



Guia do usuário

Amazon Simple Storage Service



Versão da API 2006-03-01

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Simple Storage Service: Guia do usuário

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

O que é o Amazon S3?	1
Recursos do Amazon S3	1
Classes de armazenamento	1
Gerenciamento de armazenamento	2
Gerenciamento de acesso e segurança	3
Processamento de dados	4
Registro e monitoramento do armazenamento	4
Análise e insights	5
Consistência forte	6
Como funciona o Amazon S3	6
Buckets	7
Objetos	7
Chaves	8
Versionamento do S3	8
ID da versão	8
Política de bucket	9
Pontos de acesso do S3	9
Listas de controle de acesso (ACLs)	10
Regiões	10
Modelo de consistência de dados do Amazon S3	11
Aplicações simultâneas	12
Serviços relacionados	13
Acesso ao Amazon S3	14
AWS Management Console	14
AWS Command Line Interface	14
SDKs da AWS	15
API REST do Amazon S3	15
Pagar pelo Amazon S3	16
Conformidade do PCI DSS	16
Introdução	18
Configuração	19
Cadastre-se em uma Conta da AWS	19
Criar um usuário com acesso administrativo	20
Etapa 1: Criar um bucket	21

Etapa 2: Fazer upload de um objeto	28
Etapa 3: fazer download de um objeto	29
Uso do console do S3	29
Etapa 4: copiar um objeto	30
Etapa 5: excluir os objetos e o bucket	31
Excluir um objeto	32
Esvaziar o bucket	32
Excluir bucket	33
Próximas etapas	33
Entender casos de uso comuns	34
Controlar o acesso a buckets e objetos	35
Gerencie e monitore seu armazenamento	36
Desenvolvimento com o Amazon S3	36
Aprenda com os tutoriais	37
Explore o treinamento e o suporte	39
Controle de acesso	40
Criar um bucket	40
Armazenar e compartilhar dados	43
Compartilhar recursos	44
Proteger dados	44
Tutoriais	49
Conceitos básicos	38
Otimizar custos do armazenamento	38
Gerenciar o armazenamento	38
Hospedar vídeos e sites	38
Processar dados	38
Proteger dados	39
Como transformar dados com o S3 Object Lambda	50
Pré-requisitos	52
Etapa 1: criar um bucket do S3	54
Etapa 2: fazer upload do arquivo para seu bucket do S3	55
Etapa 3: criar um ponto de acesso do S3	56
Etapa 4: Criar uma função Lambda	57
Etapa 5: configurar uma política do IAM para a função de execução da função Lambda	63
Etapa 6: criar um ponto de acesso do S3 Object Lambda	64
Etapa 7: exibir os dados transformados	66

Etapa 8: limpar	69
Próximas etapas	72
Detecção e edição de dados PII	73
Pré-requisitos: criar um usuário do IAM com permissões	74
Etapa 1: criar um bucket do S3	77
Etapa 2: fazer upload do arquivo para seu bucket do S3	78
Etapa 3: criar um ponto de acesso do S3	79
Etapa 4: configurar e implantar uma função Lambda pré-construída	80
Etapa 5: criar um ponto de acesso do S3 Object Lambda	81
Etapa 6: usar o ponto de acesso do S3 Object Lambda para recuperar o arquivo editado	83
Etapa 7: limpar	84
Próximas etapas	88
Hospedagem de streaming de vídeo	89
Pré-requisitos: registrar e configurar um domínio personalizado com o Route 53	91
Etapa 1: criar um bucket do S3	92
Etapa 2: carregar um vídeo no bucket do S3	93
Etapa 3: Criar uma identidade do acesso de origem do CloudFront	94
Etapa 4: criar uma distribuição do CloudFront	94
Etapa 5: Acessar o vídeo por meio da distribuição do CloudFront	97
Etapa 6: Configurar sua distribuição do CloudFront para usar o seu nome de domínio personalizado	98
Etapa 7: acessar o vídeo do S3 por meio da distribuição do CloudFront com o nome de domínio personalizado	103
(Opcional) Etapa 8: exibir dados sobre solicitações recebidas pela distribuição do CloudFront	104
Etapa 9: limpeza	104
Próximas etapas	109
Vídeos de transcodificação em lote	110
Pré-requisitos	112
Etapa 1: Criar um bucket do S3 para os arquivos de mídia de saída	112
Etapa 2: criar uma função do IAM para MediaConvert	114
Etapa 3: criar uma função do IAM para a função Lambda	115
Etapa 4: Criar uma função do Lambda para transcodificação de vídeo	118
Etapa 5: Configurar o inventário do Amazon S3 para seu bucket de origem do S3	135
Etapa 6: criar uma função do IAM para operações em lote do S3	140
Etapa 7: criar e executar um trabalho de operações em lote do S3	143

Etapa 8: Conferir os arquivos de mídia de saída do bucket de destino do S3	149
Etapa 9: limpar	149
Próximas etapas	152
Configurar um site estático	153
Etapa 1: Criar um bucket	154
Etapa 2: Habilitar hospedagem de site estático	154
Etapa 3: editar as configurações do Bloqueio de acesso público	156
Etapa 4: Adicionar política de bucket que torna o conteúdo do bucket publicamente disponível	158
Etapa 5: Configurar um documento de índice	159
Etapa 6: configurar um documento de erros	160
Etapa 7: testar o endpoint do site	161
Etapa 8: Limpar	162
Configurar um site estático usando um domínio personalizado	162
Antes de começar	164
Etapa 1: Registrar um domínio personalizado no Route 53	164
Etapa 2: Criar dois buckets	164
Etapa 3: Configurar o bucket de domínio raiz	166
Etapa 4: Configurar o bucket de subdomínio para redirecionamento	167
Etapa 5: Configurar o registro em log	168
Etapa 6: Fazer upload do conteúdo do site e do índice	169
Etapa 7: carregar um documento de erros	170
Etapa 8: Editar o bloqueio de acesso público	171
Etapa 9: Anexar uma política de bucket	173
Etapa 10: Testar o endpoint de domínio	175
Etapa 11: Adicionar registros de alias	176
Etapa 12: Testar o site	181
Acelerar seu site com o Amazon CloudFront	181
Limpar recursos de exemplo	186
Trabalhar com buckets	189
Visão geral dos buckets	190
Sobre permissões	191
Gerenciar o acesso público aos buckets	192
Configuração do bucket	193
Regras de nomeação	197
Regras de nomenclatura de buckets de uso geral	197

Regras de nomenclatura de buckets de diretório	199
Acessar e listar um bucket	200
.....	200
Listar buckets	202
Criação de um bucket	203
Visualizar propriedades de buckets	216
Esvaziar um bucket	218
Esvaziar um bucket com o AWS CloudTrail configurado	221
Excluir um bucket	221
Definir criptografia de bucket padrão	227
Usar criptografia SSE-KMS para operações entre contas	229
Como usar a criptografia padrão com a replicação	230
Usar Chaves de bucket do Amazon S3 com criptografia padrão	230
Configurar a criptografia padrão	231
Monitorar a criptografia padrão	237
Mountpoint para Amazon S3	238
Instalar o Mountpoint	239
Configurar e usar o Mountpoint	244
Configurar o Transfer Acceleration	247
Por que usar o Transfer Acceleration?	248
Requisitos para usar o Transfer Acceleration	248
Conceitos básicos	250
Habilitar o Transfer Acceleration	252
Ferramenta de comparação de velocidade	259
Uso de Pagamento pelo solicitante	260
Como funcionam as cobranças de Pagamento pelo solicitante	261
Configurar Pagamento pelo solicitante	262
Recuperação da configuração de requestPayment	264
Fazer download de objetos de buckets de Pagamento pelo solicitante	265
Restrições e limitações	266
Trabalhar com objetos	268
Objetos	269
Sub-recursos	270
Criar chaves de objeto	271
Diretrizes de nomeação de chave de objeto	272
Trabalhar com metadados	276

Metadados do objeto definidos pelo sistema	276
Metadados do objeto definidos pelo usuário	280
Editar os metadados do objeto	281
Fazer upload de objetos	284
Usar multipart upload	298
Processo multipart upload	299
Somadas de verificação com operações de carregamento fracionado	301
Operações simultâneas de multipart upload	302
Multipart upload e definição de preço	303
Suporte de API para multipart upload	304
Suporte da AWS Command Line Interface para carregamento fracionado	304
Suporte do AWS SDK para upload fracionado	304
API de multipart upload e permissões	305
Definir uma configuração de ciclo de vida	309
Fazer upload de um objeto usando multipart upload	312
Fazer upload de um diretório	338
Listar multipart uploads	340
Monitorar um multipart upload	343
Abortar um multipart upload	347
Copiar um objeto	352
Limites do multipart upload	359
Copiar e mover objetos	360
Para copiar um objeto	363
Como mover um objeto	373
Baixar objetos	375
Fazer download de um objeto	376
Baixar vários objetos	377
Baixar parte de um objeto	379
Baixar um objeto de outra Conta da AWS	380
Baixar objetos arquivados	381
Solução de problemas de download de objetos	381
Verificar a integridade do objeto	382
Usar algoritmos de soma de verificação compatíveis	382
Usar Content-MD5 ao carregar objetos	391
Usar Content-MD5 e a ETag para verificar objetos carregados	392
Usar somadas de verificação à direita	392

Usar somas de verificação de nível parcial para carregamentos fracionados	393
Excluir objetos	395
Excluir objetos de um bucket habilitado para versionamento de maneira programática	396
Excluir objetos de um bucket com MFA habilitada	396
Excluir um único objeto	397
Excluir vários objetos	409
Organizar e listar objetos	412
Usar prefixos	412
Listar objetos	415
Usar pastas	417
Exibir uma visão geral do objeto	422
Visualizar propriedades de objeto	423
Trabalhar com URLs pré-assinados	424
Quem pode criar um URL pré-assinado	425
Tempo de validade para URLs pré-assinados	426
Limitar recursos de pre-signed URLs	426
Compartilhar objetos com URLs pré-assinados	428
Carregar objetos com URLs pré-assinados	432
Transformar objetos	434
Criar pontos de acesso do Object Lambda	435
Usar pontos de acesso do Amazon S3 Object Lambda	451
Considerações sobre segurança	455
Escrever funções do Lambda	462
Usar funções criadas pela AWS	494
Práticas recomendadas e diretrizes para o S3 Object Lambda	496
Tutoriais do S3 Object Lambda	498
Depuração do S3 Object Lambda	498
O que é a classe S3 Express One Zone?	500
Visão geral	502
Zona de disponibilidade única	502
Buckets de diretório	502
Endpoints e endpoints de VPC do gateway	503
Autorização baseada em sessão	503
Recursos da classe S3 Express One Zone	504
Gerenciamento de acesso e segurança	504
Registrar em log e monitoramento	505

Gerenciamento de objetos	506
AWS SDKs e bibliotecas de clientes	506
Criptografia e proteção de dados	507
Signature Version 4 (SigV4) da AWS	507
Consistência forte	507
Serviços relacionados	508
Próximas etapas	509
O que diferencia a classe S3 Express One Zone?	509
Diferenças da classe S3 Express One Zone	510
Operações de API compatíveis com a classe S3 Express One Zone	511
Recursos do Amazon S3 não compatíveis com a classe S3 Express One Zone	513
Conceitos básicos da classe S3 Express One Zone	514
Configurar o AWS Identity and Access Management (IAM) com a classe S3 Express One Zone	514
Configurar endpoints de VPC do gateway	515
Trabalhar com a classe S3 Express One Zone usando o console do S3, AWS CLI e os AWS SDKs	515
Redes para a classe S3 Express One Zone	517
Endpoints	517
Configurar endpoints de gateway da VPC	518
Buckets de diretório	519
Zonas de disponibilidade	521
Nomes de bucket de diretório	521
Diretórios	521
Nomes de chave	522
Gerenciamento de acesso	522
Trabalhar com buckets de diretório	522
Regras de nomenclatura de buckets de diretório	523
Criar um bucket de diretório	524
Visualizar as propriedades	533
Gerenciar políticas de bucket	534
Esvaziar um bucket de diretório	539
Excluir um bucket de diretório	540
Listar buckets de diretório	543
Exemplos do HeadBucket	545
Trabalhar com objetos em um bucket de diretório	547

Importar objetos em um bucket de diretório	547
Usar a funcionalidade Operações em Lote com a classe S3 Express One Zone	549
Fazer upload de um objeto	552
Usar uploads multiparte com buckets de diretório	555
Copiar um objeto	584
Excluir um objeto	589
Fazer download de um objeto	593
Exemplos do HeadObject	595
Segurança para a classe S3 Express One Zone	596
Proteção e criptografia de dados	597
IAM para a classe S3 Express One Zone	598
Políticas baseadas em identidade	614
Políticas de buckets	615
CreateSession authorization	618
Melhores práticas de segurança	619
Otimizar a performance da classe S3 Express One Zone	622
Diretrizes de performance e padrões de design	623
Desenvolver com a classe S3 Express One Zone	627
Zonas de disponibilidade e regiões da classe S3 Express One Zone	628
Endpoints regionais e zonais	630
Operações de API da classe S3 Express One Zone	631
Trabalhar com pontos de acesso	633
Configurar políticas do IAM	634
Exemplos de política de ponto de acesso	634
Chaves de condição	639
Delegar controle de acesso a pontos de acesso	640
Conceder permissões para pontos de acesso entre contas	641
Criar pontos de acesso	641
Regras para nomear pontos de acesso do Amazon S3	642
Criar um ponto de acesso	643
Criar pontos de acesso restritos a uma VPC	645
Gerenciar o acesso público	648
Usar pontos de acesso	649
Acessar um bucket por meio de pontos de acesso do S3	650
Monitorar e registrar	651
Gerenciar pontos de acesso	653

Usar um alias em estilo de bucket para seu ponto de acesso	656
Usar pontos de acesso com operações do Amazon S3	658
Restrições e limitações	662
Como trabalhar com pontos de acesso multirregionais	664
Criação de pontos de acesso multirregionais	665
Regras para nomear pontos de acesso multirregionais do Amazon S3	667
Regras para escolher buckets para pontos de acesso multirregionais do Amazon S3	668
Criar um ponto de acesso multirregional do Amazon S3	669
Bloqueio de acesso público de pontos de acesso multirregionais do Amazon S3	672
Visualizar detalhes de configuração de pontos de acesso multirregionais do Amazon S3	673
Excluir um ponto de acesso multirregional	674
Configurar pontos de acesso multirregionais	675
Como configurar o AWS PrivateLink	676
Remoção do acesso a um ponto de acesso multirregional de um endpoint da VPC	679
Usando pontos de acesso multirregionais	680
Nomes de host do ponto de acesso multirregional	681
Pontos de acesso multirregionais e Amazon S3 Transfer Acceleration	683
Permissões	683
Restrições e limitações	691
Roteamento de solicitação	694
Configuração de failover	696
Replicação do bucket	704
Operações de API compatíveis	713
Monitorar e registrar	730
Segurança	734
Proteção de dados	735
Criptografia de dados	737
Criptografia do lado do servidor	739
Uso de criptografia do lado do cliente	830
Privacidade entre redes	830
Tráfego entre clientes de serviço e on-premises e os aplicativos	831
Tráfego entre recursos da AWS na mesma região	831
AWS PrivateLink para Amazon S3	831
Tipos de VPC endpoints	832
Restrições e limitações do AWS PrivateLink do Amazon S3	833
Criar um endpoint da VPC	834

Acessar endpoints da interface do Amazon S3	834
DNS privado	834
Acessar buckets, pontos de acesso e operações de API de controle do Amazon S3 de endpoints de interface do S3	837
Atualizar uma configuração de DNS on-premises	843
Criar uma política de VPC endpoint	845
Identity and Access Management	849
Público	850
Autenticando com identidades	850
Gerenciamento do acesso usando políticas	854
Como o Amazon S3 funciona com o IAM	857
Políticas e permissões	879
Políticas de buckets	883
Políticas baseadas em identidade	932
Demonstrações usando políticas	968
Solicitar autorização	1011
Políticas gerenciadas pela AWS	1020
Usar funções vinculadas a serviços para o Amazon S3 Storage Lens	1022
Solução de problemas	1027
Gerenciar o acesso com a funcionalidade Concessões de Acesso do S3	1029
Conceitos da funcionalidade Concessões de Acesso do S3	1031
Concessões de Acesso do S3 e identidades de diretórios corporativos	1034
Conceitos básicos da funcionalidade Concessões de Acesso do S3	1043
Criar uma instância da funcionalidade Concessões de Acesso do S3	1045
Registrar um local	1057
Criar concessões	1078
Solicitar acesso aos dados do Amazon S3 por meio da funcionalidade Concessões de Acesso do S3	1091
Acessar dados do S3 por meio de uma concessão de acesso	1095
Acesso entre contas da funcionalidade Concessões de Acesso do S3	1097
Usar etiquetas da AWS com a funcionalidade Concessões de Acesso do S3	1110
Limitações da funcionalidade Concessões de Acesso do S3	1113
Integrações da funcionalidade Concessões de Acesso do S3	1114
Gerenciar o acesso com ACLs	1114
Visão geral da ACL	1115
Configurar ACLs	1136

