticados têm permissões do AWS Identity and Access Management (IAM) para acessar os serviços da AWS necessários.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Esse exemplo também está disponível no [Guia do desenvolvedor do AWS SDK for JavaScript v3](#).

Serviços usados neste exemplo

- Amazon Cognito Identity
- Amazon S3
- Amazon Transcribe

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Converter texto em fala e de volta em texto usando um AWS SDK

O código de exemplo a seguir mostra como:

- Usar o Amazon Polly para sintetizar um arquivo de entrada de texto simples (UTF-8) para um arquivo de áudio.
- Carregar o arquivo de áudio em um bucket do Amazon Simple Storage Service (Amazon S3).
- Usar o Amazon Transcribe para converter o arquivo de áudio em texto.
- Exibir o texto.

Rust

[SDK for Rust](#)

[Note](#)

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Use o Amazon Polly para sintetizar um arquivo de entrada de texto simples (UTF-8) para um arquivo de áudio, carregar o arquivo de áudio para um bucket do Amazon Simple Storage Service, usar o Amazon Transcribe para converter esse arquivo de áudio em texto e exibir o texto.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- Amazon Polly
- Amazon S3
- Amazon Transcribe

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Criar um cluster do Amazon EMR de longa duração e executar várias etapas usando um AWS SDK

O exemplo de código a seguir mostra como criar um cluster do Amazon EMR de longa duração e executar várias etapas.

Python

[SDK para Python \(Boto3\)](#).

Crie um cluster do Amazon EMR de longa duração que use o Apache Spark para consultar dados históricos de revisão da Amazon com base no [Amazon Customer Reviews Dataset](#) (Conjunto de dados de avaliações de clientes da Amazon). Execute um trabalho que obtenha dados dos produtos mais bem avaliados em categorias específicas que contenham palavras-chave nos

títulos dos produtos. Os resultados do trabalho são gravados em um bucket do Amazon Simple Storage Service (Amazon S3).

- Crie um bucket do Amazon S3 e carregue um script de trabalho.
- Crie perfis do AWS Identity and Access Management (IAM).
- Crie grupos de segurança do Amazon Elastic Compute Cloud (Amazon EC2).
- Criar um cluster de longa duração e executar várias etapas do trabalho.

Este exemplo é mais bem visualizado no GitHub. Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- Amazon EC2
- Amazon EMR
- IAM
- Amazon S3

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Criar um cluster do Amazon EMR de curta duração e executar uma etapa usando um AWS SDK

O exemplo de código a seguir mostra como criar um cluster do Amazon EMR de curta duração que executa uma etapa e termina automaticamente após a conclusão dessa etapa.

Python

SDK para Python (Boto3).

Crie um cluster do Amazon EMR de curta duração que estime o valor de pi usando o Apache Spark para parallelizar um grande número de cálculos. O trabalho grava a saída em logs do Amazon EMR e em um bucket do Amazon Simple Storage Service (Amazon S3). O cluster é encerrado automaticamente após a conclusão do trabalho.

- Crie um bucket do Amazon S3 e carregue um script de trabalho.
- Crie perfis do AWS Identity and Access Management (IAM).
- Crie grupos de segurança do Amazon Elastic Compute Cloud (Amazon EC2).
- Criar um cluster de curta duração e executar uma única etapa do trabalho.

Este exemplo é mais bem visualizado no GitHub. Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- Amazon EC2
- Amazon EMR
- IAM
- Amazon S3

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Criar uma aplicação de exploração do Amazon Textract

Os exemplos de código a seguir mostram como explorar a saída do Amazon Textract por meio de uma aplicação interativa.

JavaScript

SDK para JavaScript V3

Mostra como usar o AWS SDK for JavaScript para criar uma aplicação React que usa o Amazon Textract para extrair dados de uma imagem de documento e exibi-los em uma página da Web interativa. Este exemplo é executado em um navegador da Web e requer uma identidade autenticada do Amazon Cognito como credenciais. Ele usa o Amazon Simple Storage Service (Amazon S3) para armazenamento e, para notificações, pesquisa uma fila do Amazon Simple Queue Service (Amazon SQS) que está inscrita em um tópico do Amazon Simple Notification Service (Amazon SNS).

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- Amazon Cognito Identity
- Amazon S3
- Amazon SNS
- Amazon SQS
- Amazon Textract

Python

SDK para Python (Boto3).

Mostra como usar o AWS SDK for Python (Boto3) com o Amazon Textract para detectar elementos de texto, formulários e tabelas em uma imagem de documento. A imagem de entrada e a saída do Amazon Textract são mostradas em uma aplicação Tkinter que permite explorar os elementos detectados.

- Envie uma imagem de documento para o Amazon Textract e explore a saída dos elementos detectados.
- Envie imagens diretamente para o Amazon Textract ou por meio de um bucket do Amazon Simple Storage Service (Amazon S3).
- Use APIs assíncronas para iniciar um trabalho que publica uma notificação em um tópico do Amazon Simple Notification Service (Amazon SNS) quando o trabalho for concluído.
- Faça uma pesquisa em uma fila do Amazon Simple Queue Service (Amazon SQS) para obter uma mensagem de conclusão do trabalho e exiba os resultados.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- Amazon S3
- Amazon SNS
- Amazon SQS

- Amazon Textract

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Detectar EPI em imagens com o Amazon Rekognition usando um AWS SDK

Os exemplos de código a seguir mostram como construir uma aplicação que usa o Amazon Rekognition para detectar equipamentos de proteção individual (EPI) em imagens.

Java

SDK para Java 2.x

Mostra como criar uma função do AWS Lambda que detecta imagens com equipamento de proteção individual.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- DynamoDB
- Amazon Rekognition
- Amazon S3
- Amazon SES

JavaScript

SDK para JavaScript V3

Mostra como usar o Amazon Rekognition com o AWS SDK for JavaScript a fim de construir uma aplicação para detectar equipamentos de proteção individual (EPI) em imagens localizadas em um bucket do Amazon Simple Storage Service (Amazon S3). A aplicação salva os resultados em uma tabela do Amazon DynamoDB e envia uma notificação por e-mail ao administrador com os resultados usando o Amazon Simple Email Service (Amazon SES).

Aprenda como:

- Criar um usuário não autenticado usando o Amazon Cognito.
- Analisar imagens em busca de EPI usando o Amazon Rekognition.
- Verificar um endereço de e-mail para o Amazon SES.
- Atualizar uma tabela do DynamoDB com resultados.
- Enviar uma notificação por e-mail usando o Amazon SES.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- DynamoDB
- Amazon Rekognition
- Amazon S3

- Amazon SES

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Detectar entidades em texto extraído de uma imagem usando um AWS SDK

O exemplo de código a seguir mostra como usar o Amazon Comprehend para detectar entidades em texto extraído pelo Amazon Textract de uma imagem armazenada no Amazon S3.

Python

SDK para Python (Boto3).

Mostra como usar o AWS SDK for Python (Boto3) em um bloco de anotações Jupyter para detectar entidades em texto extraído de uma imagem. Este exemplo usa o Amazon Textract para extrair texto de uma imagem armazenada no Amazon Simple Storage Service (Amazon S3) e no Amazon Comprehend para detectar entidades no texto extraído.

Este exemplo é um bloco de anotações Jupyter e deve ser executado em um ambiente que possa hospedar blocos de anotações. Para obter instruções sobre como executar o exemplo usando o Amazon SageMaker, consulte as instruções em [TextractAndComprehendNotebook.ipynb](#).

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- Amazon Comprehend
- Amazon S3
- Amazon Textract

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Detectar faces em uma imagem usando um AWS SDK

O código de exemplo a seguir mostra como:

- Salvar uma imagem em um bucket do Amazon Simple Storage Service Amazon S3).
- Usar o Amazon Rekognition (Amazon Rekognition) para detectar detalhes faciais, como faixa etária, sexo e emoções (sorriso etc.).
- Exibir esses detalhes.