Exemplos de políticas	1154
Usar o CORS	1159
Compartilhamento de recursos de origem cruzada: cenários de caso de uso	1159
Como o Amazon S3 avalia a configuração de CORS em um bucket?	1160
Como o ponto de acesso do Object Lambda é compatível com o CORS	1160
Configuração de CORS	1160
Configurar CORS	1166
Bloquear o acesso público	1175
Configurações do bloqueio de acesso público	1177
Executar operações de bloqueio de acesso público em um ponto de acesso	1181
O significado de "público"	1182
Usar o IAM Access Analyzer para S3 para revisar buckets públicos	1185
Permissões	1186
Configurar o bloqueio de acesso público	1187
Definir as configurações da conta	1187
Definir configurações de bucket e ponto de acesso	1190
Revisar o acesso ao bucket	1193
Quais informações o IAM Access Analyzer para S3 fornece?	1195
Habilitar o IAM Access Analyzer para S3	1196
Bloquear todo o acesso público	1197
Revisar e alterar o acesso ao bucket	1198
Arquivar descobertas de bucket	1199
Ativar uma descoberta de bucket arquivada	1200
Visualizar detalhes de descobertas	1200
Baixar um relatório do IAM Access Analyzer para S3	1200
Verificar a propriedade do bucket	1201
Quando usar a condição do proprietário do bucket	1202
Verificar um proprietário do bucket	1202
Exemplos	1203
Restrições e limitações	1206
Controlar a propriedade do objeto	1206
Configurações do Object Ownership	1208
Alterações introduzidas ao desabilitar ACLs	1210
Pré-requisitos para desabilitar ACLs	1213
Permissões do Object Ownership	1216
Desabilitar ACLs para todos os novos buckets	1216

Replicação e Object Ownership	1216
Configurar o Object Ownership	1217
Pré-requisitos para desabilitar ACLs	1218
Criação de um bucket	1230
Configurar o Object Ownership	1239
Visualizar configurações do Object Ownership	1243
Desabilitar ACLs para todos os novos buckets	1245
Solução de problemas	1248
Registro em log e monitoramento	1251
Validação de conformidade	1254
Resiliência	1256
Criptografia de backup	1258
Segurança da infraestrutura	1259
Análise de configuração e vulnerabilidade	1260
Melhores práticas de segurança	1261
Práticas recomendadas de segurança para o Amazon S3	1261
Práticas recomendadas de auditoria e monitoramento do Amazon S3	1267
Monitorar a segurança dos dados	1273
Gerenciar o armazenamento	1277
Usando o versionamento do S3	1278
Buckets não versionados, habilitados para versão e suspensos de versão	1278
Usando o versionamento do S3 com o ciclo de vida do S3	1279
Versionamento do S3	1280
Habilitar o versionamento em buckets	1284
Configurando a exclusão de MFA	1292
Trabalhando com objetos habilitados para versão	1294
Trabalhando com objetos suspensos de versão	1326
Usar o AWS Backup para o Amazon S3	1330
Trabalhando com objetos arquivados	1331
Restaurar objetos do S3 Glacier	1332
Restaurar objetos do S3 Intelligent-Tiering	1332
Usar o recurso Operações em Lote do S3 com solicitações de restauração	1333
Tempo de restauração	1333
Opções de recuperação de arquivamento	1334
Restaurar um objeto arquivado	1336
Usar o bloqueio de objetos	1345

Como o bloqueio de objetos do S3 funciona	1346
Considerações sobre a funcionalidade Bloqueio de Objetos	1350
Configurar o Object Lock	1356
Gerenciamento de classes de armazenamento	1367
Objetos acessados frequentemente	1367
Otimização automática de dados com padrões de acesso alterados ou desconhecidos	1368
Objetos acessados com pouca frequência	1370
Arquivamento de objetos	1372
Amazon S3 on Outposts	1374
Comparação de classes de armazenamento	1374
Configurar a classe de armazenamento de um objeto	1376
Amazon S3 Intelligent-Tiering	1377
Como o S3 Intelligent-Tiering funciona	1378
Como usar o S3 Intelligent-Tiering	1382
Como gerenciar o S3 Intelligent-Tiering	1387
Gerenciando o ciclo de vida	1390
Gerenciando o ciclo de vida do objeto	1391
Criando uma configuração de ciclo de vida	1392
Fazer a transição de objetos	1393
Expirando objetos	1402
Definir a configuração do ciclo de vida	1404
Usando outras configurações de bucket	1422
Configurar notificações de evento de ciclo de vida	1425
Elementos de configuração do ciclo de vida	1427
Exemplos de configuração do S3 Lifecycle	1439
Gerenciamento de inventário	1458
Buckets do Amazon S3 Inventory	1459
Listas de inventário	1460
Configurar o inventário do Amazon S3	1464
Configuração de notificações para conclusão de inventário	1474
Localização de inventário	1475
Consultar o inventário com o Athena	1479
Converter strings de ID de versão vazias em cadeias de caracteres null	1485
Trabalhar com o campo ACL de objetos	1488
Replicação de objetos	1491
Por que usar a replicação	1492

Quando usar a replicação entre regiões	1493
Quando usar Replicação na mesma região	1494
Quando usar a replicação bidirecional	1494
Quando usar S3 Batch Replication	1495
Requisitos para replicação	1495
O que é replicado?	1496
Configuração da replicação	1500
Replicar objetos existentes	1567
Configurações adicionais	1581
Obtenção de status da replicação	1616
Considerações adicionais	1620
Uso de tags de objeto	1623
Operações de API relacionadas à marcação de objetos	1625
Configurações adicionais	1627
Controle de acesso	1628
Gerenciar tags de objeto	1631
Usar tags de alocação de custos	1637
Mais informações	1638
Relatórios de uso e faturamento	1639
Relatórios de faturamento	1640
Relatório de uso	1643
Entender os relatórios de uso e faturamento	1646
Cobrança pelas respostas de erro do Amazon S3	1674
Uso do Amazon S3 Select	1688
Requisitos e limites	1689
Criar uma solicitação	1690
Erros	1691
Exemplos do S3 Select	1691
Referência SQL	1695
Uso do Batch Operations	1735
Conceitos básicos do Batch Operations	1735
Tutorial Operações em lote do S3	1737
Conceder permissões	1737
Criar uma tarefa	1747
Operações compatíveis	1771
Gerenciar trabalhos	1813

Rastreamento de relatórios de status e conclusão	1818
Usar tags	1833
Gerenciar o Bloqueio de objetos do S3	1850
Tutorial Operações em lote do S3	1873
Monitorar o Amazon S3	1874
Ferramentas de monitoramento	1875
Ferramentas automatizadas	1875
Ferramentas manuais	1875
Opções de registro em log	1876
Registrar em log com o CloudTrail	1879
Usar logs do CloudTrail com logs de acesso ao servidor do Amazon S3 e CloudWatch	
Logs	1881
Rastreamento do CloudTrail com chamadas de API SOAP do Amazon S3	1882
Eventos do CloudTrail	1883
Exemplos de arquivos de log	1895
Habilitar o CloudTrail	1901
Identificar solicitações do S3	1904
Registrando acesso ao servidor	1912
Como faço para habilitar a entrega de logs?	1912
Formato da chave de objeto de log	1915
Como os logs são entregues?	1916
Entrega de logs do servidor de melhor esforço	1917
As alterações do status do registro de bucket em logs entram em vigor ao longo do tempo	1917
Habilitar o log de acesso ao servidor do	1918
Formato do log	1940
Excluir arquivos de log	1955
Identificar solicitações do S3	1955
Monitoramento de métricas com o CloudWatch	1962
Métricas e dimensões	1965
Acessar métricas do CloudWatch	1982
Configurações de métricas do CloudWatch	1984
Notificações de eventos do Amazon S3	1994
Visão geral	1994
Tipos e destinos de notificações	1996
Usar o SQS, o SNS e o Lambda	2004
Usar o EventBridge	2034

Usar análises e insights	2045
Análise de classe de armazenamento	2045
Como configurar a análise de classe de armazenamento	2046
Análise de classe de armazenamento	2047
Como exportar dados de análise de classe de armazenamento?	2049
Configurar análise de classe de armazenamento	2050
Lente de Armazenamento do S3	2053
Métricas e recursos da Lente de Armazenamento do S3	2054
Noções básicas sobre o S3 Storage Lens	2056
Trabalhar com organizações	2068
Permissões do S3 Storage Lens	2071
Exibição de métricas de armazenamento	2076
Casos de uso das métricas da Lente de Armazenamento do Amazon S3	2108
Glossário de métricas	2136
Trabalhar com o S3 Storage Lens	2169
Trabalho com grupos de Lente de Armazenamento do S3	2219
Rastreamento de solicitações usando X-Ray	2260
Como o X-ray funciona com o Amazon S3	2260
Regiões disponíveis	2261
Hospedar sites estáticos	2262
Endpoints de site	2263
Exemplos de endpoint de site	2264
Adicionar um registro DNS CNAME	2265
Usar um domínio personalizado com o Route 53	2265
Principais diferenças entre um endpoint de site e um endpoint de API REST	2265
Habilitar a hospedagem de sites	2266
Configurar um documento de índice	2272
Pastas e documentos de índice	2272
Configurar um documento de índice	2273
Configurar um documento de erro personalizado	2275
Códigos de resposta HTTP do Amazon S3	2275
Configurar um documento de erro personalizado	2278
Configuração de permissões para acesso ao site	2279
Etapa 1: Editar configurações de bloqueio de acesso público do S3	2280
Etapa 2: Adicionar uma política de bucket	2282
Listas de controle de acesso de objetos	2284

Registro de tráfego da Web	2285
Configuração de um redirecionamento	2286
Redirecionar solicitações para outro host	2286
Configurar regras de redirecionamento	2287
Redirecionar solicitações para um objeto	2295
Desenvolvimento com o Amazon S3	2299
Fazer solicitações	2299
Sobre as chaves de acesso	2300
Endpoints de solicitações	2302
Fazer solicitações por meio do IPv6	2302
Fazer solicitações usando os AWS SDKs	2313
Fazer solicitações usando a API REST	2354
Usar a AWS CLI	2369
Uso da SDKs AWS	2371
Como trabalhar com AWS SDKs	2373
Especificar a versão da assinatura na autenticação de solicitações	2374
Usar a AWS SDK for Java	2383
Usar a AWS SDK for .NET	2385
Usar o AWS SDK for PHP e executar exemplos do PHP	2387
Usar o AWS SDK for Ruby - versão 3	2388
Usar a AWS SDK for Python (Boto)	2391
Usar os AWS Mobile SDKs for iOS e Android	2391
Uso da biblioteca JavaScript do AWS Amplify	2391
Usar a AWS SDK for JavaScript	2392
Uso dos REST API	2392
Roteamento de solicitação	2393
Tratamento de erros	2400
A resposta de erro de REST	2400
A resposta de erro de SOAP	2402
Melhores práticas com relação a erros do Amazon S3	2403
Referência	2404
Apêndice A: Usar a API SOAP	2405
Apêndice B: Autenticação de solicitações (AWS Signature Version 2)	2409
Otimizar a performance do Amazon S3	2455
Diretrizes de performance	2456
Avaliar a performance	2457

Dimensionar na horizontal	2458
Usar consulta na escala de bytes	2458
Solicitações de repetição	2458
Combinar o Amazon S3 e o Amazon EC2 na mesma região	2459
Usar o Transfer Acceleration para minimizar a latência	2459
Usar os AWS SDKs mais recentes	2459
Padrões de design de performance	2460
Armazenar conteúdo acessado com frequência em cache	2460
Tempo limite e repetição para aplicativos sensíveis à latência	2461
Dimensionamento horizontal e paralelização de solicitações	2462
Acelerar transferências de dados geograficamente dispersas	2464
O que é o S3 on Outposts?	2465
Como funciona o S3 on Outposts	2465
Regiões	2466
Buckets	2466
Objetos	2467
Chaves	2467
Versionamento do S3	2468
ID da versão	2468
Classe de armazenamento e criptografia	2468
Política de bucket	2469
Pontos de acesso do S3 on Outposts	2469
Recursos do S3 on Outposts	2470
Gerenciamento de acesso	2470
Registro e monitoramento do armazenamento	2471
Consistência forte	2471
Serviços relacionados	2471
Acessar o S3 no Outposts	2472
AWS Management Console	2472
AWS Command Line Interface	2472
AWS SDKs	2472
Pagar pelo S3 on Outposts	2473
Próximas etapas	2473
Configurar seu Outpost	2474
Pedir um novo Outpost	2474
Em que aspectos o S3 on Outposts é diferente?	2474

Especificações	2475
Operações de API compatíveis	2475
Recursos não compatíveis do Amazon S3	2476
Restrições de rede	2477
Conceitos básicos do S3 no Outposts	2477
Configurar o IAM	2478
Uso do console do S3	2486
Usar a AWS CLI e o SDK para Java	2489
Redes para S3 on Outposts	2496
Escolher seu tipo de acesso às redes	2497
Acessar buckets e objetos do S3 on Outposts	2497
Gerenciar conexões usando interfaces de rede elástica entre contas	2497
Trabalhar com buckets do S3 on Outposts	2498
Buckets	2498
Pontos de acesso	2499
Endpoints	2499
Operações de API no S3 on Outposts	2499
Criar e gerenciar o bucket do S3 no Outposts	2501
Criação de um bucket	2502
Adicionar etiquetas	2506
Uso de políticas de bucket	2508
Listar buckets	2516
Obter um bucket	2518
Excluir bucket	2519
Trabalhar com pontos de acesso	2521
Trabalhar com endpoints do	2535
Trabalhar com objetos usando o S3 on Outposts	2541
Copiar um objeto	2543
Obter um objeto	2544
Listar objetos	2548
Excluir objetos	2551
Usar o HeadBucket	2555
Executar um carregamento fracionado	2557
Usar pre-signed URLs	2565
Amazon S3 no Outposts com Amazon EMR local	2579
Armazenamento em cache de autorização e autenticação	2586

Segurança	2587
Criptografia de dados	2588
AWS PrivateLink para S3 on Outposts	2588
Chaves de política do Signature Version 4 (SigV4)	2595
Políticas gerenciadas pela AWS	2599
Usar funções vinculadas ao serviço	2601
Gerenciar o armazenamento do S3 on Outposts	2605
Gerenciar o versionamento do S3	2606
Criar e gerenciar uma configuração de ciclo de vida	2608
Replicar objetos para o S3 no Outposts	2617
Compartilhar o S3 on Outposts	2649
Outros serviços	2654
Monitoramento do S3 on Outposts	2655
Métricas do CloudWatch	2655
Amazon CloudWatch Events	2658
Logs do CloudTrail	2659
Desenvolver com o S3 on Outposts	2662
APIs do S3 on Outposts	2663
Configurar o cliente de controle do S3	2665
Fazer solicitações por meio do IPv6	2666
Exemplos de código	2678
Ações	2690
AbortMultipartUpload	2693
AbortMultipartUploads	2695
CompleteMultipartUpload	2697
CopyObject	2699
CreateBucket	2718
CreateMultiRegionAccessPoint	2739
CreateMultipartUpload	2742
DeleteBucket	2744
DeleteBucketAnalyticsConfiguration	2755
DeleteBucketCors	2756
DeleteBucketEncryption	2759
DeleteBucketInventoryConfiguration	2760
DeleteBucketLifecycle	2761
DeleteBucketMetricsConfiguration	2764

DeleteBucketPolicy	2765
DeleteBucketReplication	2772
DeleteBucketTagging	2773
DeleteBucketWebsite	2774
DeleteObject	2778
DeleteObjectTagging	2795
DeleteObjects	2796
DeletePublicAccessBlock	2826
GetBucketAccelerateConfiguration	2827
GetBucketAcl	2828
GetBucketAnalyticsConfiguration	2838
GetBucketCors	2839
GetBucketEncryption	2845
GetBucketInventoryConfiguration	2846
GetBucketLifecycleConfiguration	2847
GetBucketLocation	2851
GetBucketLogging	2853
GetBucketMetricsConfiguration	2854
GetBucketNotification	2856
GetBucketPolicy	2857
GetBucketPolicyStatus	2865
GetBucketReplication	2866
GetBucketRequestPayment	2867
GetBucketTagging	2869
GetBucketVersioning	2870
GetBucketWebsite	2871
GetObject	2875
GetObjectAcl	2901
GetObjectLegalHold	2908
GetObjectLockConfiguration	2911
GetObjectRetention	2914
GetObjectTagging	2918
GetPublicAccessBlock	2920
HeadBucket	2921
HeadObject	2925
ListBucketAnalyticsConfigurations	2930

ListBucketInventoryConfigurations	2931
ListBuckets	2933
ListMultipartUploads	2944
ListObjectVersions	2947
ListObjects	2952
ListObjectsV2	2954
PutBucketAccelerateConfiguration	2974
PutBucketAcl	2977
PutBucketCors	2988
PutBucketEncryption	2997
PutBucketLifecycleConfiguration	2998
PutBucketLogging	3007
PutBucketNotification	3014
PutBucketNotificationConfiguration	3017
PutBucketPolicy	3023
PutBucketReplication	3032
PutBucketRequestPayment	3036
PutBucketTagging	3038
PutBucketVersioning	3039
PutBucketWebsite	3040
PutObject	3048
PutObjectAcl	3077
PutObjectLegalHold	3082
PutObjectLockConfiguration	3085
PutObjectRetention	3091
RestoreObject	3095
SelectObjectContent	3100
UploadPart	3106
Cenários	3108
Criar um URL pré-assinado	3109
Criar uma página da web que oferece uma lista de objetos do Amazon S3	3146
Fazer download de objetos em um diretório local	3148
Obter um objeto de um ponto de acesso multirregional	3149
Obter um objeto de um bucket que foi modificado	3151
Conceitos básicos de buckets e objetos	3155
Conceitos básicos de criptografia	3235

Conceitos básicos de etiquetas	3241
Obter a configuração de retenção legal de um objeto	3244
Bloquear objetos do Amazon S3	3247
Gerenciar listas de controle de acesso (ACLs)	3288
Gerenciar objetos versionados em lotes com uma função do Lambda	3293
Analisar URIs	3294
Executar uma cópia multipart	3297
Realizar um carregamento fracionado	3300
Monitorar uploads e downloads	3304
Teste de unidade e integração com SDK	3307
Fazer upload em um bucket	3316
Fazer upload ou download de arquivos grandes	3317
Upload de fluxo de tamanho desconhecido	3358
Usar somas de verificação	3361
Trabalhar com objetos versionados	3366
Exemplos sem servidor	3373
Invocar uma função do Lambda em um acionador do Amazon S3	3373
Exemplos entre serviços	3385
Criar uma aplicação Amazon Transcribe	3386
Converter texto em fala e de volta em texto	3386
Criar uma aplicação com tecnologia sem servidor para gerenciar fotos	3387
Criar uma aplicação de exploração do Amazon Textract	3392
Detectar EPI em imagens	3393
Detectar entidades em texto extraído de uma imagem	3395
Detectar faces em uma imagem	3395
Detectar objetos em imagens	3396
Detectar pessoas e objetos em um vídeo	3400
Salvar o EXIF e outras informações de imagem	3401
Como transformar dados com o S3 Object Lambda	3402
Solução de problemas	3403
Solucionar erros de acesso negado (403 proibido)	3403
Políticas de bucket e políticas do IAM	3404
Configurações de ACL do Amazon S3	3407
Configurações do bloqueio de acesso público do S3	3410
Configurações de criptografia do Amazon S3	3411
Configurações do bloqueio de objetos do S3	3413

Política de endpoint da VPC	3414
Políticas do AWS Organizations	3414
Configurações do ponto de acesso	3414
Solução de problemas com operações em lote	3415
Relatório de trabalhos não é entregue quando há um problema com as permissões ou quando o modo de retenção está habilitado	3416
Falha na replicação em lote: a geração do manifesto não encontrou chaves que correspondessem aos critérios do filtro	3416
Falhas na replicação em lote após a adição de uma nova regra de replicação	3417
Objetos com falha nas operações em lote do S3 com o erro 400 InvalidRequest	3417
Crie uma falha de trabalho com a tag de trabalho ativada	3418
Acesso negado para ler o manifesto	3418
Solução de problemas do CORS	3419
Solução de problemas do ciclo de vida	3420
Executei uma operação de lista no bucket e vi objetos que achei que estavam expirados ou que tinham sido transferidos de acordo com uma regra de ciclo de vida.	3420
Como faço para monitorar o andamento da minha regra de ciclo de vida para verificar se ela está ativa?	3421
Minha quantidade de objetos do S3 ainda aumenta, mesmo depois de configurar as regras do ciclo de vida em um bucket com versionamento ativado.	3422
Como esvaziar meu bucket do S3 usando regras de ciclo de vida?	3423
Minha fatura do Amazon S3 aumentou após a transição de objetos para uma classe de armazenamento de menor custo.	3423
Eu atualizei minha política de bucket, mas meus objetos do S3 ainda estão sendo excluídos pelas regras de ciclo de vida expiradas.	3424
Posso recuperar objetos do S3 que expiraram pelas regras do ciclo de vida do S3?	3425
Solução de problemas de replicação	3425
Dicas de solução de problemas para a replicação do S3	3425
Erros de replicação em lote	3432
Solução de problemas do registro em log de acessos ao servidor	3433
Mensagens de erro comuns ao configurar o registro em log	3433
Solução de problemas de falha de entrega	3434
Solucionar problemas de versionamento	3436
Quero recuperar objetos que foram excluídos acidentalmente em um bucket com versionamento ativado	3436
Quero excluir permanentemente objetos com versionamento	3438

Ocorreu uma degradação do desempenho depois de ativar o versionamento do bucket	3439
Obter IDs de solicitação do Amazon S3 para AWS Support	3441
Usar HTTP para obter IDs de solicitação	3441
Usar um navegador da web para obter IDs de solicitação	3442
Usar os AWS SDKs para obter IDs de solicitação	3442
Usar a AWS CLI para obter IDs de solicitação	3445
Usar o Windows PowerShell para obter IDs de solicitação	3445
Usar eventos de dados do AWS CloudTrail para obter IDs de solicitação	3445
Usar o registro de acesso ao servidor do S3 para obter IDs de solicitação	3445
Histórico do documento	3446
Atualizações anteriores	3483
Glossário do AWS	3511

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.

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Tópicos

- [Recursos do Amazon S3](#)
- [Como funciona o Amazon S3](#)
- [Modelo de consistência de dados do Amazon S3](#)
- [Serviços relacionados](#)
- [Acesso ao Amazon S3](#)
- [Pagar pelo Amazon S3](#)
- [Conformidade do PCI DSS](#)

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 nas classes S3 Standard ou S3 Express One Zone para acesso frequente, economizar custos armazenando dados acessados com pouca frequência nas classes S3 Standard-IA ou S3 One Zone-IA e arquivar dados

com os menores custos nas classes S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive.

A classe Amazon S3 Express One Zone é uma classe de armazenamento de zona única e alto desempenho do Amazon S3 desenvolvida com o propósito específico de fornecer acesso consistente aos dados e com latência inferior a dez milissegundos para as aplicações mais sensíveis à latência. A classe S3 Express One Zone é a classe de armazenamento de objetos em nuvem de menor latência disponível atualmente, com velocidades de acesso aos dados até dez vezes mais rápidas e custos de solicitação 50% mais baixos do que a classe S3 Standard. A classe S3 Express One Zone é a primeira classe de armazenamento do S3 em que é possível selecionar uma única zona de disponibilidade com a opção de manter o armazenamento de objetos e os recursos de computação na mesma localização, o que fornece a maior velocidade de acesso possível. Além disso, para aumentar ainda mais a velocidade de acesso e oferecer suporte a centenas de milhares de solicitações por segundo, os dados são armazenados em um novo tipo de bucket: um bucket de diretório do Amazon S3. Para obter mais informações, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

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 ter mais informações, consulte [Uso de classes de armazenamento do Amazon S3](#). 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.

- [Ciclo de vida do S3](#): configure uma política de ciclo de vida para gerenciar os 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.

- [Replcação 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 e segurança

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 do Bloqueio de Acesso Público são ativadas no nível do bucket. Recomendamos que você mantenha todas as configurações do Bloqueio de Acesso Público habilitadas, a menos que precise desativar uma ou mais delas para seu caso de uso específico. Para ter mais informações, consulte [Configurar o bloqueio de acesso público para seus buckets do S3](#).
- [AWS Identity and Access Management \(IAM\)](#): o IAM é um serviço da Web que ajuda você a controlar de maneira segura o acesso aos recursos da AWS, incluindo recursos do Amazon S3. Com o IAM, é possível gerenciar, de maneira centralizada, permissões que controlam quais recursos da AWS os usuários poderão acessar. Você usa o IAM para controlar quem é autenticado (fez login) e autorizado (tem permissões) a usar os recursos.
- [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 de usuários do IAM para controle de acesso, em vez de ACLs. As políticas são uma opção de controle de acesso simplificada e mais flexível. Com políticas de bucket e políticas de ponto de acesso, é possível definir regras que se aplicam amplamente a todas as solicitações para seus recursos do Amazon S3. Para obter mais informações sobre os casos específicos em que você usaria ACLs, em vez de políticas baseadas em recursos ou políticas de usuário do IAM, consulte [Gerenciar o acesso com ACLs](#).
- [Propriedade de objeto do S3](#): assuma a propriedade de cada objeto do bucket simplificando o gerenciamento de acesso para dados armazenados no Amazon S3. A Propriedade de objeto do S3 é uma configuração no nível do bucket do Amazon S3 que você pode usar para desabilitar ou habilitar as ACLs. Por padrão, as ACLs estão desabilitadas. Com as ACLs desabilitadas, o proprietário do bucket possui todos os objetos do bucket e gerencia o acesso aos dados exclusivamente usando políticas de gerenciamento de acesso.
- [IAM Access Analyzer para S3](#): avalie e monitore as políticas de acesso ao bucket do S3, garantindo que elas forneçam apenas o acesso pretendido aos 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, HEAD e LIST 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 60 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](#).

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 ter mais informações, consulte [Modelo de consistência de dados do Amazon S3](#).

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](#)
- [Objetos](#)
- [Chaves](#)
- [Versionamento do S3](#)
- [ID da versão](#)
- [Política de bucket](#)

- [Pontos de acesso do S3](#)
- [Listas de controle de acesso \(ACLs\)](#)
- [Regiões](#)

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, acesse o [Console do Service Quotas](#).

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](#) (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](#) 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](#).

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 \(nome\)](#) e um [ID da versão](#) (se o Versionamento do S3 estiver habilitado no bucket). Para obter mais informações sobre objetos, consulte [Visão geral de objetos Amazon S3](#).

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](#).

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 ter mais informações, consulte [Usando o versionamento em buckets do S3](#).

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 ter mais informações, consulte [Usando o versionamento em buckets do S3](#).

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 obter mais informações, consulte [Exemplos de políticas de bucket do Amazon S3](#).

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

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 \(ACL\)](#).

A Propriedade de objetos do S3 é uma configuração no nível do bucket do Amazon S3 que você pode usar para controlar a propriedade de objetos carregados no bucket e para desabilitar ou habilitar as ACLs. Por padrão, a Propriedade de Objetos está definida com a configuração Imposto pelo proprietário do bucket e todas as ACLs estão desabilitadas. Quando as ACLs são desabilitadas, o proprietário do bucket possui todos os objetos do bucket e gerencia o acesso a eles exclusivamente usando políticas de gerenciamento de acesso.

A maioria dos casos de uso modernos no Amazon S3 não exige mais o uso de ACLs. Recomendamos manter as ACLs desabilitadas, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com as ACLs desabilitadas, é possível usar políticas para controlar o acesso a todos os objetos no bucket, independentemente de quem carregou os objetos para o bucket. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

Regiões

Você pode escolher a região da Região da AWS geográfica onde o Amazon S3 armazena os buckets criados. É possível escolher uma r[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.

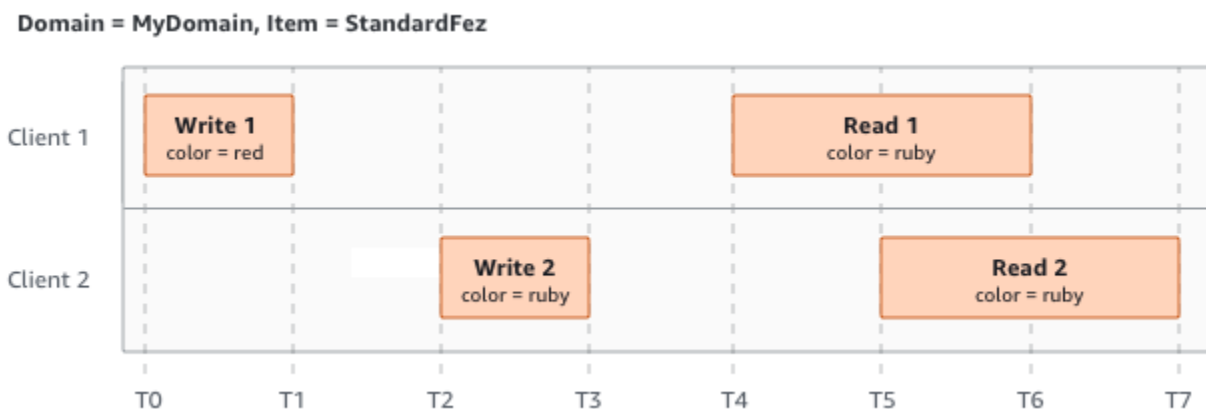
As configurações de bucket têm um modelo de consistência eventual. Especificamente, isso significa que:

- 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

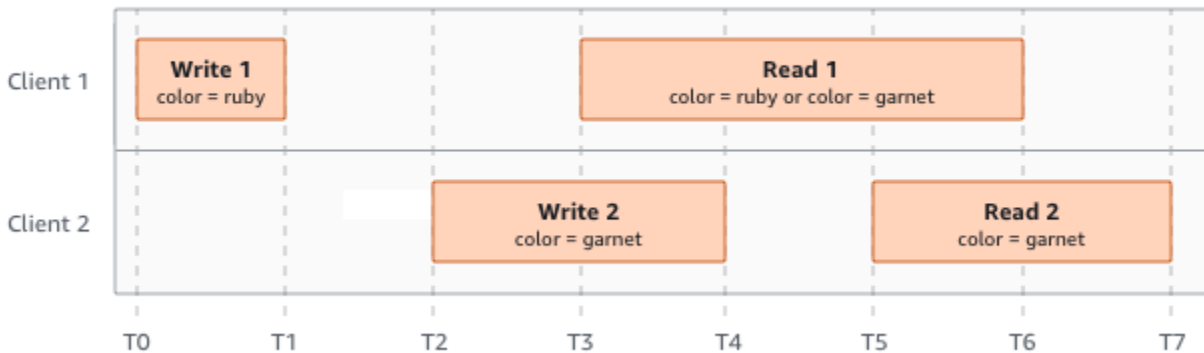
Esta seção fornece exemplos de comportamento a serem esperados do Amazon S3 quando vários clientes estão gravando nos mesmos itens.

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`.