Rust

SDK for Rust

Note

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Salve a imagem em um bucket do Amazon Simple Storage Service com um prefixo carregamentos, use o Amazon Rekognition para detectar detalhes faciais, como faixa etária, sexo e emoções (sorriso etc.), e exiba esses detalhes.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- Amazon Rekognition
- Amazon S3

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Detectar objetos em imagens com o Amazon Rekognition usando um AWS SDK

Os exemplos de código a seguir mostram como construir uma aplicação que usa o Amazon Rekognition para detectar objetos por categoria em imagens.

.NET

AWS SDK for .NET

Mostra como usar a API .NET do Amazon Rekognition para construir uma aplicação que usa o Amazon Rekognition para identificar objetos por categoria em imagens localizadas em um bucket do Amazon Simple Storage Service (Amazon S3). A aplicação envia uma notificação por e-mail ao administrador com os resultados usando o Amazon Simple Email Service (Amazon SES).

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- Amazon Rekognition
- Amazon S3
- Amazon SES

Java

SDK para Java 2.x

Mostra como usar a API Java do Amazon Rekognition para construir uma aplicação que usa o Amazon Rekognition para identificar objetos por categoria em imagens localizadas em um bucket do Amazon Simple Storage Service (Amazon S3). A aplicação envia uma notificação por e-mail ao administrador com os resultados usando o Amazon Simple Email Service (Amazon SES).

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- Amazon Rekognition
- Amazon S3

- Amazon SES

JavaScript

SDK para JavaScript V3

Mostra como usar o Amazon Rekognition com o AWS SDK for JavaScript para construir uma aplicação que usa o Amazon Rekognition para identificar objetos por categoria em imagens localizadas em um bucket do Amazon Simple Storage Service (Amazon S3). A aplicação envia uma notificação por e-mail ao administrador com os resultados usando o Amazon Simple Email Service (Amazon SES).

Aprenda como:

- Criar um usuário não autenticado usando o Amazon Cognito.
- Analisar imagens em busca de objetos usando o Amazon Rekognition.
- Verificar um endereço de e-mail para o Amazon SES.
- Enviar uma notificação por e-mail usando o Amazon SES.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- Amazon Rekognition
- Amazon S3
- Amazon SES

Kotlin

SDK para Kotlin

Note

Essa documentação é de pré-lançamento para um recurso em versão de pré-visualização. Está sujeita a alteração.

Mostra como usar a API Kotlin do Amazon Rekognition para construir uma aplicação que usa o Amazon Rekognition para identificar objetos por categoria em imagens localizadas em um bucket do Amazon Simple Storage Service (Amazon S3). A aplicação envia uma notificação por e-mail ao administrador com os resultados usando o Amazon Simple Email Service (Amazon SES).

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- Amazon Rekognition
- Amazon S3
- Amazon SES

Python

SDK para Python (Boto3).

Mostra como usar o AWS SDK for Python (Boto3) para criar uma aplicação Web que permite a você fazer o seguinte:

- Carregar fotos em um bucket do Amazon Simple Storage Service (Amazon S3).
- Usar o Amazon Rekognition para analisar e rotular as fotos.
- Usar o Amazon Simple Email Service (Amazon SES) para enviar relatórios de análise da imagem por e-mail.

Este exemplo contém dois componentes principais: uma página da Web escrita em JavaScript que é construída com o React e um serviço REST escrito em Python que é construído com o Flask-RESTful.

Você pode usar a página da Web do React para:

- Exibir uma lista de imagens que estão armazenadas no bucket do S3.
- Carregar imagens do computador para o bucket do S3.
- Exibir imagens e rótulos que identificam os itens detectados na imagem.
- Obter um relatório de todas as imagens no bucket do S3 e enviar um relatório por e-mail.

A página da Web chama o serviço REST. O serviço envia solicitações à AWS para realizar as seguintes ações:

- Obter e filtrar a lista de imagens no bucket do S3.
- Carregar fotos no bucket do S3.
- Usar o Amazon Rekognition para analisar fotos individuais e obter uma lista dos rótulos que identifiquem os itens detectados nas fotos.
- Analisar todas as fotos no bucket do S3 e usar o Amazon SES para enviar um relatório por e-mail.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- Amazon Rekognition
- Amazon S3
- Amazon SES

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Detectar pessoas e objetos em um vídeo com o Amazon Rekognition usando um AWS SDK

Os exemplos de código a seguir mostram como detectar pessoas e objetos em um vídeo com o Amazon Rekognition.

Java

SDK para Java 2.x

Mostra como usar a API Java do Amazon Rekognition a fim de construir uma aplicação para detectar faces e objetos em vídeos localizados em um bucket do Amazon Simple Storage Service (Amazon S3). A aplicação envia uma notificação por e-mail ao administrador com os resultados usando o Amazon Simple Email Service (Amazon SES).

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- Amazon Rekognition
- Amazon S3
- Amazon SES

JavaScript

SDK para JavaScript V3

Mostra como usar o Amazon Rekognition com o AWS SDK for JavaScript a fim de construir uma aplicação para detectar faces e objetos em vídeos localizados em um bucket do Amazon Simple Storage Service (Amazon S3). A aplicação envia uma notificação por e-mail ao administrador com os resultados usando o Amazon Simple Email Service (Amazon SES).

Aprenda como:

- Criar um usuário não autenticado usando o Amazon Cognito.
- Analisar imagens em busca de EPI usando o Amazon Rekognition.
- Verificar um endereço de e-mail para o Amazon SES.
- Enviar uma notificação por e-mail usando o Amazon SES.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- Amazon Rekognition
- Amazon S3
- Amazon SES

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Salvar o EXIF e outras informações de imagem usando um AWS SDK

O código de exemplo a seguir mostra como:

- Obter informações de EXIF de um arquivo JPG, JPEG ou PNG.
- Carregar o arquivo de imagem em um bucket do Amazon Simple Storage Service (Amazon S3).
- Usar o Amazon Rekognition (Amazon Rekognition) para identificar os três principais atributos (rótulos no Amazon Rekognition) no arquivo.
- Adicionar as informações de EXIF e de rótulo a uma tabela do Amazon DynamoDB (DynamoDB) na região.

Rust

SDK for Rust

Note

Esta documentação destina-se a um SDK na versão de previsualização. O SDK está sujeito a alterações e não deve ser usado em ambientes de produção.

Obtenha informações de EXIF de um arquivo JPG, JPEG ou PNG, carregue o arquivo de imagem para um bucket do Amazon Simple Storage Service, use o Amazon Rekognition para identificar os três principais atributos (rótulos no Amazon Rekognition) no arquivo e adicione as informações de EXIF e de rótulo a uma tabela do Amazon DynamoDB na região.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços usados neste exemplo

- DynamoDB
- Amazon Rekognition
- Amazon S3

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK \(p. 1259\)](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Troubleshooting

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

- [Solução de problemas do Amazon S3 por sintoma \(p. 1610\)](#)
- [Obter IDs de solicitação do Amazon S3 para AWS Support \(p. 1611\)](#)
- [Tópicos relacionados \(p. 1614\)](#)

Para obter outros tópicos de como solucionar problemas e obter suporte, consulte o seguinte:

- [Solução de problemas do CORS \(p. 618\)](#)
- [Tratar erros de REST e SOAP \(p. 1276\)](#)
- [AWS Support Documentação do](#)

Para obter informações sobre como solucionar problemas relacionados a ferramentas de terceiros, consulte [Como obter os IDs de solicitação do Amazon S3](#) nos Fóruns do desenvolvedor da AWS.