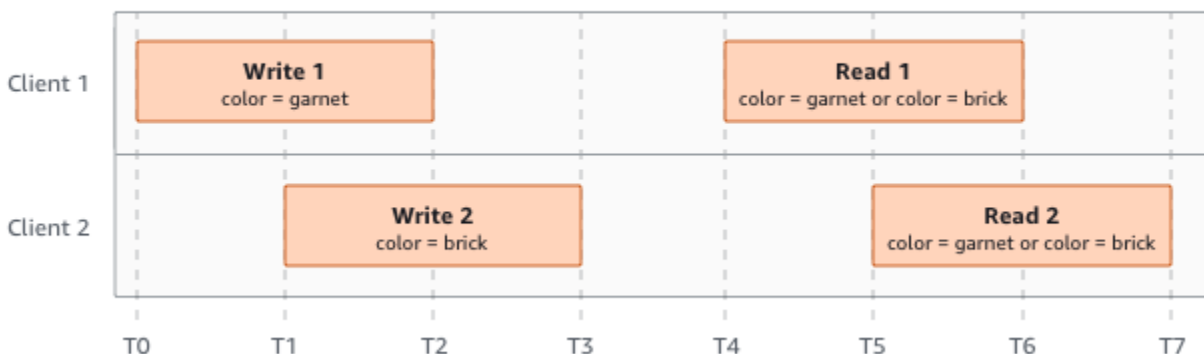
No próximo exemplo, a W2 não é encerrada antes do início da R1. Portanto, R1 pode retornar `color = ruby` ou `color = garnet`. No entanto, como W1 e W2 terminam antes do início do R2, o R2 retorna `color = garnet`.

Domain = MyDomain, Item = StandardFez



No último exemplo, o W2 começa antes que o W1 tenha recebido uma confirmação. 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 a 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

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:

- [Amazon Elastic Compute Cloud \(Amazon EC2\)](#): oferece 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 aplicações 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 o 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.

SDKs da AWS

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. É possível enviar solicitações para o Amazon S3 usando a API REST ou as bibliotecas SDK da AWS, que envolvem a API REST do Amazon S3 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, 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 calcularão a assinatura das chaves fornecidas. Para obter mais informações sobre como fazer solicitações ao Amazon S3, consulte [Fazer solicitações](#).

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 correspondesse 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](#).

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](#).

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Vídeo: Conceitos básicos do Amazon S3

Pré-requisitos

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

Tópicos

- [Pré-requisito: Configuração do Amazon S3](#)
- [Etapa 1: criar seu primeiro bucket do S3](#)
- [Etapa 2: fazer upload de um objeto para o seu bucket](#)
- [Etapa 3: fazer download de um objeto](#)
- [Etapa 4: copiar seu objeto para uma pasta](#)
- [Etapa 5: excluir seus objetos e bucket](#)
- [Próximas etapas](#)

- [Práticas recomendadas de controle de acesso](#)

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

- [Cadastre-se em uma Conta da AWS](#)
- [Criar um usuário com acesso administrativo](#)

Cadastre-se em uma Conta da AWS

Se você ainda não tem 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 inscrição envolve receber uma chamada telefônica e digitar um código de verificação no teclado do telefone.

Quando você se cadastra em uma Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como uma prática recomendada de segurança, atribua o acesso administrativo para um usuário e use somente o usuário-raiz para executar [tarefas que requerem o acesso de usuário-raiz](#).

A AWS 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 Minha conta.

Criar um usuário com acesso administrativo

Depois de se cadastrar em uma Conta da AWS, proteja seu Usuário raiz da conta da AWS, habilite o AWS IAM Identity Center e crie um usuário administrativo para não usar o usuário raiz em tarefas cotidianas.

Proteger seu Usuário raiz da conta da AWS

1. Faça login no [AWS Management Console](#) como o proprietário da conta ao escolher a opção Usuário raiz e inserir o endereço de e-mail da Conta da AWS. Na próxima página, digite sua senha.

Para obter ajuda ao fazer login usando o usuário-raiz, consulte [Signing in as the root user](#) (Fazer login como usuário-raiz) no Guia do usuário do Início de Sessão da AWS.

2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

Para obter instruções, consulte [Habilitar um dispositivo MFA virtual para o usuário-raiz de sua conta da Conta da AWS \(console\)](#) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilitar o IAM Identity Center.

Para obter instruções, consulte [Habilitar AWS IAM Identity Center](#) no Guia do usuário do AWS IAM Identity Center.

2. No Centro de Identidade do IAM, conceda o acesso administrativo para um usuário.

Para obter um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte [Configurar o acesso dos usuários com o Diretório do Centro de Identidade do IAM padrão](#) no Guia do usuário do AWS IAM Identity Center.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com seu usuário do Centro de Identidade do IAM, use a URL de login que foi enviada ao seu endereço de e-mail quando você criou o usuário do Centro do Usuário do IAM.

Para obter ajuda com o login utilizando um usuário do Centro de Identidade do IAM, consulte [Fazer login no portal de acesso da AWS](#), no Guia do usuário do Início de Sessão da AWS.

Atribuir acesso para usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

Para obter instruções, consulte [Create a permission set](#) no Guia do usuário do AWS IAM Identity Center.

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

Para obter instruções, consulte [Add groups](#) no Guia do usuário do AWS IAM Identity Center.

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

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

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. Na barra de navegação na parte superior da página, escolha o nome da Região da AWS exibida no momento. Em seguida, escolha a região na qual você deseja criar um bucket.

 Note

Para reduzir a latência e os custos e atender aos requisitos regulatórios, selecione uma região perto de você. Os objetos armazenados em uma região nunca saem dessa região, a menos que você os transfira para outra região. Para obter uma lista de Regiões da AWS do Amazon S3, consulte [Endpoints de AWS service \(Serviço da AWS\)](#) no Referência geral da Amazon Web Services.

3. No painel de navegação à esquerda, escolha Buckets.
4. Selecione Create bucket (Criar bucket).


A página Create bucket (Criar bucket) é aberta.

5. Em Configuração geral, visualize a Região da AWS onde o bucket será criado.
6. Em Tipo de bucket, escolha Propósito geral.
7. Para Nome do bucket, insira um nome para o bucket.

O nome do bucket deve:


- Seja exclusivo em 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) Regions).
- Ter entre 3 e 63 caracteres.
- Consistem apenas em letras minúsculas, números, pontos (.) e hífens (-). 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.
- Inicie e termine com uma letra ou um número.

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

 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.

8. O AWS Management Console permite que você copie as configurações de um bucket existente para seu novo bucket. Se você não quiser copiar as configurações de um bucket existente, vá para a próxima etapa.

 Note

Essa opção:

- Não está disponível na AWS CLI, mas apenas no console.
- Não está disponível para buckets de diretório.
- Não copia a política do bucket existente para o novo bucket.

Para copiar as configurações de um bucket existente, em Copiar configurações do bucket existente, selecione Escolher bucket. A janela Escolher bucket é aberta. Encontre o bucket com as configurações que você gostaria de copiar e selecione Escolher bucket. A janela Escolher bucket é fechada e a janela Criar bucket é reaberta.

Em Copiar configurações do bucket existente, agora você verá o nome do bucket selecionado. Você também verá uma opção Restaurar padrões, que pode ser usada para remover as configurações do bucket copiado. Revise as configurações restantes do bucket na página Criar bucket. Você verá que agora elas correspondem às configurações do bucket que você selecionou. Você pode pular para a etapa final.

9. 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

- Imposto pelo proprietário do bucket (padrão): 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 de acesso aos dados no bucket do S3. O bucket usa políticas exclusivamente para definir o controle de acesso.

Por padrão, as ACLs estão desabilitadas. A maioria dos casos de uso modernos no Amazon S3 não exige mais o uso de ACLs. Recomendamos manter as ACLs desabilitadas, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

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 Proprietário do bucket preferido para exigir que todos os uploads do Amazon S3 incluam a ACL `bucket-owner-full-control` pré-configurada, será possível [adicionar uma política de bucket](#) que só permita uploads de objetos que usem 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

A configuração padrão é imposto pelo proprietário do Bucket. Para aplicar a configuração padrão e manter as ACLs desabilitadas, somente a permissão `s3:CreateBucket` é necessária. Para habilitar ACLs, é necessário ter a permissão `s3:PutBucketOwnershipControls`.

10. Em Configurações de bloqueio de acesso público para este bucket, selecione as configurações de bloqueio de acesso público que deseja aplicar ao bucket.

Por padrão, todas as quatro configurações do Bloqueio de Acesso Público são habilitadas. Recomendamos que você mantenha todas as configurações habilitadas, a menos que precise desativar uma ou mais delas para seu caso de uso específico. Para obter mais informações sobre como bloquear o acesso público, consulte [Bloquear o acesso público ao armazenamento do Amazon S3](#).

 Note

Para habilitar todas as configurações do Bloqueio de Acesso Público, somente a permissão `s3:CreateBucket` é necessária. Para desabilitar qualquer configuração do Bloqueio de Acesso Público, é necessário ter a permissão `s3:PutBucketPublicAccessBlock`.

11. (Opcional) Em Bucket Versioning (Versionamento do bucket), você pode escolher se deseja manter variantes de objetos no bucket. Para obter mais informações sobre versionamento, consulte [Usando o versionamento em buckets do S3](#).

Para desabilitar ou habilitar o versionamento em seu bucket, escolha Disable (Desabilitar) ou Enable (Habilitar).

12. (Opcional) Em Tags (Etiquetas), você pode optar por adicionar etiquetas ao seu bucket. As etiquetas são pares de chave-valor usados para categorizar o armazenamento.

Para adicionar uma etiqueta, escolha Add Tag (Adicionar etiqueta) e insira uma chave e um valor para cada etiqueta.

13. Em Default encryption (Criptografia padrão), escolha Edit (Editar).
14. Para configurar a criptografia padrão, em Tipo de criptografia, selecione uma das seguintes opções:
 - Chave gerenciada pelo Amazon S3 (SSE-S3)
 - Chave do AWS Key Management Service (SSE-KMS)

 Important

Se você usar a opção SSE-KMS em sua configuração de criptografia padrão, estará sujeito à cota de solicitações por segundo (RPS) do AWS KMS. Para obter mais informações sobre as cotas do AWS KMS e como solicitar um aumento de cota, consulte [Cotas](#) no Guia do desenvolvedor do AWS Key Management Service.

Os buckets e novos objetos são criptografados com criptografia do lado do servidor com uma chave gerenciada do Amazon S3 como nível básico da configuração de criptografia. Para

obter mais informações sobre criptografia padrão, consulte [Definir o comportamento padrão da criptografia para os buckets do Amazon S3](#).

Para obter mais informações sobre como usar a criptografia no lado do servidor do Amazon S3 para criptografar seus dados, consulte [Usar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#).

15. Se você selecionou Chave do AWS Key Management Service (SSE-KMS), faça o seguinte:
 - a. Em Chave do AWS KMS, especifique sua chave do KMS de uma das seguintes maneiras:
 - Para escolher entre uma lista de chaves do KMS disponíveis, selecione Escolher entre suas AWS KMS keys e escolha sua chave do KMS na lista de chaves disponíveis.

As chaves Chave gerenciada pela AWS (aws/s3) e as chaves gerenciadas pelo cliente são exibidas nessa lista. Para ter mais informações sobre chaves gerenciadas pelo cliente, consulte [Chaves de clientes e chaves da AWS](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para inserir o ARN da chave do KMS, escolha Inserir ARN da AWS KMS key e insira o ARN da chave do KMS no campo exibido.
- Para criar uma chave gerenciada pelo cliente no console do AWS KMS, selecione Criar uma chave do KMS.

Para ter mais informações sobre como criar uma AWS KMS key, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

Important

Você só pode usar chaves do KMS disponíveis na mesma Região da AWS que o bucket. O console do Amazon S3 lista somente as primeiras 100 chaves do KMS na mesma região que o bucket. Para usar uma chave do KMS que não esteja listada, você deve inserir o ARN da chave do KMS. Se quiser usar uma chave do KMS que seja de propriedade de outra conta, primeiro você deverá ter permissão para usar a chave e, depois, inserir o ARN da chave do KMS. Para obter mais informações sobre permissões entre contas para chaves do KMS, consulte [Criar chaves do KMS que outras contas podem usar](#) no Guia do desenvolvedor do AWS Key Management Service. Para obter mais informações sobre a SSE-KMS, consulte [Especificando criptografia no lado do servidor com o AWS KMS \(SSE-KMS\)](#).


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 ter mais informações, consulte [Identificar chaves do KMS 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 [Usar criptografia do lado do servidor com o AWS KMS \(SSE-KMS\)](#).

- b. Quando você configura seu bucket para usar a criptografia padrão com o SSE-KMS, também pode habilitar as chaves de bucket do S3. As chaves de bucket do S3 diminuem o custo de criptografia reduzindo o tráfego de solicitações do Amazon S3 para o AWS KMS. Para ter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3](#).

Para usar Chaves de bucket do S3, em Bucket Key (Chave de bucket), escolha Enable (Habilitar).

16. (Opcional) Se você quiser habilitar o Bloqueio de Objetos do S3, faça o seguinte:
 - a. Selecione Advanced settings (Configurações avançadas).

 Important

A ativação do bloqueio de objetos também permite o versionamento para o bucket. Depois de habilitar, é necessário definir as configurações de retenção padrão e retenção legal do bloqueio de objetos para proteger novos objetos contra exclusão ou substituição.

- b. Se quiser habilitar o bloqueio de objetos, escolha Enable (Habilitar), leia o aviso que aparece e confirme-o.

Para ter mais informações, consulte [Usar o bloqueio de objetos do S3](#).

Note

Para criar um bucket com bloqueio de objetos habilitado, você deve ter as seguintes permissões: `s3:CreateBucket`, `s3:PutBucketVersioning` e `s3:PutBucketObjectLockConfiguration`.

17. Selecione 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](#).

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.

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

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 (Fazer 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 (Carregar).

O upload de um objeto no bucket foi realizado corretamente.

Próxima etapa

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

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.

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Uso do console do S3

Esta seção explica como usar o console do Amazon S3 para fazer baixar um objeto de um bucket do S3.

Note

- Só é possível baixar um objeto por vez.
- Se você usar o console do Amazon S3 para baixar um objeto cujo nome da chave termine com um ponto (.), o ponto será removido do nome da chave do objeto baixado. Para reter o ponto no final do nome do objeto baixado, você deve usar a AWS Command Line Interface (AWS CLI), AWS SDKs ou a API REST do Amazon S3.

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:

- Marque a caixa de seleção ao lado do objeto e escolha Fazer download. Se você quiser baixar o objeto em uma pasta específica, no menu Ações, escolha Fazer download como.
- Se você quiser baixar uma versão específica do objeto, ative Mostrar versões (ao lado da caixa de pesquisa). Marque a caixa de seleção ao lado da versão do objeto que você deseja e escolha Fazer Download. Se você quiser baixar o objeto em uma pasta específica, no menu Ações, escolha Fazer download como.

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](#).

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.

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

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.

- Escolha Actions (Ações) e escolha Copy (Copiar) na lista de opções exibida.

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

- Escolha a pasta de destino:

- Escolha Browse S3 (Navegar no S3).
- 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.

- 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/`.

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

O Amazon S3 copia seus objetos na pasta de destino.

Próxima etapa

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

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.

 Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Tópicos

- [Excluir um objeto](#)
- [Esvaziar o bucket](#)
- [Excluir bucket](#)

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. Selecione o objeto que deseja excluir.
3. Escolha Excluir nas opções no canto superior direito.
4. Na página Excluir objetos, digite **delete** para confirmar a exclusão de seus objetos.
5. Escolha Delete objects (Excluir objetos).

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 algumas 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.

 Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Tópicos

- [Entender casos de uso comuns](#)
- [Controlar o acesso a buckets e objetos](#)
- [Gerencie e monitore seu armazenamento](#)
- [Desenvolvimento com o Amazon S3](#)
- [Aprenda com os tutoriais](#)
- [Explore o treinamento e o suporte](#)

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 ter mais informações, consulte [Hospedagem de um site estático usando o Amazon S3](#).
- 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](#).