Solução de 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 \(p. 1610\)](#)
- [Comportamento inesperado ao acessar buckets definidos com CORS \(p. 1611\)](#)

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 do Amazon S3 Inventory. 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 \(p. 786\)](#).

A equipe do Amazon S3 incentiva os clientes a investigarem aplicações que sobrescrevem repetidamente o mesmo objeto do S3, criando potencialmente milhões de versões desse objeto, para determinar se

a aplicação está funcionando conforme o esperado. Se você tem um caso de uso que exige milhões de versões para um ou mais objetos do S3, entre em contato com a equipe do AWS Support em [AWS Support](#) para discutir seu caso de uso para que possamos ajudar você a determinar a solução ideal para o cenário do seu caso de uso.

Para ajudar a evitar esse problema, considere as seguintes práticas recomendadas:

- Habilite uma política de expiração “NonCurrentVersion” de gerenciamento do ciclo de vida e uma política “ExpiredObjectDeleteMarker” para expirar as versões anteriores de objetos e marcadores de exclusão sem objetos de dados associados no bucket.
- Mantenha sua estrutura de diretórios o mais plana possível e torne cada nome de diretório exclusivo.

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 \(p. 618\)](#).

Obter IDs de solicitação do Amazon S3 para AWS Support

Sempre que você precisar entrar em contato com o AWS Support devido a 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 das solicitações incluindo logs de acesso do S3 e eventos/eventos de dados do CloudTrail.

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 como entrar em contato com o AWS Support, consulte [Entre em contato conosco](#).

Tópicos

- [Usar o HTTP para obter IDs de solicitação \(p. 1611\)](#)
- [Usar um navegador da web para obter IDs de solicitação \(p. 1612\)](#)
- [Uso dos AWS SDKs para obter IDs de solicitação \(p. 1612\)](#)
- [Usar o AWS CLI para obter IDs de solicitação \(p. 1614\)](#)

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 do desenvolvedor da AWS.

Uso dos AWS SDKs para obter IDs de solicitação

As seções a seguir incluem informações para configuração do log usando um 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+ey7N9WgVhSnQ6JEXAMPLEZb7hSQDASK+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 for PHP.

Usar o SDK for 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 [Registro em log detalhado](#) no tópico Registro de AWS SDK para chamadas de Java no Guia do desenvolvedor do AWS SDK for Java.

Usar o AWS SDK for .NET para obter IDs de solicitação

Você pode configurar o log no AWS SDK for .NET usando a ferramenta incorporada de log `System.Diagnostics`. Para obter mais informações, consulte a postagem [Registro em log com o AWS SDK for .NET](#) no Blog do desenvolvedor 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 para Python (Boto3) para obter IDs de solicitação

Com o SDK para Python (Boto3), é possível registrar em log respostas específicas, o que permite capturar apenas os cabeçalhos relevantes. O código a seguir mostra como registrar em log partes da resposta para um arquivo:

```
import logging
import boto3
logging.basicConfig(filename='logfile.txt', level=logging.INFO)
logger = logging.getLogger(__name__)
s3 = boto3.resource('s3')
response = s3.Bucket(bucket_name).Object(object_key).put()
logger.info("HTTPStatusCode: %s", response['ResponseMetadata']['HTTPStatusCode'])
logger.info("RequestId: %s", response['ResponseMetadata']['RequestId'])
logger.info("HostId: %s", response['ResponseMetadata']['HostId'])
logger.info("Date: %s", response['ResponseMetadata']['HTTPHeaders']['date'])
```

Também é possível capturar exceções e registrar em log informações relevantes quando uma exceção é gerada. Para obter detalhes, consulte [Discernir informações úteis de respostas de erro](#) no Guia do desenvolvedor do Boto3.

Além disso, é possível configurar o Boto3 para gerar logs de depuração detalhados usando o seguinte código:

```
import boto3
boto3.set_stream_logger('', logging.DEBUG)
```

Para obter mais informações, consulte [set_stream_logger](#) na Referência do Boto3.

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.

- Como 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)
```

- Como 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 \(p. 618\)](#)
- [Tratar erros de REST e SOAP \(p. 1276\)](#)
- [AWS Support Documentação do](#)

Para obter informações sobre como solucionar problemas relacionados a ferramentas de terceiros, consulte [Como obter os IDs de solicitação do Amazon S3](#) nos Fóruns do desenvolvedor da AWS.

Histórico do documento

- Versão atual da API: 2006-03-01

A tabela a seguir descreve as alterações importantes em cada versão da Referência da API do Amazon Simple Storage Service e do Guia do usuário do Amazon S3. Para receber notificações sobre atualizações dessa documentação, você pode se inscrever em um feed RSS.

update-history-change	update-history-description	update-history-date
AWS Backup para o Amazon S3 (p. 1615)	O AWS Backup é um serviço totalmente gerenciado e baseado em políticas que você pode usar para definir uma política de backup central e proteger seus dados do Amazon S3. Para obter mais informações, consulte Usar o AWS Backup para o Amazon S3 .	18 de fevereiro de 2022
Usar o S3 Batch Replication para replicar objetos existentes (p. 1615)	O S3 Batch Replication oferece uma maneira de replicar objetos que existiam antes que uma configuração de replicação estivesse em vigor. Isso é feito por meio do uso de um trabalho de operações em lote. Isso é diferente da replicação em tempo real, que copia contínua e automaticamente novos objetos nos buckets do Amazon S3. Para obter mais informações, consulte Replicating existing objects with S3 Batch Replication (Replicar objetos existentes com o S3 Batch Replication).	8 de fevereiro de 2022
Novo nome: S3 Glacier Flexible Retrieval (p. 1615)	A classe de armazenamento Glacier foi renomeada para S3 Glacier Flexible Retrieval. Essa alteração não afeta a API.	30 de novembro de 2021
Nova configuração do S3 Object Ownership para desabilitar ACLs (p. 1615)	Adote a configuração aplicada pelo proprietário do bucket em Object Ownership para desabilitar ACLs para o bucket e os objetos contidos nele e assumir a propriedade de cada objeto de seu bucket, simplificando o gerenciamento de acesso para dados armazenados no Amazon S3. Para obter mais informações, consulte Controlar a propriedade de objetos e desabilitar ACLs para seu bucket .	30 de novembro de 2021

Nova classe de armazenamento S3 Intelligent-Tiering (p. 1615)	O S3 Intelligent-Tiering Archive Instant Access é uma classe de armazenamento adicional do S3 Intelligent-Tiering. Para obter mais informações, consulte Como o S3 Intelligent-Tiering funciona .	30 de novembro de 2021
Nova classe de armazenamento S3 Glacier Instant Retrieval (p. 1615)	Agora é possível colocar objetos na classe de armazenamento do S3 Glacier Instant Retrieval. Para obter mais informações sobre essa classe de armazenamento, consulte Usar as classes de armazenamento do Amazon S3 .	30 de novembro de 2021
Pré-visualização do AWS Backup para o Amazon S3 (p. 1615)	O AWS Backup é um serviço totalmente gerenciado e baseado em políticas que você pode usar para definir uma política de backup central e proteger seus dados do Amazon S3. Para obter mais informações, consulte Usar o AWS Backup para o Amazon S3 .	30 de novembro de 2021
AWS Identity and Access Management Access Analyzer para o Amazon S3 (p. 1615)	O IAM Access Analyzer executa verificações de política para validar sua política em relação à gramática das políticas e às práticas recomendadas e do IAM. Para saber mais sobre a validação de políticas usando o IAM Access Analyzer, consulte Validação de políticas do IAM Access Analyzer no Guia do usuário IAM.	30 de novembro de 2021
Novos tipos de evento (p. 1615)	Para ver os novos tipos de evento adicionados às notificações de eventos do Amazon S3, consulte Notificações de eventos do Amazon S3 .	29 de novembro de 2021
Habilitar o Amazon EventBridge em buckets (p. 1615)	Você pode habilitar o EventBridge em buckets do Amazon S3 para enviar eventos ao Amazon EventBridge. Consulte Usar o EventBridge .	29 de novembro de 2021
Novos filtros do S3 Lifecycle (p. 1615)	É possível criar regras de ciclo de vida com base no tamanho do objeto ou especificar quantas versões de objeto desatualizadas deverão ser mantidas. Consulte Exemplos de configuração do S3 Lifecycle .	23 de novembro de 2021

Publicar métricas do Amazon S3 Storage Lens no Amazon CloudWatch (p. 1615)	É possível publicar métricas de uso e atividade do S3 Storage Lens no Amazon CloudWatch para criar uma visão unificada de sua integridade operacional nos painéis do CloudWatch. Também é possível usar os recursos do CloudWatch, como alarmes e ações acionadas, matemática métrica e detecção de anomalias, para monitorar e realizar ações sobre as métricas do S3 Storage Lens. Além disso, as APIs do CloudWatch permitem que aplicações, inclusive provedores de terceiros, acessem as métricas do S3 Storage Lens. Para obter mais informações, consulte Monitorar métricas do S3 Storage Lens no CloudWatch .	22 de novembro de 2021
Pontos de acesso multirregionais (p. 1615)	Você pode usar pontos de acesso multirregionais para criar um endpoint global que as aplicações podem usar para atender a solicitações de buckets do Amazon S3, localizados em várias regiões da AWS. Você pode usar esse ponto de acesso multirregional para rotear dados para o bucket de menor latência. Para obter mais informações sobre os pontos de acesso multirregionais e como usá-los, consulte Pontos de acesso multirregionais no Amazon S3 .	2 de setembro de 2021
O Amazon S3 on Outposts adiciona acesso local direto às aplicações (p. 1615)	Execute suas aplicações fora da nuvem privada virtual (VPC) do AWS Outposts e acesse seus dados do S3 on Outposts. Também é possível acessar objetos do S3 on Outposts diretamente da rede local. Para obter mais informações sobre como configurar endpoints do S3 on Outposts usando endereços IP de propriedade do cliente (ColP) e acessar seus objetos criando um gateway local na rede local, consulte Como acessar o Amazon S3 on Outposts usando pontos de acesso somente para VPC .	29 de julho de 2021

Alias de ponto de acesso do Amazon S (p. 1615)	Quando você cria um ponto de acesso, o Amazon S3 gera automaticamente um alias que poderá ser usado no lugar de um nome de bucket para acesso a dados. É possível usar esse alias de ponto de acesso em vez de um nome do recurso da Amazon (ARN) para qualquer operação de plano de dados do ponto de acesso. Para obter mais informações, consulte Usar um alias em estilo de bucket para seu ponto de acesso .	26 de julho de 2021
O Amazon S3 Inventory e o S3 Batch Operations são compatíveis com status da chave do bucket (p. 1615)	O Amazon S3 Inventory e o Batch Operations oferecem suporte à identificação e à cópia de objetos existentes com chaves de bucket do S3. As chaves de Bucket do S3 aceleram a redução dos custos de criptografia no lado do servidor para objetos existentes. Para obter mais informações, consulte Amazon S3 Inventory e Objeto Copy do Batch Operations .	3 de junho de 2021
Snapshot da conta de métricas do Amazon S3 Storage Lens (p. 1615)	O snapshot da conta do S3 Storage Lens exibe o armazenamento total, a contagem de objetos e o tamanho médio do objeto na página inicial (Buckets) resumindo as métricas do painel padrão. Para obter mais informações, consulte Snapshot da conta de métricas do S3 Storage Lens .	5 de maio de 2021
Aumento do suporte ao endpoint do Amazon S3 no Outposts (p. 1615)	O S3 on Outposts agora suporta até 100 endpoints por Outpost. Para obter mais informações, consulte Restrições de rede do S3 no Outposts .	29 de abril de 2021
Notificações de eventos do Amazon S3 no Outposts usando Amazon CloudWatch Events (p. 1615)	Você pode usar o CloudWatch Events para criar uma regra para capturar qualquer evento da API do S3 no Outposts e ser notificado por meio de todos os destinos compatíveis do CloudWatch. Para obter mais informações, consulte Como receber notificações de eventos do S3 no Outposts usando o CloudWatch Events .	19 de abril de 2021

S3 Object Lambda (p. 1615)	Com o S3 Object Lambda, você pode adicionar seu próprio código às solicitações GET do Amazon S3 para modificar e processar dados, conforme eles são retornados para uma aplicação. Você pode usar o código personalizado para modificar os dados retornados por solicitações S3 GET padrão para filtrar linhas, redimensionar imagens dinamicamente, editar dados confidenciais e muito mais. Para obter mais informações, consulte Transformar objetos .	18 de março de 2021
AWS PrivateLink (p. 1615)	Com o AWS PrivateLink para Amazon S3, você pode se conectar diretamente ao S3 usando um endpoint de interface em sua VPC em vez de se conectar via Internet. Os endpoints de interface são diretamente acessíveis a partir de aplicações on-premises ou em uma diferente Região da AWS. Para obter mais informações, consulte AWS PrivateLink para Amazon S3 .	2 de fevereiro de 2021