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 do Bloqueio de Acesso Público são ativadas no nível do bucket.
- [Identidades do AWS Identity and Access Management \(IAM\)](#): use o IAM ou o AWS IAM Identity Center para criar identidades do IAM em sua Conta da AWS a fim de gerenciar o acesso aos recursos do Amazon S3. Por exemplo, é possível usar o IAM com o Amazon S3 para controlar o tipo de acesso que um usuário ou grupo de usuários tem a um bucket do Amazon S3 pertencente à sua Conta da AWS. Para obter mais informações sobre as identidades do IAM e as práticas recomendadas, consulte [Identidades do IAM \(usuários, grupos de usuários e perfis\)](#) no Guia do usuário do IAM.
- [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 de usuários do IAM para controle de acesso, em vez de ACLs. As políticas são uma opção de controle de acesso simplificada e mais flexível. Com políticas de bucket e políticas de ponto de acesso, é possível definir regras que se aplicam amplamente a todas as solicitações para seus recursos do Amazon S3. Para obter mais informações sobre os casos específicos em que você usaria ACLs, em vez de políticas baseadas em recursos ou políticas de usuário do IAM, consulte [Gerenciamento de identidade e acesso para o Amazon S3](#).
- [Propriedade de objeto do S3](#): assuma a propriedade de cada objeto do bucket simplificando o gerenciamento de acesso para dados armazenados no Amazon S3. A Propriedade de objeto do S3 é uma configuração no nível do bucket do Amazon S3 que você pode usar para desabilitar ou habilitar as ACLs. Por padrão, as ACLs estão desabilitadas. Com as ACLs desabilitadas, o proprietário do bucket possui todos os objetos do bucket e gerencia o acesso aos dados exclusivamente usando políticas de gerenciamento de acesso.

- [IAM Access Analyzer para S3](#): avalie e monitore as políticas de acesso ao bucket do S3, garantindo que elas forneçam apenas o acesso pretendido aos recursos do S3.

Gerencie e monitore seu armazenamento

- [Como gerenciar seu armazenamento](#): 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 S3 para recuperação de desastres, o ciclo de vida do S3 para gerenciar custos de armazenamento e o bloqueio de objetos do S3 para atender aos requisitos de conformidade.
- [Monitoramento de seu armazenamento](#): 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](#): 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](#) para entender, analisar e otimizar 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, buckets ou prefixos. Use a [Análise de classe de armazenamento](#) 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 ter mais informações, consulte [Fazer solicitações](#).

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 ter mais informações, consulte [Desenvolver com o Amazon S3 usando a API REST](#).

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](#).

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](#).

SDKs e Explorers da AWS

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 AWS SDKs, 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 ter mais informações, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers](#).

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.

Aprenda com os tutoriais

Você pode começar usando tutoriais passo a passo para saber mais sobre a 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.

Conceitos básicos

- [Tutorial: Armazenar e recuperar um arquivo com o Amazon S3](#)
- [Tutorial: Começar a usar o S3 Intelligent-Tiering](#)
- [Tutorial: Começar a usar as classes de armazenamento do Amazon S3 Glacier](#)

Otimizar custos do armazenamento

- [Tutorial: Começar a usar o S3 Intelligent-Tiering](#)
- [Tutorial: Começar a usar as classes de armazenamento do Amazon S3 Glacier](#)
- [Tutorial: Otimizar custos e obter visibilidade do uso com a Lente de Armazenamento do S3](#)

Gerenciar o armazenamento

- [Tutorial: Introdução aos pontos de acesso multirregionais do Amazon S3](#)
- [Tutorial: replicar objetos existentes nos buckets do Amazon S3 com o S3 Batch Replication](#)

Hospedar vídeos e sites

- [Tutorial: Hospedagem de transmissão sob demanda com o Amazon S3, Amazon CloudFront e Amazon Route 53](#)
- [Tutorial: configurar um site estático no Amazon S3](#)
- [Tutorial: Configurar um site estático usando um domínio personalizado registrado no Route 53](#)

Processar dados

- [Tutorial: Como transformar dados para sua aplicação com o S3 Object Lambda](#)
- [Tutorial: Detectar e editar dados PII com o S3 Object Lambda e o Amazon Comprehend](#)
- [Tutorial: Using S3 Object Lambda to dynamically watermark images as they are retrieved](#) (Tutorial: Como usar o S3 Object Lambda para colocar marca d'água em imagens de maneira dinâmica à medida que são recuperadas)

- [Tutorial: Vídeos de transcodificação em lote com operações em lote do S3, AWS Lambda e AWS Elemental MediaConvert](#)

Proteger dados

- [Tutorial: Verificar a integridade dos dados no Amazon S3 com somas de verificação adicionais](#)
- [Tutorial: Replicar dados em uma região ou entre Regiões da AWS usando a Replicação do S3](#)
- [Tutorial: Proteger dados no Amazon S3 contra exclusão acidental ou bugs de aplicações usando o Versionamento do S3, o Bloqueio de Objeto do S3 e a Replicação do S3](#)
- [Tutorial: replicar objetos existentes nos buckets do Amazon S3 com o S3 Batch Replication](#)

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](#).

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.

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Tópicos

- [Criar um bucket](#)
- [Armazenar e compartilhar dados](#)
- [Compartilhar recursos](#)
- [Proteger dados](#)

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

A Propriedade de objeto do S3 é uma configuração no nível do bucket do Amazon S3 que você pode usar para controlar a propriedade de objetos carregados no bucket e para desabilitar ou habilitar as listas de controle de acesso (ACLs). Por padrão, a Propriedade de Objetos está definida com a configuração Imposto pelo proprietário do bucket e todas as ACLs estão desabilitadas. Com as ACLs desabilitadas, o proprietário do bucket possui cada objeto no bucket e gerencia o acesso aos dados exclusivamente usando políticas.

Há três configurações em Propriedade de objeto que podem ser usadas tanto para controlar a propriedade de objetos carregados no bucket como para desabilitar ou habilitar ACLs:

ACLs desabilitadas

- Imposto pelo proprietário do bucket (padrão): 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): 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](#).

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, as quatro configurações de Bloquear todo o acesso público são habilitadas para novos buckets. Recomendamos que você mantenha todas as quatro configurações habilitadas, a menos que precise desativar uma ou mais delas para seu caso de uso específico. É possível modificar as configurações individuais do Bloqueio de Acesso Público usando o console do Amazon S3.

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

Se precisar conceder acesso a usuários específicos, é recomendável usar políticas do AWS Identity and Access Management (IAM) em vez de desabilitar todas as configurações do Bloqueio de Acesso Público. Manter o Bloqueio de Acesso Público é uma prática recomendada de segurança. 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 solicitado tenha recebido permissão específica.

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

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](#).

Políticas de buckets

Com as políticas de bucket, é possível personalizar o acesso ao bucket para ajudar a garantir que somente os usuários que você aprovou possam acessar recursos e executar ações neles. Além das políticas de bucket, recomendamos 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 [Políticas de bucket para o Amazon S3](#).

Ao criar políticas, evite o uso de caracteres curinga (*) no elemento Principal, porque ele permite que qualquer pessoa acesse os recursos do Amazon S3. Em vez disso, é melhor listar explicitamente usuários ou grupos que têm permissão para acessar o bucket. Em vez de incluir um caractere 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 políticas de bucket usando chaves de condição](#).

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 um grupo ou perfil do IAM, você poderá usar endpoints da VPC 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](#).

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, recomendamos implementar o versionamento do S3 e o Bloqueio de objetos do S3. Esses recursos ajudam a evitar alterações acidentais em dados críticos e permitem reverter ações não intencionais. Esse recurso de reversão 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](#). Para obter mais informações sobre Bloqueio de objetos, consulte [Usar o bloqueio de objetos do S3](#).

Gerenciamento do ciclo de vida do objeto para eficiência de custos

Para gerenciar os objetos de modo que eles sejam armazenados de maneira econômica durante todo o ciclo de vida, é possível emparelhar configurações de ciclo de vida com o versionamento de objetos do S3. As configurações de ciclo de vida definem as ações que você deseja que o Amazon S3 execute durante a vida útil de um objeto. Por exemplo, é possível criar uma configuração 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 configuração de ciclo de vida para todos os objetos no bucket ou para um subconjunto de objetos usando um prefixo ou uma tag compartilhados.

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

Replicação entre regiões para vários locais de escritório

Ao criar buckets que são acessados por diferentes locais de escritório, considere a implementação da Replicação do S3 Entre Regiões Diferentes. 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 desse recurso aumenta os custos de armazenamento.

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

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. Ao escrever a política de bucket para o site estático, aceite apenas ações `s3:GetObject` e não permissões `ListObject` e `PutObject`. 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](#).

Recomendamos manter o Bloqueio de Acesso Público habilitado. Se quiser manter todas as quatro configurações do Bloqueio de Acesso Público habilitadas e hospedar um site estático, use o controle de acesso à origem (OAC) do Amazon CloudFront. O Amazon CloudFront fornece os recursos necessários para configurar um site estático seguro. Os sites estáticos do Amazon S3 só são compatíveis com 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 essas ferramentas possam ser usadas para atingir o mesmo objetivo, algumas delas podem se adequar melhor do que outras às configurações existentes.

Propriedade de objeto do S3

A Propriedade de objeto do S3 é uma configuração no nível do bucket do Amazon S3 que você pode usar para controlar a propriedade de objetos carregados no bucket e para desabilitar ou habilitar

as listas de controle de acesso (ACLs). Por padrão, a Propriedade de Objetos está definida com a configuração Imposto pelo proprietário do bucket e todas as ACLs estão desabilitadas. Com as ACLs desabilitadas, o proprietário do bucket possui cada objeto no bucket e gerencia o acesso aos dados exclusivamente usando políticas.

A maioria dos casos de uso modernos no Amazon S3 não exige mais o uso de ACLs. Recomendamos manter as ACLs desabilitadas, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente.

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

Políticas de usuário

É possível compartilhar recursos com um grupo limitado de pessoas usando políticas de usuários e grupos 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. Depois, é possível usar uma política de usuário do IAM para compartilhar o bucket com esse grupo. 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](#).

Listas de controle de acesso

Como regra geral, recomendamos que você use políticas de bucket do S3 ou políticas de usuários do IAM para controle de acesso. Usar políticas em vez de ACLs simplifica o gerenciamento de permissões. As ACLs do Amazon S3 são o mecanismo de controle de acesso original do Amazon S3 anterior ao IAM. No entanto, determinados cenários de controle de acesso exigem o uso de ACLs. Por exemplo, suponha que o proprietário do bucket deseja conceder permissão para objetos, mas nem todos os objetos pertencem ao proprietário do bucket. Nesse caso, o proprietário do objeto deve, primeiro, conceder permissão ao proprietário do bucket usando uma ACL do objeto.

A maioria dos casos de uso modernos no Amazon S3 não exige mais o uso de ACLs. 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 Imposto pelo proprietário do bucket para a Propriedade de Objetos do S3, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Quando a configuração Imposto pelo proprietário do bucket estiver habilitada, as solicitações para definir listas de controle de acesso (ACLs) ou atualizar ACLs falharão e retornarão 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](#).

Prefixos

Ao tentar compartilhar recursos específicos de um bucket, é possível replicar permissões para pastas usando prefixos. O console do Amazon S3 é compatível com o conceito de pasta como um meio de agrupar objetos usando um prefixo de nome compartilhado para objetos. Portanto, para conceder uma permissão explícita a um usuário do IAM para acessar os recursos associados a esse prefixo, é possível especificar esse prefixo nas condições da política de usuário do IAM.

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

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 permitir o acesso aos recursos marcados, use a condição `ResourceTag/key-name` em uma política de usuário do IAM.

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 [Proteger os dados usando criptografia do lado do servidor](#).

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 do lado do cliente](#).

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](#).

O Amazon S3 é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, por um perfil ou por um AWS service (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 Amazon 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.

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Conceitos básicos

- [Tutorial: Armazenar e recuperar um arquivo com o Amazon S3](#)
- [Tutorial: Começar a usar o S3 Intelligent-Tiering](#)
- [Tutorial: Começar a usar as classes de armazenamento do Amazon S3 Glacier](#)

Otimizar custos do armazenamento

- [Tutorial: Começar a usar o S3 Intelligent-Tiering](#)
- [Tutorial: Começar a usar as classes de armazenamento do Amazon S3 Glacier](#)
- [Tutorial: Otimizar custos e obter visibilidade do uso com a Lente de Armazenamento do S3](#)

Gerenciar o armazenamento

- [Tutorial: Introdução aos pontos de acesso multirregionais do Amazon S3](#)
- [Tutorial: replicar objetos existentes nos buckets do Amazon S3 com o S3 Batch Replication](#)

Hospedar vídeos e sites

- [Tutorial: Hospedagem de transmissão sob demanda com o Amazon S3, Amazon CloudFront e Amazon Route 53](#)
- [Tutorial: configurar um site estático no Amazon S3](#)
- [Tutorial: Configurar um site estático usando um domínio personalizado registrado no Route 53](#)

Processar dados

- [Tutorial: Como transformar dados para sua aplicação com o S3 Object Lambda](#)
- [Tutorial: Detectar e editar dados PII com o S3 Object Lambda e o Amazon Comprehend](#)
- [Tutorial: Using S3 Object Lambda to dynamically watermark images as they are retrieved](#) (Tutorial: Como usar o S3 Object Lambda para colocar marca d'água em imagens de maneira dinâmica à medida que são recuperadas)
- [Tutorial: Vídeos de transcodificação em lote com operações em lote do S3, AWS Lambda e AWS Elemental MediaConvert](#)

Proteger dados

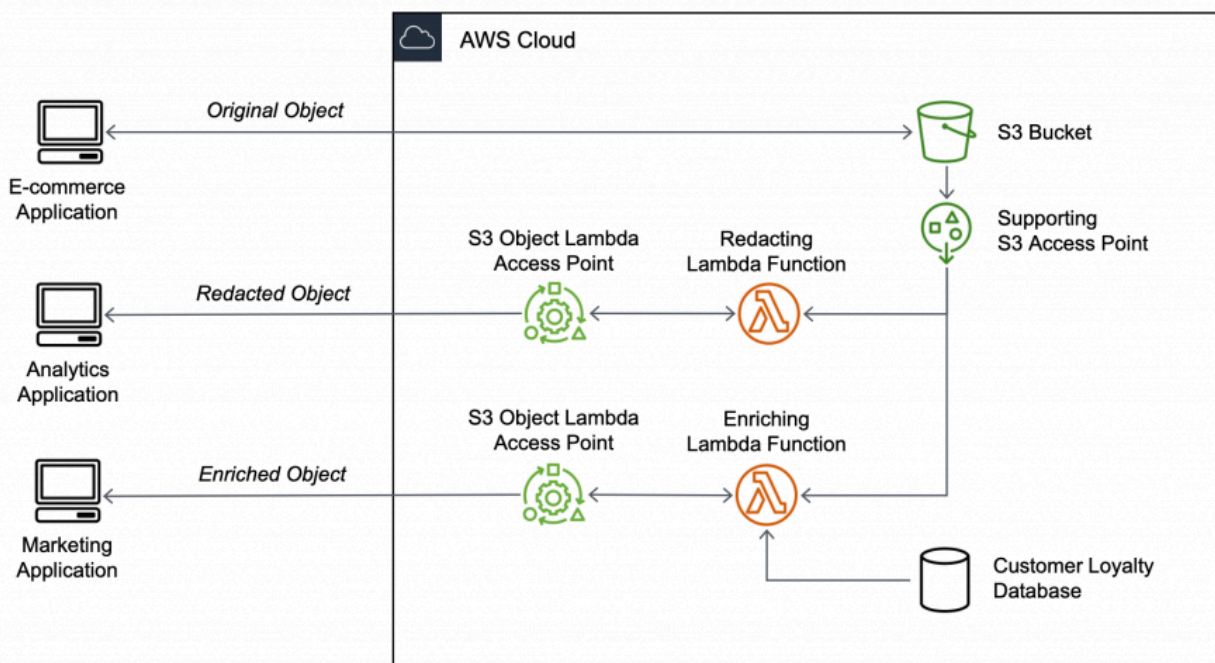
- [Tutorial: Verificar a integridade dos dados no Amazon S3 com somas de verificação adicionais](#)
- [Tutorial: Replicar dados em uma região ou entre Regiões da AWS usando a Replicação do S3](#)
- [Tutorial: Proteger dados no Amazon S3 contra exclusão acidental ou bugs de aplicações usando o Versionamento do S3, o Bloqueio de Objeto do S3 e a Replicação do S3](#)
- [Tutorial: replicar objetos existentes nos buckets do Amazon S3 com o S3 Batch Replication](#)

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 do Lambda especificada é chamada para processar quaisquer dados recuperados de um bucket do S3 por meio do ponto de acesso do S3 de suporte. Depois, 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](#)
- [Etapa 1: criar um bucket do S3](#)
- [Etapa 2: fazer upload do arquivo para seu bucket do S3](#)
- [Etapa 3: criar um ponto de acesso do S3](#)
- [Etapa 4: Criar uma função Lambda](#)
- [Etapa 5: configurar uma política do IAM para a função de execução da função Lambda](#)
- [Etapa 6: criar um ponto de acesso do S3 Object Lambda](#)
- [Etapa 7: exibir os dados transformados](#)
- [Etapa 8: limpar](#)
- [Próximas etapas](#)

Pré-requisitos

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

Subetapas

- [Criar um usuário do IAM com permissões em sua Conta da AWS \(console\)](#)
- [Instale o Python 3.8 ou posterior em sua máquina local](#)

Criar um usuário do IAM com permissões em sua Conta da AWS (console)

Você pode criar um usuário do IAM para o tutorial. Para concluir este tutorial, o usuário do IAM deve anexar as seguintes políticas do IAM para acessar recursos da AWS e executar ações específicas. Para obter mais informações sobre como criar um usuário do IAM, consulte [Criar usuários do IAM \(console\)](#) no Guia do usuário do IAM.