Gerenciar a capacidade do Amazon S3 em Outposts com AWS CloudTrail (p. 1615)	Os eventos de gerenciamento do S3 no Outposts estão disponíveis por meio dos logs do CloudTrail. Para obter mais informações, consulte Como gerenciar a capacidade do S3 no Outposts com o CloudTrail .	21 de dezembro de 2020
Forte consistência (p. 1615)	O Amazon S3 oferece uma forte consistência de leitura após gravação para PUTs e DELETEs de objetos no bucket do Amazon S3 em todas as Regiões da AWS. Além disso, as operações de leitura no Amazon S3 Select, Listas de controle de acesso do Amazon S3, Tags de objeto do Amazon S3 e metadados de objeto (por exemplo, objeto HEAD) são muito consistentes. Para obter mais informações, consulte Modelo de consistência de dados do Amazon S3 .	1º de dezembro de 2020

Sincronização de modificação de réplica do Amazon S3 (p. 1615)	A sincronização de modificação de réplica do Amazon S3 mantém metadados de objeto, como tags, ACLs e configurações de bloqueio de objetos em sincronia entre objetos de origem e réplicas. Quando esse recurso está habilitado, o Amazon S3 replica as alterações de metadados feitas no objeto de origem ou nas cópias da réplica. Para obter mais informações, consulte Replicar alterações de metadados com sincronização de modificação de réplica .	1º de dezembro de 2020
Amazon S3 Bucket Keys (p. 1615)	As chaves de bucket do Amazon S3 reduzem o custo da criptografia no lado do servidor do Amazon S3 usando o AWS Key Management Service (SSE-KMS). Essa nova chave no nível de bucket para criptografia no lado do servidor pode reduzir os custos de solicitações do AWS KMS em até 99%, diminuindo o tráfego de solicitações do Amazon S3 para o AWS KMS. Para obter mais informações, consulte Reducir o custo do SSE-KMS usando chaves de bucket do S3 .	1º de dezembro de 2020

Amazon S3 Storage Lens (p. 1615)	O Amazon S3 Storage Lens agrupa suas métricas de uso e atividade e exibe as informações no snapshot da conta na página inicial do console (Buckets) do Amazon S3, em painéis interativos ou por meio de uma exportação de métricas que você pode obter por download no formato CSV ou Parquet. Você pode usar o painel para visualizar insights e tendências, sinalizar discrepâncias e receber recomendações para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados. Você pode usar o S3 Storage Lens por meio do AWS Management Console, da AWS CLI, de AWS SDKs ou da API REST. Para obter mais informações, consulte Avaliar a atividade e o uso do armazenamento com o S3 Storage Lens .	18 de novembro de 2020
Rastreamento de solicitações do S3 usando o AWS X-Ray (p. 1615)	O Amazon S3 se integra ao X-Ray para obter uma cadeia de solicitações que se integra ao X-Ray para propagar o contexto de rastreamento e fornecer uma cadeia de solicitações com nós upstream e downstream . Para obter mais informações, consulte Rastrear solicitações usando o X-Ray .	16 de novembro de 2020
Métricas de replicação do S (p. 1615)	As métricas de replicação do S3 fornecem métricas detalhadas para as regras de replicação na configuração de replicação. Para obter mais informações, consulte Métricas de replicação e notificações de eventos do Amazon S3 .	9 de novembro de 2020
Acesso ao arquivo em S3 Intelligent-Tiering S3 e acesso ao arquivo profundo (p. 1615)	O acesso ao arquivamento e o acesso ao arquivamento profundo do S3 Intelligent-Tiering são camadas de armazenamento adicionais sob o S3 Intelligent-Tiering. Para obter mais informações, consulte Classe de armazenamento para otimizar automaticamente objetos acessados com frequência e pouca frequência .	9 de novembro de 2020

Replicação de marcadores de exclusão (p. 1615)	Com a replicação de marcadores de exclusão, você pode garantir que os marcadores de exclusão sejam copiados para seus buckets de destino para suas regras de replicação. Para obter mais informações, consulte Usando replicação de marcadores de exclusão .	9 de novembro de 2020
Propriedade de objetos do S (p. 1615)	Propriedade de objetos é uma configuração de bucket do S3 que pode ser usada para controlar a propriedade de novos objetos que são carregados nos buckets. Para obter mais informações, consulte Usar o S3 Object Ownership .	2 de outubro de 2020
Amazon S3 on Outposts (p. 1615)	Com o Amazon S3 on Outposts, é possível criar buckets do S3 no AWS Outposts, além de armazenar e recuperar facilmente objetos on-premises para aplicações que exigem acesso a dados locais, processamento de dados local e residência de dados. Você pode usar o S3 on Outposts por meio do AWS Management Console, da AWS CLI, dos AWS SDKs ou da API REST. Para obter mais informações, consulte Uso do Amazon S3 no Outposts .	30 de setembro de 2020
Condição do proprietário do bucket (p. 1615)	É possível usar a condição de proprietário do bucket do Amazon S3 para garantir que os buckets usados nas operações do S3 pertençam às Contas da AWS esperadas. Para obter mais informações, consulte Condição do proprietário do bucket .	11 de setembro de 2020
Suporte a operações em lote do S3 para retenção de bloqueio de objetos (p. 1615)	Agora você pode usar operações em lote com bloqueio de objeto do S3 para aplicar configurações de retenção a muitos objetos do Amazon S3 de uma só vez. Para obter mais informações, consulte Definir datas de retenção do bloqueio de objeto do S3 com o S3 Batch Operations .	4 de maio de 2020

Suporte a operações em lote do S3 para retenção legal de bloqueio de objetos (p. 1615)	Agora você pode usar operações em lote com bloqueio de objeto do S3 para adicionar retenção legal a muitos objetos do Amazon S3 de uma só vez. Para obter mais informações, consulte Usar o S3 Batch Operations para definir a retenção legal do bloqueio de objeto do S3 .	4 de maio de 2020
Tags de trabalho para operações em lote do S (p. 1615)	Você pode adicionar tags aos trabalhos de operações em lote do S3 para controlar e rotular esses trabalhos. Para obter mais informações, consulte Tags para trabalhos do S3 Batch Operations .	16 de março de 2020
Pontos de acesso do Amazon S (p. 1615)	Os pontos de acesso do Amazon S3 simplificam o gerenciamento do acesso a dados em escala para conjuntos de dados compartilhados no S3. Os pontos de acesso são nomeados endpoints de rede anexados a buckets que podem ser usados para executar operações de objeto do S3. Para obter mais informações, consulte Como gerenciar o acesso a dados com pontos de acesso do Amazon S3 .	2 de dezembro de 2019
Analizador de acesso para Amazon S3 (p. 1615)	O Analisador de acesso para Amazon S3 alerta sobre buckets do S3 configurados para permitir o acesso a qualquer pessoa na Internet ou a outras Contas da AWS, incluindo contas fora da organização. Para obter mais informações, consulte Usar o Access Analyzer para Amazon S3 .	2 de dezembro de 2019
Controle do tempo de replicação do S3 (S3 RTC) (p. 1615)	O recurso Controle do tempo de replicação do S3 (S3 RTC) replica a maioria dos objetos que você carrega para o Amazon S3 em segundos, bem como 99,99% desses objetos em 15 minutos. Para obter mais informações, consulte Replicar objetos usando o S3 Replication Time Control (S3 RTC) .	20 de novembro de 2019

Replicação para a mesma região (p. 1615)	A replicação para a mesma região (SRR) é usada para copiar objetos entre buckets do Amazon S3 na mesma Região da AWS. Para obter informações sobre as replicações entre regiões e para a mesma região, consulte Replicação .	18 de setembro de 2019
Suporte da replicação entre regiões ao bloqueio de objetos do S (p. 1615)	A replicação entre regiões agora oferece suporte ao bloqueio de objetos. Para obter mais informações, consulte O que faz o Amazon S3 Replicate? .	28 de maio de 2019
Operações em lote do S (p. 1615)	Usando o recurso Operações em lote do S3, você pode executar operações em lote de grande escala em objetos do Amazon S3. O recurso Operações em lote do S3 pode executar uma única operação em listas de objetos que você especificar. Uma única tarefa pode realizar a operação especificada em bilhões de objetos contendo exabytes de dados. Para obter mais informações, consulte Executar o S3 Batch Operations .	30 de abril de 2019
Região Ásia-Pacífico (Hong Kong) (p. 1615)	Agora o Amazon S3 está disponível na região Ásia-Pacífico (Hong Kong). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte Regiões e endpoints na Referência geral da AWS.	24 de abril de 2019
Adição de um novo campo aos logs de acesso ao servidor (p. 1615)	O Amazon S3 adicionou o seguinte campo aos logs de acesso ao servidor: Transport Layer Security (TLS) version (Versão do Transport Layer Security [TLS]). Para obter mais informações, consulte Formato do log de acesso ao servidor .	28 de março de 2019
Nova classe de armazenamento de arquivos (p. 1615)	Agora o Amazon S3 oferece uma nova classe de armazenamento de arquivos, DEEP_ARCHIVE, para armazenar objetos raramente acessados. Para obter mais informações, consulte Classes de armazenamento	27 de março de 2019

Adição de novos campos aos logs de acesso ao servidor (p. 1615)	O Amazon S3 adicionou os seguintes novos campos aos logs de acesso ao servidor: Host Id (ID do host), Signature Version (Versão do Signature), Cipher Suite (Pacote de criptografia), Authentication Type (Tipo de autenticação) e Host Header (Cabeçalho de host). Para obter mais informações, consulte Formato do log de acesso ao servidor .	5 de março de 2019
Suporte para arquivos do Amazon S3 Inventory no formato Parquet (p. 1615)	O Amazon S3 agora dá suporte ao formato Apache Parquet (Parquet) além dos formatos de arquivo Apache optimized row columnar (ORC) e Comma-Separated Values (CSV – Valores separados por vírgulas) para arquivos de saída de inventário. Para obter mais informações, consulte Inventário .	4 de dezembro de 2018
Bloqueio de objetos do S (p. 1615)	O Amazon S3 agora oferece a funcionalidade do Bloqueio de objetos que fornece proteções de gravação única, leitura múltipla, para objetos do Amazon S3. Para obter mais informações, consulte Bloquear objetos .	26 de novembro de 2018
Restaurar atualização rápida (p. 1615)	Usando a atualização rápida de restauração do Amazon S3, altere a velocidade de uma restauração da classe de armazenamento S3 Glacier Flexible Retrieval para mais rapidez durante o andamento da restauração. Para obter mais informações, consulte Restaurar objetos arquivados .	26 de novembro de 2018
Restaurar notificações de evento (p. 1615)	As notificações de evento do Amazon S3 agora oferecem suporte a eventos de início e conclusão durante a restauração de objetos na classe de armazenamento S3 Glacier Flexible Retrieval. Para obter mais informações, consulte Notificações de eventos .	26 de novembro de 2018

PUT diretamente para a classe de armazenamento S3 Glacier Flexible Retrieval (p. 1615)	A operação PUT do Amazon S3 agora oferece suporte à especificação do S3 Glacier Flexible Retrieval como a classe de armazenamento durante a criação de objetos. Anteriormente, era necessário fazer a transição de objetos para a classe de armazenamento S3 Glacier Flexible Retrieval de outra classe de armazenamento do Amazon S3. Além disso, ao usar a replicação entre regiões (CRR) do S3, você já especifica S3 Glacier Flexible Retrieval como a classe de armazenamento para objetos replicados. Para obter mais informações sobre a classe de armazenamento S3 Glacier Flexible Retrieval, consulte Classes de armazenamento . Para obter mais informações sobre como especificar a classe de armazenamento para objetos replicados, consulte Visão geral da configuração de replicação . Para obter mais informações sobre as alterações de PUT diretamente para API REST do S3 Glacier Flexible Retrieval, consulte Histórico do documento: PUT diretamente para S3 Glacier Flexible Retrieval .	26 de novembro de 2018
Nova classe de armazenamento (p. 1615)	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 Classes de armazenamento	26 de novembro de 2018
Bloqueio de acesso público do Amazon S (p. 1615)	O Amazon S3 agora inclui a possibilidade de bloquear acesso público a buckets e objetos por bucket ou conta. Para obter mais informações, consulte Usar o Amazon S3 Block Public Access .	15 de novembro de 2018

Filtrar melhorias nas regras de replicação entre regiões (CRR) (p. 1615)	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 Configuração da CRR: visão geral da configuração da replicação .	19 de setembro de 2018
Novos recursos do Amazon S3 Select (p. 1615)	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>).	5 de setembro de 2018
Atualizações agora disponíveis em RSS (p. 1615)	Agora você pode assinar um RSS Feed para receber notificações sobre atualizações no Guia do usuário do Amazon S3.	19 de junho de 2018

Atualizações anteriores

A tabela a seguir descreve as alterações importantes em cada versão do Guia do usuário do Amazon S3 antes de 19 de junho de 2018.

Alteração	Descrição	Data
Atualização de exemplos de código	Exemplos de código atualizados: <ul style="list-style-type: none">C#: atualizamos todos os exemplos para usar o padrão assíncrono baseado em tarefas. Para obter mais informações, consulte APIs assíncronas de Amazon Web Services para .NET no Guia do desenvolvedor do AWS SDK for .NET. Agora os exemplos de código são compatíveis com a versão 3 do AWS SDK for .NET.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 Criar clientes de serviço.PHP: atualização de todos os exemplos para usar o SDK para AWS SDK for PHP 3.0. Para obter mais informações sobre o AWS SDK for PHP 3.0, consulte AWS SDK for PHP.	30 de abril de 2018

Alteração	Descrição	Data
	<ul style="list-style-type: none"> Ruby: atualização do código de exemplo para que os exemplos funcionem com o AWS SDK for Ruby versão 3. 	
O Amazon S3 agora relata classes de armazenamento S3 Glacier Flexible Retrieval e ONEZONE_IA para 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 S3 Glacier Flexible Retrieval):</p> <ul style="list-style-type: none"> 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 Uso de classes de armazenamento do Amazon S3 (p. 731). Para objetos da classe de armazenamento S3 Glacier Flexible Retrieval, as métricas de armazenamento relatam as seguintes sobrecargas: <ul style="list-style-type: none"> Uma sobrecarga de 32 KB por objeto, cobrada de acordo com o preço da classe de armazenamento S3 Glacier Flexible Retrieval Uma sobrecarga de 8 KB por objeto, cobrada de acordo com a definição de preço da classe de armazenamento STANDARD <p>Para obter mais informações, consulte Transição de objetos usando o Amazon S3 Lifecycle (p. 748).</p> <p>Para obter mais informações sobre métricas de armazenamento, consulte Monitoramento de métricas com o Amazon CloudWatch (p. 1067).</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 Uso de classes de armazenamento do Amazon S3 (p. 731) .	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 Filtragem e recuperação de dados usando o Amazon S3 Select (p. 908) .	4 de abril de 2018
Região Ásia-Pacífico (Osaka-Local)	<p>O Amazon S3 agora está disponível na região Ásia-Pacífico (Osaka-Local). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte Regiões e endpoints na Referência geral da AWS.</p> <p>Important</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 seu representante de vendas.</p>	12 de fevereiro de 2018

Alteração	Descrição	Data
Carimbo de data e hora da criação do Amazon S3 Inventory	O Amazon S3 Inventory agora conta com um carimbo de data e hora com a data e a hora de início da criação do relatório do Amazon S3 Inventory. Você pode usar o timestamp 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 Europa (Paris)	O Amazon S3 já está disponível na região Europa (Paris). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte Regiões e endpoints na Referência geral da AWS.	18 de dezembro de 2017
Região da China (Ningxia)	Agora o Amazon S3 está disponível na região da China (Ningxia). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte Regiões e endpoints na Referência geral da AWS.	11 de dezembro de 2017
Consulta de arquivos com SQL	O Amazon S3 agora oferece suporte à consulta de arquivos de dados do S3 Glacier com SQL. Para obter mais informações, consulte Consultar objetos arquivados (p. 719) .	29 de novembro de 2017
Suporte para arquivos do Amazon S3 Inventory no formato ORC	Agora, o Amazon S3 é compatível com arquivos de saída de inventário no formato colunar de linhas otimizado do Apache (ORC) e no formato de valores separados por vírgulas (CSV). Além disso, agora você pode consultar o inventário do Amazon S3 usando o SQL padrão usando o Amazon Athena, o Amazon Redshift Spectrum e outras ferramentas como Presto , Apache Hive e Apache Spark . Para obter mais informações, consulte Inventário do Amazon S3 (p. 786) .	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 chaves gerenciadas pelo Amazon S3 (SSE-S3) ou chaves gerenciadas pelo AWS KMS (SSE-KMS). Para obter mais informações, consulte Definir o comportamento padrão da criptografia para os buckets do Amazon S3 (p. 142) .	06 de novembro de 2017
Status de criptografia no Amazon S3 Inventory	Agora, o Amazon S3 é compatível e inclui o status da criptografia no Amazon S3 Inventory 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 Amazon S3 Inventory 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 Inventário do Amazon S3 (p. 786) .	06 de novembro de 2017

Alteração	Descrição	Data
Melhorias da replicação entre regiões (CRR)	<p>Agora, a replicação entre regiões oferece suporte ao seguinte:</p> <ul style="list-style-type: none"> 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 Alterar o proprietário da réplica (p. 864). Por padrão, o Amazon S3 não replica objetos no seu bucket de origem que foram criados usando a criptografia no lado do servidor usando chaves armazenadas no AWS KMS. Mas, na sua configuração de CRR, você pode orientar o Amazon S3 a replicar esses objetos. Para obter mais informações, consulte Replicação de objetos criados com criptografia no lado do servidor (SSE) usando chaves do KMS (p. 867). 	06 de novembro de 2017