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.
- [CloudWatchLogsFullAccess](#): concede acesso total ao CloudWatch Logs.

Note

Para simplificar, este tutorial cria e usa um usuário do IAM. Depois de concluir este tutorial, lembre-se de [Excluir o usuário do IAM](#). Para uso em produção, recomendamos que você siga as [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM. Uma das práticas recomendadas exige que os usuários humanos usem a federação com um provedor de identidades para acessar a AWS usando credenciais temporárias. Outra prática recomendada é exigir que as workloads usem credenciais temporárias com perfis do IAM para acessar a AWS. Para saber como usar o AWS IAM Identity Center para criar usuários com credenciais temporárias, consulte [Getting started](#) (Conceitos básicos) no Guia do usuário do AWS IAM Identity Center.

Este tutorial também usa políticas de acesso total gerenciadas pela AWS. 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](#).

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](#).

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](#).

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](#).

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](#).

8. Selecione 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](#) (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](#). 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](#).

5. No campo Bucket name (Nome do bucket), insira o nome do bucket criado na [Etapa 1](#) (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](#).

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](#).

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](#).

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

Etapa 4: Criar uma função Lambda

Para transformar dados originais, crie uma função do 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](#)
- [Crie uma função Lambda com uma função de execução \(console\)](#)
- [Implante seu código de função Lambda com arquivos .zip e configure a função Lambda \(console\)](#)

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
from botocore.config import Config

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

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

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

    # Write object back to S3 Object Lambda
    s3 = boto3.client('s3', config=Config(signature_version='s3v4'))
    # The WriteGetObjectResponse API sends the transformed data
    # back to S3 Object Lambda and then to the user
    s3.write_get_object_response(
        Body=transformed_object,
        RequestRoute=request_route,
        RequestToken=request_token)

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

Note

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 solicitações GetObject no Lambda](#).

Quando você está gravando uma função do 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 do 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](#) e [Escrever funções do Lambda para pontos de acesso do S3 Object Lambda](#).

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 AWS SDK 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, 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:

```
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 do 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](#), 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 do 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 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 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](#) (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 o perfil de execução da função do Lambda, escolha a guia Permissions (Permissões). Depois, no menu Add Permissions (Adicionar permissões), 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 Anexar políticas.

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

Um ponto de acesso do S3 Object Lambda fornece a flexibilidade de chamar uma função do 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 do Lambda para chamar e fornecer o contexto do evento no formato JSON como parâmetros personalizados para o Lambda usar.

Para 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 Nome do Ponto de acesso Lambda do objeto, informe o nome que você deseja usar para o ponto de acesso Lambda do objeto (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](#) (por exemplo, **tutorial-access-point**) e, em seguida, escolha Choose supporting Access Point (Escolher ponto de acesso de suporte).
6. Para recuperar objetos do bucket do S3 para que a função do Lambda processe, selecione GetObject em S3 APIs (APIs do S3).
7. 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](#) (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](#).
8. Para Lambda function version (Versão da função Lambda), escolha \$LATEST (a versão mais recente da função Lambda que você criou na [Etapa 4](#)).
9. (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](#).

10. (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 do 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 Lambda do objeto que chamam a mesma função do Lambda, você pode configurar cargas úteis com diferentes parâmetros, estendendo, assim, a flexibilidade da função do Lambda.

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

11. (Opcional) Para Métricas de solicitação – opcional, escolha Desativar ou Ativar para adicionar o monitoramento do Amazon S3 ao ponto de acesso Lambda do objeto. As métricas de solicitação são cobradas na taxa padrão do Amazon CloudWatch. Para obter mais informações, consulte [Preço do CloudWatch](#).
12. 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.
13. 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](#)
- [Execute um script Python para imprimir os dados originais e transformados](#)

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 Pontos de acesso do Object Lambda, escolha o ponto de acesso do S3 Object Lambda que você criou na [Etapa 6](#) (por exemplo, **tutorial-object-lambda-accesspoint**).

4. Na guia Objetos do 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](#).

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](#) para recuperar dados do S3.

O exemplo de script Python a seguir imprime os dados originais do bucket do 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 Pontos de acesso Lambda de objeto, selecione o botão de opção à esquerda do ponto de acesso do S3 Object Lambda que você criou na [Etapa 6](#) (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 na máquina local para imprimir os dados originais (por exemplo, `tutorial.txt`) do bucket do S3 e os dados transformados (por exemplo, `tutorial.txt`) do ponto de acesso do S3 Object Lambda). Você pode usar o seguinte script de exemplo:

```
import boto3
from botocore.config import Config

s3 = boto3.client('s3', config=Config(signature_version='s3v4'))

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

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

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

7. 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](#) 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](#).

```
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](#)
- [Exclua o ponto de acesso do S3](#)
- [Exclua a função de execução de sua função Lambda](#)
- [Excluir a função Lambda](#)
- [Excluir o grupo de logs do CloudWatch](#)
- [Exclua o arquivo original no bucket de origem do S3](#)
- [Exclua o bucket de origem do S3](#)
- [Excluir o usuário do IAM](#)

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 Pontos de acesso Lambda de objeto, selecione o botão de opção à esquerda do ponto de acesso do S3 Object Lambda que você criou na [Etapa 6](#) (por exemplo, **tutorial-object-lambda-accesspoint**).
4. Escolha Excluir.
5. Confirme se deseja excluir o ponto de acesso do Lambda, inserindo o nome no campo de texto exibido e escolha 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](#) (por exemplo, **tutorial-access-point**) e escolha o botão de opção ao lado do nome do ponto de acesso.
4. Escolha 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 do 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](#) (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](#) (por exemplo, **tutorial-object-lambda-function**).
3. Escolha Ações e, em seguida, escolha 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](#) (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](#) (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 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](#) (por exemplo, **tutorial-bucket**).
4. Escolha 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 Excluir.

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](#).

- 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.

Para obter mais informações, consulte [Tutorial: Using S3 Object Lambda to dynamically watermark images as they are retrieved](#) (Tutorial: Como usar o S3 Object Lambda para colocar marca d'água em imagens de maneira dinâmica à medida que são recuperadas).

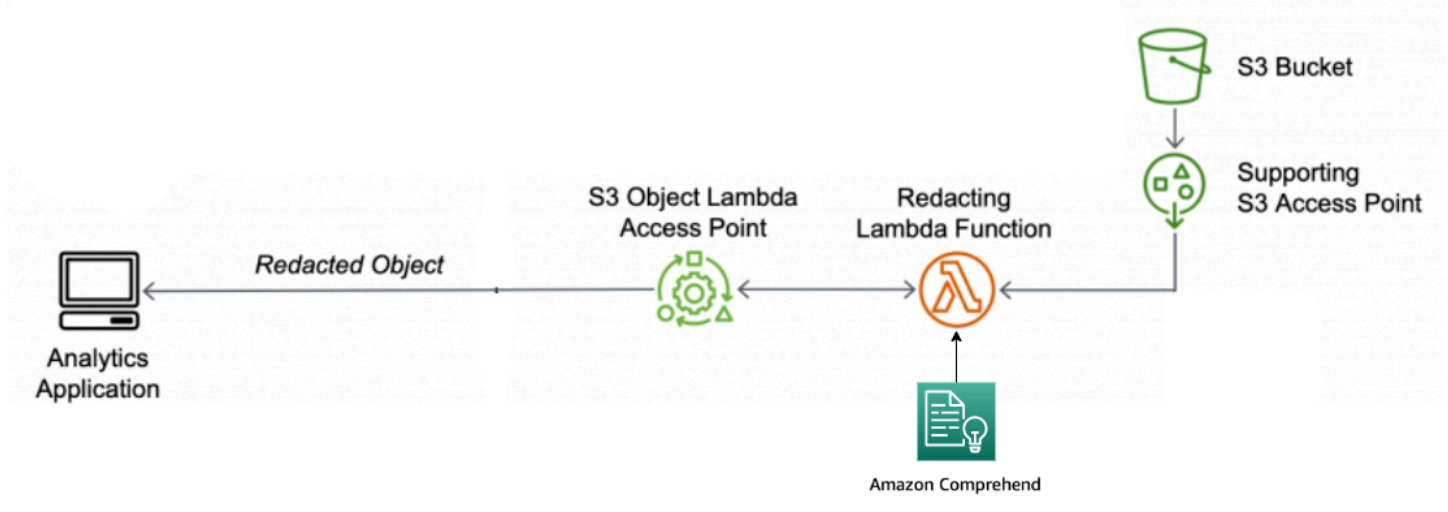
- Implementação de regras de autorização personalizadas para acessar dados.

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

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 do 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 do 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. Depois, 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 `ComprehendPiiRedactionS3ObjectLambda`. 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](#)
- [Etapa 1: criar um bucket do S3](#)
- [Etapa 2: fazer upload do arquivo para seu bucket do S3](#)
- [Etapa 3: criar um ponto de acesso do S3](#)
- [Etapa 4: configurar e implantar uma função Lambda pré-construída](#)
- [Etapa 5: criar um ponto de acesso do S3 Object Lambda](#)
- [Etapa 6: usar o ponto de acesso do S3 Object Lambda para recuperar o arquivo editado](#)
- [Etapa 7: limpar](#)
- [Próximas etapas](#)

Pré-requisitos: criar um usuário do IAM com permissões

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

Você pode criar um usuário do IAM para o tutorial. 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 cria e usa um usuário do IAM. Depois de concluir este tutorial, lembre-se de [Excluir o usuário do IAM](#). Para uso em produção, recomendamos que você siga as [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM. Uma das práticas recomendadas exige que os usuários humanos usem a federação com um provedor de identidades para acessar a AWS usando credenciais temporárias. Outra prática recomendada é exigir que as workloads usem credenciais temporárias com perfis do IAM para acessar a AWS. Para saber como usar o AWS IAM Identity Center para criar usuários com credenciais temporárias, consulte [Getting started](#) (Conceitos básicos) no Guia do usuário do AWS IAM Identity Center.

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](#).

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 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 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 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](#).

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](#).

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](#).

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](#).

8. Selecione 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](#) (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](#). 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](#).

5. No campo Bucket name (Nome do bucket), insira o nome do bucket criado na [Etapa 1](#) (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](#).

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](#).
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](#).
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, paraPiiEntityTypes, 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 Leiamos) 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 S3 Object Lambda

Um ponto de acesso do S3 Object Lambda fornece a flexibilidade de invocar uma função do 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 do 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](#).

Para 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 Nome do Ponto de acesso Lambda do objeto, informe o nome que você deseja usar para o ponto de acesso Lambda do objeto (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](#) (por exemplo, **tutorial-pii-access-point**) e, em seguida, escolha Choose supporting Access Point (Escolher ponto de acesso de suporte).
6. Para recuperar objetos do bucket do S3 para que a função do Lambda processe, selecione GetObject em S3 APIs (APIs do S3).
7. 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](#) (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](#).
8. Para Lambda function version (Versão da função Lambda), escolha \$LATEST (a versão mais recente da função Lambda que você implantou na [Etapa 4](#)).
9. (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](#).

10. (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 do 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 Lambda do objeto que chamam a mesma função do Lambda, você pode configurar cargas úteis com diferentes parâmetros, estendendo, assim, a flexibilidade da função do Lambda.

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

11. (Opcional) Para Métricas de solicitação – opcional, escolha Desativar ou Ativar para adicionar o monitoramento do Amazon S3 ao ponto de acesso Lambda do objeto. As métricas de solicitação

são cobradas na taxa padrão do Amazon CloudWatch. Para obter mais informações, consulte [Preço do CloudWatch](#).

12. 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.

13. 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 Pontos de acesso do Object Lambda, escolha o ponto de acesso do S3 Object Lambda que você criou na [Etapa 5](#) (por exemplo, **tutorial-pii-object-lambda-accesspoint**).
4. Na guia Objetos do 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](#).

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: limpar

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

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 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](#) (por exemplo, **tutorial-pii-object-lambda-accesspoint**).
4. Escolha Excluir.
5. Confirme se deseja excluir o ponto de acesso do Lambda, inserindo o nome no campo de texto exibido e escolha 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](#) (por exemplo, **tutorial-pii-access-point**) e escolha o botão de opção ao lado do nome do ponto de acesso.
4. Escolha 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](#) (por exemplo, **serverlessrepo-ComprehendPiiRedactionS3ObjectLambda**).
3. Escolha Ações e, em seguida, escolha 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](#) (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](#) (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 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](#) (por exemplo, **tutorial-bucket**).
4. Escolha 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](#) (por exemplo, **serverlessrepo-ComprehendPiiRedactionS3ObjectLambda**).
3. Escolha 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 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 Políticas.
3. Na página Policies (Políticas), insira o nome da política gerenciada pelo cliente que você criou em [Prerequisites](#) (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 Ações e, em seguida, escolha 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 Excluir.

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 do 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 um perfil do IAM e só tem acesso a um ponto de acesso do S3 Object Lambda (gerenciado por meio de políticas do IAM). Depois, anexe cada função do 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](#).

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

- 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.

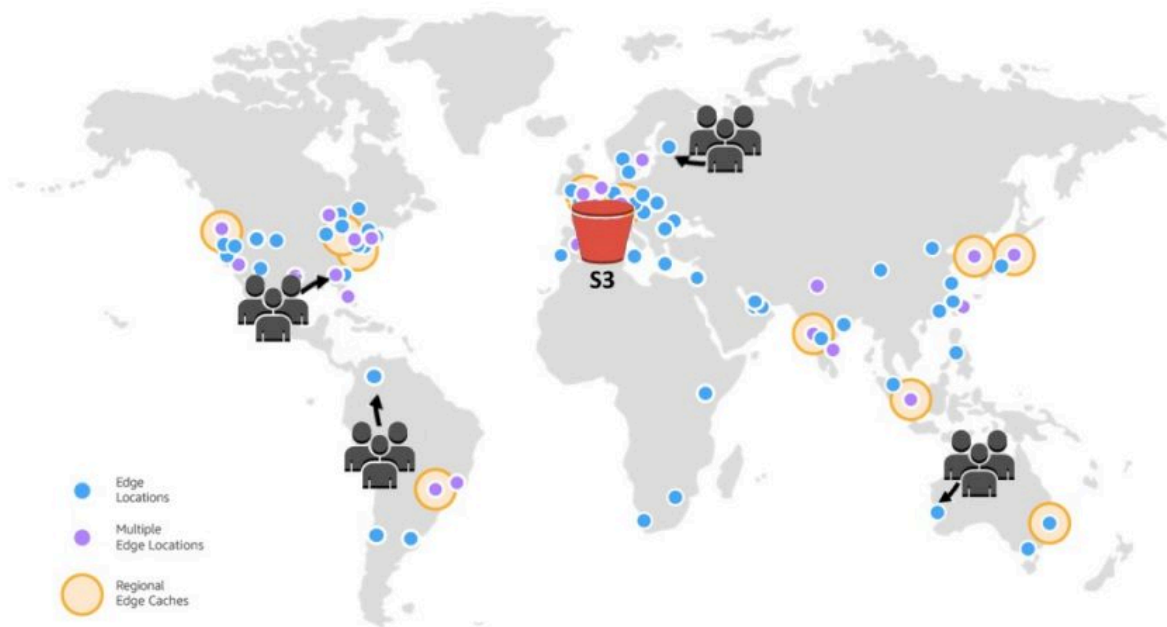
Para obter mais informações sobre como transformar dados com o S3 Object Lambda e sua própria função Lambda, consulte [Tutorial: Como transformar dados para sua aplicação com o S3 Object Lambda](#).

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](#)
- [Etapa 1: criar um bucket do S3](#)
- [Etapa 2: carregar um vídeo no bucket do S3](#)
- [Etapa 3: Criar uma identidade do acesso de origem do CloudFront](#)
- [Etapa 4: criar uma distribuição do CloudFront](#)
- [Etapa 5: Acessar o vídeo por meio da distribuição do CloudFront](#)
- [Etapa 6: Configurar sua distribuição do CloudFront para usar o seu nome de domínio personalizado](#)
- [Etapa 7: acessar o vídeo do S3 por meio da distribuição do CloudFront com o nome de domínio personalizado](#)

- [\(Opcional\) Etapa 8: exibir dados sobre solicitações recebidas pela distribuição do CloudFront](#)
- [Etapa 9: limpeza](#)
- [Próximas etapas](#)

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](#).

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](#).

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](#).

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](#).

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](#).

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](#) (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 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/v4/home>.
2. Na seção Segurança do painel de navegação esquerdo, escolha Acesso de origem.
3. Na guia Identidades, escolha Criar uma identidade de acesso de origem.
4. Insira um nome (por exemplo, **S3-OAI**) para a nova identidade do acesso de origem.
5. Escolha Criar.

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](#)
- [Revise a política de bucket](#)

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/v4/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](#) (por exemplo, **tutorial-bucket**).
5. Em Acesso de origem, selecione Identidades de acesso herdadas.
6. Em Origin access identity (Identidade do acesso de origem), escolha a identidade do acesso de origem que você criou na [Etapa 3](#) (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 aba 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/v4/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](#) (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 do navegador anterior 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](#) (por exemplo, **tutorial-bucket**).
8. Na lista Objects (Objetos), escolha o nome do vídeo que você carregou na [Etapa 2](#) (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 colocou 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](#)
- [Adicione um nome de domínio alternativo à distribuição do CloudFront](#)
- [Crie um registro do DNS para encaminhar o tráfego do seu nome de domínio alternativo para o nome de domínio da distribuição do CloudFront](#)
- [Verifique se o IPv6 está habilitado para sua distribuição e crie outro registro do DNS, se necessário](#)

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](#). 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 Revisar.
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/v4/home>.
2. No painel de navegação esquerdo, escolha Distributions (Distribuições).
3. Escolha o ID da distribuição que você criou na [Etapa 4](#).
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. Escolha Salvar alterações.
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 e hora 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](#) (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 Tipo de registro, selecione 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](#).

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](#) (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/v4/home>.
 - b. No painel de navegação esquerdo, escolha Distributions (Distribuições).
 - c. Escolha o ID da distribuição do CloudFront que você criou na [Etapa 4](#).
 - 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](#) (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](#).
- 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/v4/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](#) (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](#).
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/v4/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](#).

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](#)
- [Excluir o registro DNS](#)
- [Exclua a zona hospedada de seu domínio personalizado](#)
- [Exclua o nome de domínio personalizado do Route 53](#)
- [Exclua o vídeo original no bucket de origem do S3](#)
- [Exclua o bucket de origem do S3](#)

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/v4/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](#) (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](#) no Guia do desenvolvedor do Amazon Route 53. Se você só deseja excluir o registro do DNS criado na [Etapa 6](#), 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](#) (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](#)).

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 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 DNNSSEC, 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](#) (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 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](#) (por exemplo, **tutorial-bucket**).
4. Escolha 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).

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 a fim de transcodificar em lote um conjunto de vídeos para uma variedade de formatos de mídia de saída, consulte [Tutorial: Vídeos de transcodificação em lote com operações em lote do S3, AWS Lambda e AWS Elemental MediaConvert](#).

- 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 [Tutorial: Configurar um site estático usando um domínio personalizado registrado no Route 53](#) e [Acelerar seu site com o Amazon CloudFront](#).

- 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 ter mais informações, consulte [Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration](#).

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

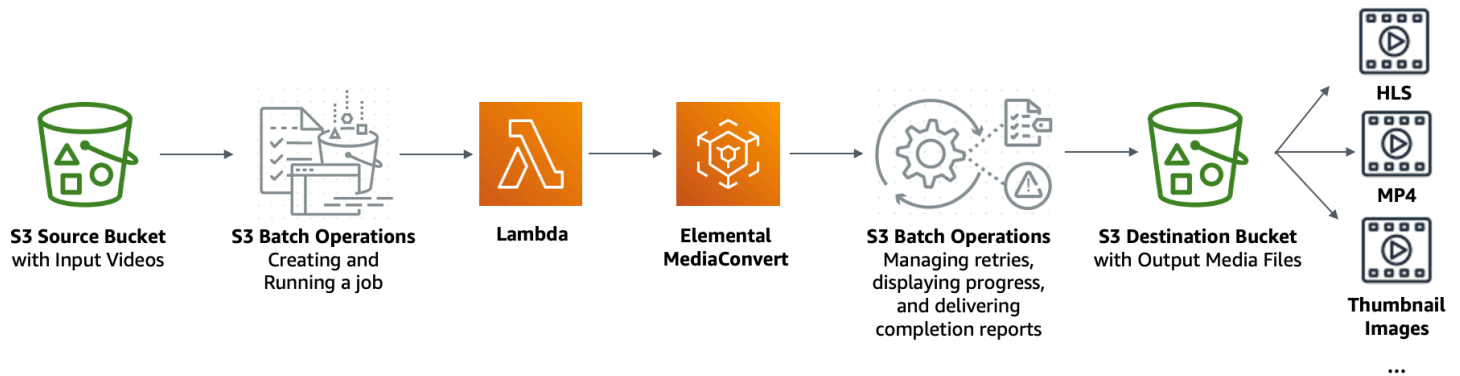
Os consumidores de vídeo usam dispositivos de todas as formas, tamanhos e épocas para desfrutar de conteúdo de mídia. Essa grande variedade de dispositivos representa um desafio para criadores e distribuidores de conteúdo. Em vez um formato de tamanho único, os vídeos precisam abranger uma ampla variedade 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. Depois, as funções do Lambda chamam o [AWS Elemental MediaConvert](#) para realizar 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](#)
- [Etapa 1: Criar um bucket do S3 para os arquivos de mídia de saída](#)
- [Etapa 2: criar uma função do IAM para MediaConvert](#)
- [Etapa 3: criar uma função do IAM para a função Lambda](#)
- [Etapa 4: Criar uma função do Lambda para transcodificação de vídeo](#)
- [Etapa 5: Configurar o inventário do Amazon S3 para seu bucket de origem do S3](#)
- [Etapa 6: criar uma função do IAM para operações em lote do S3](#)

- [Etapa 7: criar e executar um trabalho de operações em lote do S3](#)
- [Etapa 8: Conferir os arquivos de mídia de saída do bucket de destino do S3](#)
- [Etapa 9: limpar](#)
- [Próximas etapas](#)

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](#).

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](#).

Para obter mais informações sobre como carregar vídeos para o bucket de origem do S3, consulte [Fazer upload de objetos](#). 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](#).

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](#)
- [Para adicionar uma configuração CORS a um bucket de saída do S3](#)

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 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](#) 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 bloqueie todo o acesso público aos seus 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](#).

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 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 aba 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](#).

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

7. Escolha Salvar alterações.

Etapa 2: criar uma função do IAM para MediaConvert

Para usar o AWS Elemental MediaConvert para transcodificar vídeos de entrada armazenados no bucket do S3, você deve ter um perfil de serviço do AWS Identity and Access Management (IAM)

para conceder ao MediaConvert permissões para ler e gravar arquivos de vídeo 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 função do MediaConvert no IAM \(console\)](#) no Guia 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 [Amazon Resource Names \(ARNs\)](#) (Nomes de recurso da Amazon (ARNs) em AWS General Reference (Referência geral)).

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](#)
- [Incorpore uma política em linha para a função do IAM da sua função Lambda](#)

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 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 Próximo: etiquetas.
7. (Opcional) Adicione etiquetas à política gerenciada.
8. Escolha Next: Review (Próximo: revisar).
9. Em Nome do perfil, 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](#) (por exemplo, **tutorial-mediaconvert-role**).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
```

```
        "logs:CreateLogStream",
        "logs:PutLogEvents"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "Logging"
},
{
    "Action": [
        "iam:PassRole"
    ],
    "Resource": [
        "arn:aws:iam::111122223333:role/tutorial-mediaconvert-role"
    ],
    "Effect": "Allow",
    "Sid": "PassRole"
},
{
    "Action": [
        "mediaconvert:*"
    ],
    "Resource": [
        "*"
    ],
    "Effect": "Allow",
    "Sid": "MediaConvertService"
},
{
    "Action": [
        "s3:*"
    ],
    "Resource": [
        "*"
    ],
    "Effect": "Allow",
    "Sid": "S3Service"
}
]
}
```