Região Europa (Londres)	O Amazon S3 já está disponível na região Europa (Londres). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte Regiões e endpoints na Referência geral da AWS.	13 de dezembro de 2016
Região do Canadá (Central)	O Amazon S3 já está disponível na região Canadá (Central). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte Regiões e endpoints na Referência geral da AWS.	8 de dezembro de 2016
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"> 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). 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. 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. Você também pode personalizar métricas do CloudWatch e eventos do CloudTrail para exibir informações por filtros de tags específicos. <p>Para obter mais informações, consulte Categorizando seu armazenamento usando tags (p. 878).</p>	29 de novembro de 2016

Alteração	Descrição	Data
Agora, o Amazon S3 Lifecycle oferece suporte a filtros baseados em etiquetas	Agora, o Amazon S3 oferece suporte para filtragem com base em tag na configuração de ciclo de vida. Você já pode especificar regras do ciclo de vida nas quais é possível determinar um prefixo de chave, uma ou mais tags de objeto ou uma combinação das duas para selecionar um subconjunto de objetos ao qual a regra do ciclo de vida se aplica. Para obter mais informações, consulte Gerenciando seu ciclo de vida de armazenamento (p. 746) .	29 de novembro de 2016
Métricas de solicitação do CloudWatch para buckets	O Amazon S3 agora oferece suporte a 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 Monitoramento de métricas com o Amazon CloudWatch (p. 1067) .	29 de novembro de 2016
Inventário do Amazon S3	O Amazon S3 agora oferece suporte para inventário de armazenamento. O Amazon S3 Inventory 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 comum). Para obter mais informações, consulte Inventário do Amazon S3 (p. 786) .	29 de novembro de 2016
Análise do Amazon S3: análise de classe de armazenamento	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. Para obter mais informações, consulte Análise do Amazon S3 – Análise de classe de armazenamento (p. 1116) o Guia do usuário do Amazon S3.	29 de novembro de 2016
Novas recuperações de dados expressas e em massa ao restaurar objetos arquivados no S3 Glacier	O Amazon S3 agora oferece suporte para recuperações de dados expressas e em massa, além de recuperações padrão ao restaurar objetos arquivados no S3 Glacier. Para obter mais informações, consulte Restaurar um objeto arquivado (p. 715) .	21 de novembro de 2016

Alteração	Descrição	Data
Registro de objetos do CloudTrail	O CloudTrail oferece suporte ao registro em log 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 Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail (p. 1023) .	21 de novembro de 2016
US East (Ohio) Region	O Amazon S3 já está disponível na região Leste dos EUA (Ohio). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte Regiões e endpoints na Referência geral da AWS.	17 de outubro de 2016
Suporte a IPv6 para o Amazon S3 Transfer Acceleration	O Amazon S3 agora oferece suporte ao Protocolo de Internet versão 6 (IPv6) para o Amazon S3 Transfer Acceleration. Você pode se conectar ao Amazon S3 por IPv6 usando o novo endpoint de pilha dupla para Transfer Acceleration. Para obter mais informações, consulte Conceitos básicos do Amazon S3 Transfer Acceleration (p. 149) .	6 de outubro de 2016
Suporte a IPv6	O Amazon S3 agora oferece suporte ao Protocolo de Internet versão 6 (IPv6). Você pode acessar o Amazon S3 por IPv6 usando endpoints de pilha dupla. Para obter mais informações, consulte Fazer solicitações para o Amazon S3 por meio do IPv6 (p. 1213) .	11 de agosto de 2016
Asia Pacific (Mumbai) Region	O Amazon S3 já está disponível na região da Ásia-Pacífico (Mumbai). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte Regiões e endpoints na Referência geral da AWS.	27 de junho de 2016
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 locais da borda do Amazon CloudFront distribuídos globalmente. Para obter mais informações, consulte Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration (p. 147) .	19 de abril de 2016
Suporte de ciclo de vida para remover os marcadores de exclusão de objeto expirado	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 expirado em um bucket com versões. Para obter mais informações, consulte Elementos para descrever ações de ciclo de vida (p. 771) .	16 de março de 2016

Alteração	Descrição	Data
A configuração de ciclo de vida de bucket agora oferece suporte para parar 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 parar 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 interrupção, o Amazon S3 exclui todas as partes carregadas e para o multipart upload.</p> <p>Para obter informações conceituais, consulte os seguintes tópicos no Guia do usuário do Amazon S3:</p> <ul style="list-style-type: none"> • Abortar um multipart upload (p. 206) • Elementos para descrever ações de ciclo de vida (p. 771) <p>As seguintes operações de API foram atualizadas para oferecer suporte à nova ação:</p> <ul style="list-style-type: none"> • Ciclo de vida de PUT Bucket: a configuração de XML agora permite que você especifique a ação <code>AbortIncompleteMultipartUpload</code> em uma regra de configuração de ciclo de vida. • Listar partes e Iniciar o multipart upload: 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 interrupção e qual regra de ciclo de vida é aplicável. 	16 de março de 2016
Asia Pacific (Seoul) Region	O Amazon S3 já está disponível na região Ásia-Pacífico (Seul). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte Regiões e endpoints na Referência geral da AWS.	6 de janeiro de 2016
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 Exemplos de chave de condição do Amazon S3 (p. 447).</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 API de multipart upload e permissões (p. 183).</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)". Essa é só uma atualização de nome de região, não há nenhuma alteração na funcionalidade.	11 de dezembro de 2015

Alteração	Descrição	Data
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 Uso de classes de armazenamento do Amazon S3 (p. 731).</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 Gerenciando seu ciclo de vida de armazenamento (p. 746).</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 Replicação de objetos (p. 801).</p>	16 de setembro de 2015
AWS CloudTrailIntegração do	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 Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail (p. 1023) .	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 do bucket, consulte cotas de serviço da AWS na Referência geral da AWS. Para obter mais informações, consulte Uso da SDKs AWS (p. 131) e Restrições e limitações do bucket (p. 159) .	4 de agosto de 2015
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 Regiões (p. 6) .	4 de agosto de 2015

Alteração	Descrição	Data
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 Notificações de eventos do Amazon S3 (p. 1083) .	28 de julho de 2015
Integração do Amazon CloudWatch	A nova integração do Amazon CloudWatch permite monitorar e definir alarmes de uso do Amazon S3 por meio de 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 Monitoramento de métricas com o Amazon CloudWatch (p. 1067) .	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 Esvaziar um bucket (p. 137) .	16 de julho de 2015
Políticas de bucket para endpoints da Amazon VPC	O Amazon S3 adicionou suporte para políticas de bucket para VPC endpoints. Você pode usar políticas de bucket do S3 para controlar o acesso a buckets de VPC endpoints específicos 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 Controlar o acesso a partir de VPC endpoints com políticas de bucket (p. 517) .	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 Notificações de eventos do Amazon S3 (p. 1083) .	9 de abril de 2015
Replicação entre regiões	O Amazon S3 já oferece suporte à 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 Replicação de objetos (p. 801) .	24 de março de 2015
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 Notificações de eventos do Amazon S3 (p. 1083) .	13 de novembro de 2014

Alteração	Descrição	Data
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 KMS. 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 criptografia no lado do servidor com AWS KMS, consulte Proteção de dados usando criptografia no lado do servidor com o AWS Key Management Service.</p>	12 de novembro de 2014
Europe (Frankfurt) Region	O Amazon S3 já está disponível na região Europa (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 Criptografia do servidor (com chaves de criptografia fornecidas pelo cliente).</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 Gerenciando seu ciclo de vida de armazenamento (p. 746) .	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 Identity and Access Management no Amazon S3 (p. 419) .	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 Registrar em log as solicitações com registro em log de acesso ao servidor (p. 1041) .	26 de novembro de 2013