7. Escolha Revisar política.
8. Em Nome, insira **tutorial-lambda-policy**.
9. Escolha 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](#)
- [Crie uma função Lambda com uma função de execução \(console\)](#)
- [Implante sua função Lambda com arquivos .zip e configure a função Lambda \(console\)](#)

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 bitrate 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 [Exemplos de configuração de trabalhos do AWS Elemental MediaConvert em JSON](#) no Guia 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",
    "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](#).

```

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

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

Input parameter "event": The S3 Batch Operations event as a request

```

for the Lambda function.

Input parameter "context": Context about the event.

Output: A result structure that Amazon S3 uses to interpret the result of the operation. It is a job response returned back to S3 Batch Operations.

```
"""
```

```
def handler(event, context):
```

```
    invocation_schema_version = event['invocationSchemaVersion']
```

```
    invocation_id = event['invocationId']
```

```
    task_id = event['tasks'][0]['taskId']
```

```
    source_s3_key = event['tasks'][0]['s3Key']
```

```
    source_s3_bucket = event['tasks'][0]['s3BucketArn'].split(':::')[0]
```

```
    source_s3 = 's3://' + source_s3_bucket + '/' + source_s3_key
```

```
    result_list = []
```

```
    result_code = 'Succeeded'
```

```
    result_string = 'The input video object was converted successfully.'
```

```
    # The type of output group determines which media players can play
```

```
    # the files transcoded by MediaConvert.
```

```
    # For more information, see Creating outputs with AWS Elemental MediaConvert.
```

```
    output_group_type_dict = {
```

```
        'HLS_GROUP_SETTINGS': 'HlsGroupSettings',
```

```
        'FILE_GROUP_SETTINGS': 'FileGroupSettings',
```

```
        'CMAF_GROUP_SETTINGS': 'CmafGroupSettings',
```

```
        'DASH_ISO_GROUP_SETTINGS': 'DashIsoGroupSettings',
```

```
        'MS_SMOOTH_GROUP_SETTINGS': 'MsSmoothGroupSettings'
```

```
    }
```

```
    try:
```

```
        job_name = 'Default'
```

```
        with open('job.json') as file:
```

```
            job_settings = json.load(file)
```

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

```
        # The path of each output video is constructed based on the values of
```

```
        # the attributes in each object of OutputGroups in the job.json file.
```

```
        destination_s3 = 's3://{0}/{1}/{2}' \
```

```
            .format(os.environ['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
}
```

4. 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)

1. 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-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 Exiting role (Função de saída), escolha o nome da função do IAM que você criou para sua função do Lambda na [Etapa 3](#) (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 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 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](#).

- 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](#). 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 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. Selecione 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 [Preços 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](#)
- [Configure o inventário do Amazon S3 para o bucket de origem de vídeo do S3](#)
- [Confira o relatório de inventário para o seu bucket de origem de vídeo S3](#)

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 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 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. Escolha Salvar alterações.

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](#).

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 [Configurar um inventário usando o console do S3 e Conceder permissão para que o Amazon S3 use a sua chave gerenciada pelo cliente para criptografia](#).

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).

Para obter mais informações, consulte [Configurar um inventário usando o console do S3](#).

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](#), 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](#).

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 `manifesto.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](#)
- [Crie uma função do IAM de operações em lote do S3 e anexe políticas de permissão](#)

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 Políticas.
3. Escolha 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](#) para armazenar os arquivos `manifest.json` (por exemplo, `tutorial-bucket-3`).
- O nome do bucket que você criou na [Etapa 1](#) 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](#). 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](#) (por exemplo, `tutorial-lambda-convert`).
 - Escolha Copy ARN (Copiar ARN).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3Get",
      "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 Próximo: etiquetas.
7. Selecione Next: Review (Próximo: revisar).
8. No campo Name (Nome), insira **tutorial-s3batch-policy**.
9. Escolha 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 Próximo: etiquetas.
8. Escolha Next: Review (Próximo: revisar).
9. Em Nome do perfil, 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](#). 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](#), a entrega do primeiro relatório de inventário pode levar até 48 horas.

Subetapas

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

- [\(Opcional\) Monitore cada trabalho de transcodificação de vídeo MediaConvert no console do MediaConvert](#)

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 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](#) 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 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](#) (por exemplo, **tutorial-lambda-convert**).
 - c. Para Lambda function version (Versão da função do Lambda), mantenha o valor padrão \$LATER.
9. Escolha 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](#).

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](#) (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](#). 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](#) (por exemplo, **tutorial-s3batch-role**).
13. Escolha 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 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](#) (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](#) (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: limpar

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](#)
- [Excluir a função Lambda](#)
- [Excluir o grupo de logs do CloudWatch](#)

- [Exclua as funções do IAM junto com as políticas em linha das funções do IAM](#)
- [Exclua a política do IAM gerenciada pelo cliente](#)
- [Esvaziar os buckets do S3](#)
- [Excluir os buckets do S3](#)

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](#) (por exemplo, **tutorial-inventory-config**).
6. Escolha Delete (Excluir) e, em seguida, escolha Confirm (Confirmar).

Excluir a função Lambda

1. Abra o console 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](#) (por exemplo, **tutorial-lambda-convert**).
4. Escolha Ações e, em seguida, escolha Excluir.
5. Na caixa de diálogo Delete function (Excluir função), escolha Delete (Excluir).

Excluir o grupo de logs do CloudWatch

1. Abra o console 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](#) (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](#), [Etapa 3](#) e [Etapa 6](#), 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](#), 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 Políticas.
3. Escolha o botão de opção ao lado da política que você criou na [Etapa 6](#) (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](#), [Etapa 1](#) e [Etapa 5](#), 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](#), [Etapa 1](#) e [Etapa 5](#), 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 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).

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](#).
- 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

Important

O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).

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.

Important

O tutorial a seguir exige que o Bloqueio de Acesso Público seja desabilitado. Recomendamos manter o Bloqueio de Acesso Público habilitado. Se quiser manter todas as quatro configurações do Bloqueio de Acesso Público habilitadas e hospedar um site estático, use o controle de acesso à origem (OAC) do Amazon CloudFront. O Amazon CloudFront fornece os recursos necessários para configurar um site estático seguro. Os sites estáticos do Amazon S3 só são compatíveis com endpoints HTTP. O Amazon 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.

Tópicos

- [Etapa 1: Criar um bucket](#)
- [Etapa 2: Habilitar hospedagem de site estático](#)

- [Etapa 3: editar as configurações do Bloqueio de acesso público](#)
- [Etapa 4: Adicionar política de bucket que torna o conteúdo do bucket publicamente disponível](#)
- [Etapa 5: Configurar um documento de índice](#)
- [Etapa 6: configurar um documento de erros](#)
- [Etapa 7: testar o endpoint do site](#)
- [Etapa 8: Limpar](#)

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](#).

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](#).

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](#).

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 ter mais informações, consulte [Configurar um documento de erro personalizado](#).

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](#).

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](#) 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 bloqueie 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](#) 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 bloqueie todo o acesso público aos seus 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 Documentos 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](#).

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](#).

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](#) e [Configurar um documento de erro personalizado](#).

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](#).

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 o 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 [Tutorial: Configurar um site estático usando um domínio personalizado registrado no Route 53](#).

Etapa 8: Limpar

Se tiver criado o site apenas como exercício de aprendizado, você poderá excluir os recursos da AWS alocados, de maneira que deixe de acumular cobranças. Depois que você excluir os recursos da AWS, o site deixará de estar disponível. Para ter mais informações, consulte [Excluir um bucket](#).