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 usar a API REST ou os AWS SDKs.	20 de setembro de 2013

Alteração	Descrição	Data
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, consulte Visão geral das variáveis de política do IAM no Guia do usuário do IAM.</p> <p>Para obter exemplos de variáveis de políticas no Amazon S3, consulte Exemplos de política de usuário (p. 545).</p>	3 de abril de 2013
Suporte do console para Pagamento pelo solicitante	<p>Agora você pode configurar seu bucket para pagamentos pelo solicitante usando o console do Amazon S3. Para obter mais informações, consulte Configuração de buckets de Pagamento pelo solicitante para transferências de armazenamento e uso (p. 155).</p>	31 de dezembro de 2012
Suporte de domínio raiz para hospedagem de sites	<p>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 Hospedagem de um site estático usando o Amazon S3 (p. 1188).</p>	27 de dezembro de 2012
Revisão do console	<p>O console do Amazon S3 foi atualizado. Os tópicos de documentação que se referem ao console foram revisados conforme necessário.</p>	14 de dezembro de 2012
Suporte para arquivamento de dados no S3 Glacier	<p>Agora, o Amazon S3 oferece suporte a uma opção de armazenamento que permite utilizar o serviço de armazenamento de baixo custo do S3 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 S3 Glacier. Você pode definir regras facilmente em um bucket usando o console do Amazon S3 ou usando programaticamente a API do Amazon S3 ou AWS SDKs.</p> <p>Para obter mais informações, consulte Gerenciando seu ciclo de vida de armazenamento (p. 746).</p>	13 de novembro de 2012

Alteração	Descrição	Data
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 (Opcional) Configurar um redirecionamento de uma página da Web (p. 1203).</p> <p>Para obter informações sobre a hospedagem de sites, consulte Hospedagem de um site estático usando o Amazon S3 (p. 1188).</p>	4 de outubro de 2012
Suporte para compartilhamento de recursos de origem cruzada (CORS)	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 aplicações Web cliente avançados no Amazon S3 e permitir seletivamente o acesso de domínio cruzado aos recursos do Amazon S3. Para obter mais informações, consulte Usar o compartilhamento de recursos de origem cruzada (CORS) (p. 607) .	31 de agosto de 2012
Suporte para tags de alocação de custos	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 Usar tags de alocação de custos para buckets do S3 (p. 888) .	21 de agosto de 2012
Suporte para acesso à API protegido por MFA nas políticas de bucket	<p>O Amazon S3 agora oferece suporte para acesso à API protegido por MFA, 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 AWS Multi-Factor Authentication. 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 Adicionar uma política de bucket para exigir MFA (p. 524).</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
Nova região compatível	O Amazon S3 agora oferece suporte à região da América do Sul (São Paulo). Para obter mais informações, consulte Métodos de acesso a um bucket (p. 135) .	14 de dezembro de 2011

Alteração	Descrição	Data
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 Excluir objetos do Amazon S3 (p. 237) .	7 de dezembro de 2011
Nova região compatível	O Amazon S3 agora oferece suporte à região do Oeste dos EUA (Oregon). Para obter mais informações, consulte Buckets e regiões (p. 135) .	8 de novembro de 2011
Atualização da documentação	Correções de erros de documentação.	8 de novembro de 2011
Atualização da documentação	Além de correções de erros de documentação, esta versão inclui os seguintes aprimoramentos: <ul style="list-style-type: none"> Novas seções de criptografia do lado do servidor usando o AWS SDK for PHP e o AWS SDK for Ruby (consulte Especificação de criptografia do Amazon S3 (p. 380)). Nova seção sobre como criar e testar amostras de Ruby (consulte Usar o AWS SDK for Ruby - versão 3 (p. 1269)). 	17 de outubro de 2011
Suporte para criptografia do lado do servidor	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 Management Console. Para saber mais sobre a criptografia de dados, acesse Usar criptografia de dados .	4 de outubro de 2011
Atualização da documentação	Além de correções de erros de documentação, esta versão inclui os seguintes aprimoramentos: <ul style="list-style-type: none"> Adição de exemplos de Ruby e PHP à seção Fazer solicitações (p. 1211). Adição de seções que descrevem como gerar e usar pre-signed URLs. Para obter mais informações, consulte Compartilhar objetos usando URLs pré-assinados (p. 275) e Compartilhar objetos usando URLs pré-assinados (p. 275). Atualização de uma seção existente para apresentar o AWS Explorer para Eclipse e Visual Studio. Para obter mais informações, consulte Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers (p. 1258). 	22 de setembro de 2011

Alteração	Descrição	Data
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 AWS SDK 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 Fazer solicitações (p. 1211).</p> <p>Para obter mais informações sobre o suporte do IAM para credenciais de segurança temporárias, consulte Credenciais de segurança temporárias no Guia do usuário do IAM.</p>	3 de agosto de 2011
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, Upload Part (Copy). Você pode usar essa operação do multipart upload para copiar objetos com até 5 TB. Para obter mais informações, consulte Cópia de objetos (p. 215).</p> <p>Para obter informações conceituais sobre a API do multipart upload, consulte Carregar e copiar objetos usando multipart upload (p. 180).</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. Ou seja, 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 Criar uma função para delegar permissões a um usuário do IAM no Guia do usuário do IAM.</p> <p>Para obter mais informações sobre como especificar principais em uma política de bucket, consulte Principais (p. 440).</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 Referência geral da AWS .	1 de março de 2011

Alteração	Descrição	Data
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, http://mywebsite.com/subfolder) 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 Hospedagem de um site estático usando o Amazon S3 (p. 1188) .	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 Fazer download de um objeto (p. 223) .	14 de janeiro de 2011
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 Carregar e copiar objetos usando multipart upload (p. 180) .	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 Carregar e copiar objetos usando multipart upload (p. 180) .	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 Políticas de bucket e políticas de usuário (p. 437)	17 de setembro de 2010
O Amazon S3 funciona com o IAM	Esse serviço agora é integrado ao AWS Identity and Access Management (IAM). Para obter mais informações, acesse Serviços da AWS que funcionam com o IAM no Guia do usuário do IAM.	2 de setembro de 2010
Notificações	O recurso de notificações do Amazon S3 permite configurar um bucket para que o Amazon S3 publique uma mensagem em um tópico do Amazon Simple Notification Service (Amazon SNS) quando o Amazon S3 detecta um evento-chave em um bucket. Para obter mais informações, consulte Configuração de notificação de eventos de bucket (p. 1083) .	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 Políticas de bucket e políticas de usuário (p. 437) .	6 de julho de 2010

Alteração	Descrição	Data
Sintaxe de estilo 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 Hospedagem virtual (p. 1249) .	9 de junho de 2010
Novo endpoint para a Europa (Irlanda)	O Amazon S3 agora fornece um endpoint para a Europa (Irlanda): http://s3-eu-west-1.amazonaws.com .	9 de junho de 2010
Console	Agora você pode usar o Amazon S3 por meio do AWS Management Console. Você pode ler sobre toda a funcionalidade do Amazon S3 no console no Manual do usuário do Amazon Simple Storage Service.	9 de junho de 2010
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 Reduced Redundancy Storage (RRS) (p. 1) .	12 de maio de 2010
Nova região compatível	O Amazon S3 agora oferece suporte à região Ásia-Pacífico (Cingapura). Para obter mais informações, consulte Buckets e regiões (p. 135) .	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 Versionamento (p. 5) e Usar versionamento (p. 678) .	8 de fevereiro de 2010
Nova região compatível	O Amazon S3 agora oferece suporte à 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 Buckets e regiões (p. 135) .	2 de dezembro de 2009
AWS SDK for .NET	AWSA 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 Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers (p. 1258) .	11 de novembro de 2009

Glossário da AWS

Para obter a terminologia mais recente da AWS, consulte o [Glossário da AWS](#) na Referência geral da AWS.