Tutorial: 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](#).

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](#)
- [Etapa 1: Registrar um domínio personalizado no Route 53](#)
- [Etapa 2: Criar dois buckets](#)
- [Etapa 3: Configurar o bucket de domínio raiz para hospedagem de sites](#)

- [Etapa 4: Configurar o bucket de subdomínio para redirecionamento de sites](#)
- [Etapa 5: Configurar o registro em log para o tráfego do site](#)
- [Etapa 6: Fazer upload do conteúdo do site e do índice](#)
- [Etapa 7: carregar um documento de erros](#)
- [Etapa 8: Editar configurações do S3 Block Public Access](#)
- [Etapa 9: Anexar uma política de bucket](#)
- [Etapa 10: Testar o endpoint de domínio](#)
- [Etapa 11: Adicionar registros de alias para seu domínio e subdomínio](#)
- [Etapa 12: Testar o site](#)
- [Acelerar seu site com o Amazon CloudFront](#)
- [Limpar seus recursos de exemplo](#)

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](#).

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](#).

- 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](#).

- 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](#).

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](#).

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 ter mais informações, consulte [Configurar um documento de erro personalizado](#).

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](#).

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](#). 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`).
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`).

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 Imposto pelo proprietário do bucket para a Propriedade de Objetos do S3 a fim de desabilitar ACLs, o CloudFront não poderá gravar logs no bucket. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

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 Documentos 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](#).

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](#).

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](#) e [Configurar um documento de erro personalizado](#).

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](#).

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](#) 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 bloqueie 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](#) 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 bloqueie todo o acesso público aos seus 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](#). 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 o 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 .

- 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.

- 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).
- Em Evaluate target health (Avaliar integridade do destino), escolha No (Não).
- Escolha Define simple record (Definir registro simples).
- 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 reprojeto. 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.

- 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 cola 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](#).

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.

Usar o CloudFront sem o Route 53

Os tutoriais nesta página usam o Route 53 para apontar para sua distribuição do CloudFront. No entanto, se você quiser servir conteúdo hospedado em um bucket do Amazon S3 usando o CloudFront sem usar o Route 53, consulte [Tutoriais do Amazon CloudFront: configurar uma distribuição dinâmica de conteúdo para o Amazon S3](#). Ao servir conteúdo hospedado em um bucket do Amazon S3 usando o CloudFront, você pode usar qualquer nome de bucket, e tanto HTTP quanto HTTPS são compatíveis.

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](#)
- [Etapa 2: Atualizar os conjuntos de registros do domínio e do subdomínio](#)
- [\(Opcional\) Etapa 3: verificar os arquivos de log](#)

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/v4/home>.


2. Escolha Criar distribuição.
3. 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.

4. 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.

5. 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 ter mais informações, consulte [Configurar um documento de índice](#).

- 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 Imposto pelo proprietário do bucket para a Propriedade de Objetos do S3 a fim de desabilitar ACLs, o CloudFront não poderá gravar logs no bucket. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

- 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](#).

- 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.

- 6. Escolha Criar distribuição.

- 7. 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.

- 8. Registre o valor de Domain Name (Nome do domínio) mostrado no console do CloudFront, por exemplo, `dj4p1rv6mvubz.cloudfront.net`.
- 9. 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](#). 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](#)
- [Etapa 2: exclua a zona hospedada do Route 53](#)
- [Etapa 3: Desabilitar o registro em log e excluir o bucket do S3](#)

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/v4/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 ter mais informações, consulte [Excluir um bucket](#).

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

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

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](#).

Tópicos

- [Visão geral dos buckets](#)
- [Regras de nomeação de bucket](#)
- [Acessar e listar um bucket do Amazon S3](#)
- [Criação de um bucket](#)

- [Visualização das propriedades de um bucket do S3](#)
- [Esvaziar um bucket](#)
- [Excluir um bucket](#)
- [Definir o comportamento padrão da criptografia para os buckets do Amazon S3](#)
- [Trabalhar com o Mountpoint para Amazon S3](#)
- [Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration](#)
- [Configuração de buckets de Pagamento pelo solicitante para transferências de armazenamento e uso](#)
- [Restrições e limitações do bucket](#)

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, acesse o [Console do Service Quotas](#).

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](#) (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](#).

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](#).

O Amazon S3 cria buckets na região que você especificar. Para reduzir 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

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

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](#)
- [Gerenciar o acesso público aos buckets](#)
- [Opções de configuração do bucket](#)

Sobre permissões

É possível usar suas credenciais de Usuário raiz 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 raiz 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.

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 [Gerenciamento de identidade e acesso para o Amazon S3](#).

Gerenciar o acesso público aos buckets

O acesso público aos buckets e objetos é concedido por meio de políticas de bucket, listas de controle de acesso (ACLs) 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 garantir que todos os seus buckets e objetos do Amazon S3 tenham o acesso público bloqueado, todas as quatro configurações do Bloqueio de Acesso Público devem estar habilitadas por padrão ao criar um bucket. Recomendamos ativar todas as quatro configurações do Bloqueio de Acesso Público na sua conta também. 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 nível determinado de acesso público aos buckets ou objetos, por exemplo, para hospedar um site estático como descrito em [Hospedagem de um site estático](#)

[usando o Amazon S3](#), é possível personalizar as configurações individualmente para atender aos seus casos de uso de armazenamento. Para ter mais informações, consulte [Bloquear o acesso público ao armazenamento do Amazon S3](#).

No entanto, é altamente recomendável manter o Bloqueio de Acesso Público habilitado. Se quiser manter todas as quatro configurações do Bloqueio de Acesso Público habilitadas e hospedar um site estático, use o controle de acesso à origem (OAC) do Amazon CloudFront. O Amazon CloudFront fornece os recursos necessários para configurar um site estático seguro. Os sites estáticos do Amazon S3 só são compatíveis com endpoints HTTP. O Amazon 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.

Note

Se você vir um `ERROR` ao listar seus buckets e as respectivas configurações de acesso público, talvez você não tenha as permissões necessárias. Verifique 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)	<p>Você pode configurar seu bucket para autorizar solicitações de origem cruzada.</p> <p>Para obter mais informações, consulte Usar o compartilhamento de recursos de origem cruzada (CORS).</p>
notificação de evento	<p>Você pode permitir que seu bucket envie notificações de eventos do bucket especificado.</p> <p>Para ter mais informações, consulte Notificações de eventos do Amazon S3.</p>
ciclo de vida	<p>Você pode definir regras de ciclo de vida para objetos em seu bucket que têm um ciclo de vida bem definido. Por exemplo, você pode definir uma regra para arquivar objetos um ano após a criação ou excluir um objeto 10 anos após a criação.</p> <p>Para obter mais informações, consulte Gerenciando seu ciclo de vida de armazenamento.</p>
location	<p>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.</p>

Sub-recurso	Descrição
registro em log	<p>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.</p> <p>Para obter mais informações, consulte Registrar em log as solicitações com registro em log de acesso ao servidor.</p>
bloqueio de objetos	<p>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.</p> <p>Para ter mais informações, consulte Usar o bloqueio de objetos do S3.</p>
política e ACL (lista de controle de acesso)	<p>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.</p> <p>Para ter mais informações, consulte Gerenciamento de identidade e acesso para o Amazon S3.</p>
replicação	<p>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>

Sub-recurso	Descrição
requestPayment	<p>Por padrão, a Conta da AWS que cria o bucket (o proprietário do bucket) paga pelos downloads do bucket. Usando esse sub-recurso, o proprietário do bucket pode especificar que a pessoa que solicita o download será cobrada pelo download. O Amazon S3 fornece uma API para gerenciamento desse sub-recurso.</p> <p>Para obter mais informações, consulte Configuração de buckets de Pagamento pelo solicitante para transferências de armazenamento e uso.</p>
marcação	<p>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.</p> <p>Para obter mais informações, consulte Relatórios de uso e faturamento do Amazon S3.</p>
aceleração de transferência	<p>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.</p> <p>Para obter mais informações, consulte Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration.</p>
versionamento	<p>O versionamento ajuda a recuperar substituições e exclusões acidentais.</p> <p>Recomendamos o versionamento como melhor prática para impedir a exclusão ou substituição de objetos por engano.</p> <p>Para obter mais informações, consulte Usando o versionamento em buckets do S3.</p>

Sub-recurso	Descrição
site	<p>Você pode configurar seu bucket para hospedagem de site estático. O Amazon S3 armazena essa configuração criando um sub-recurso site.</p> <p>Para ter mais informações, consulte Hospedagem de um site estático usando o Amazon S3.</p>

Regras de nomeação de bucket

As seguintes regras se aplicam à nomenclatura de buckets de uso geral e buckets de diretório no Amazon S3:

Tópicos

- [Regras de nomenclatura de buckets de uso geral](#)
- [Regras de nomenclatura de buckets de diretório](#)

Regras de nomenclatura de buckets de uso geral

As seguintes regras de nomenclatura se aplicam à buckets de uso geral:

- 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 hífen (-).
- 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 do bucket não devem iniciar com os prefixos sthree- e sthree-configurator.
- Os nomes de bucket não podem terminar com o sufixo -s3alias. Esse sufixo se reserva a nomes de alias de ponto de acesso. Para ter mais informações, consulte [Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3](#).
- Os nomes de bucket não podem terminar com o sufixo --o1-s3. Esse sufixo é reservado a nomes de alias de ponto de acesso de objeto do Lambda. Para ter mais informações, consulte

[Como usar um alias no estilo do bucket para o ponto de acesso do Object Lambda do bucket do S3.](#)

- 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](#).

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](#). Para obter mais informações sobre hospedagem de sites estáticos, consulte [Hospedagem de um site estático usando o Amazon S3](#).

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](#).

Exemplos de nomes de buckets de uso geral

Os seguintes exemplos de nomes de buckets 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)

Regras de nomenclatura de buckets de diretório

Os nomes dos buckets de diretório devem:

- Ser únicos dentro da Região da AWS e da zona de disponibilidade escolhidas.
- Ter entre 3 e 63 caracteres, incluindo o sufixo.
- Consistir em letras minúsculas, números e hifens (-).
- Inicie e termine com uma letra ou um número.
- Deve incluir o seguinte sufixo: `--azid--x-s3`.

Note

Quando você cria um bucket de diretório usando o console, um sufixo é adicionado automaticamente ao nome base que você fornece. Esse sufixo inclui o ID da zona de disponibilidade que você escolheu.

Ao criar um bucket de diretório usando uma API, você deve fornecer o sufixo completo, incluindo o ID da zona de disponibilidade, na solicitação. Consulte uma lista de IDs de zona de disponibilidade em [Zonas de disponibilidade e regiões da classe S3 Express One Zone](#).

Acessar e listar um bucket do Amazon S3


Para listar e acessar seus buckets do Amazon S3, você pode usar várias ferramentas. Analise as seguintes ferramentas para determinar qual abordagem se adapta ao seu caso de uso:

- **Console do Amazon S3:** com o console do Amazon S3, você pode acessar um bucket e modificar as propriedades. Você também pode executar a maioria das operações de bucket usando a IU do console, sem precisar escrever nenhum código.
- **AWS CLI:** caso precise acessar vários buckets, você pode economizar tempo usando AWS Command Line Interface (AWS CLI) para automatizar tarefas comuns e repetitivas. A capacidade de script e a consistência para ações comuns são considerações frequentes à medida que as organizações crescem. Para ter mais informações, consulte [Desenvolvimento com o Amazon S3 usando a AWS CLI](#).
- **API REST do Amazon S3:** você pode usar a API REST do Amazon S3 para escrever os próprios programas e acessar buckets de maneira programática. O Amazon S3 oferece suporte a uma arquitetura de API na qual buckets e objetos são recursos, cada um com um URI de recurso que identifica exclusivamente o recurso. Para ter mais informações, consulte [Desenvolver com o Amazon S3 usando a API REST](#).

Dependendo do caso de uso do seu bucket do Amazon S3, existem diferentes métodos recomendados para acessar os dados subjacentes nos seus buckets. A lista a seguir inclui casos de uso comuns para acessar seus dados.

- **Sites estáticos** — você pode usar o Amazon S3 para hospedar um site estático. Neste caso de uso, você pode configurar o bucket do S3 para funcionar como um site. Para obter orientação nas etapas de hospedagem de um site no Amazon S3, consulte [Tutorial: configurar um site estático no Amazon S3](#).

Para hospedar um site estático com configurações de segurança como Bloquear acesso público habilitado, recomendamos usar o Amazon CloudFront com Origin Access Control (OAC) e implementar cabeçalhos de segurança adicionais, como HTTPS. Para obter mais informações, consulte [Conceitos básicos de site estático seguro](#).

 Note

O Amazon S3 oferece suporte tanto ao URL de [estilo hospedado virtual](#) quanto de [estilo de caminho](#) para acesso estático a sites. 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 ter mais informações, consulte [Restrições e limitações do bucket](#).

- Conjuntos de dados compartilhados: à medida que você escala no Amazon S3, é comum adotar um modelo multilocatário, no qual você atribui diferentes clientes finais ou unidades de negócios a prefixos exclusivos em um bucket compartilhado. Ao usar os [pontos de acesso Amazon S3](#), você pode dividir uma política de grande bucket em políticas de ponto de acesso separadas e discretas para cada aplicativo que precisa acessar o conjunto de dados compartilhado. Essa abordagem simplifica o foco na criação da política de acesso correta para um aplicativo, sem interromper o que qualquer outro aplicativo está fazendo no conjunto de dados compartilhado. Para ter mais informações, consulte [Gerenciamento de acesso a dados com pontos de acesso do Amazon S3](#).
- Workloads de alto throughput — O Mountpoint para Amazon S3 é um cliente de arquivo de código aberto de alto throughput para montar um bucket do Amazon S3 como um sistema de arquivos local. Com o Mountpoint, os aplicativos podem acessar arquivos armazenados no Amazon S3 por meio de operações do sistema de arquivos, como abrir e ler. O Mountpoint traduz automaticamente essas operações em chamadas de API de objetos do S3, dando aos aplicativos acesso ao armazenamento elástico e ao throughput do Amazon S3 por meio de uma interface de arquivo. Para ter mais informações, consulte [Trabalhar com o Mountpoint para Amazon S3](#).
- Aplicativos multirregionais — os pontos de acesso multirregionais do Amazon S3 fornecem um endpoint global que os aplicativos podem usar para atender solicitações de buckets S3 localizados em várias Regiões da AWS. Você pode usar pontos de acesso multirregionais para criar aplicações de várias regiões com a mesma arquitetura usada em uma única região, depois executar essas aplicações em qualquer lugar do mundo. Em vez de enviar solicitações pela Internet pública, os pontos de acesso multirregionais fornecem resiliência de rede integrada com aceleração de solicitações baseadas na Internet para o Amazon S3. Para ter mais informações, consulte [Pontos de acesso multirregionais no Amazon S3](#).

- Criação de novos aplicativos — você pode usar os AWS SDKs ao desenvolver aplicativos com o Amazon S3. Os AWS SDKs simplificam suas tarefas de programação agrupando a API REST subjacente do Amazon S3. Para criar aplicativos para dispositivos móveis e online conectados, você pode usar os AWS Mobile SDKs e a biblioteca AWS Amplify JavaScript. Para ter mais informações, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers](#).
- Protocolo de transferência de arquivos (SFTP) Secure Shell (SSH): se estiver tentando transferir dados confidenciais com segurança pela Internet, você pode usar um servidor habilitado para SFTP com seu bucket do Amazon S3. AWS SFTP é um protocolo de rede compatível com toda a funcionalidade de segurança e autenticação do SSH. Com esse protocolo, você tem controle aprimorado sobre a identidade, as permissões e as chaves do usuário ou pode usar políticas do IAM para gerenciar o acesso. Para associar um servidor habilitado para SFTP ao bucket do Amazon S3, crie o servidor habilitado para SFTP. Em seguida, você configura contas de usuário e associa o servidor a um bucket do Amazon S3. Para obter instruções sobre esse processo, consulte [AWS Transfer for SFTP — Serviço SFTP totalmente gerenciado para Amazon S3](#) em AWS Blogs.

Listar buckets

Para listar todos os buckets, você precisa ter a permissão `s3:ListAllMyBuckets`. Para acessar um bucket, tenha também as permissões AWS Identity and Access Management (IAM) necessárias para listar o conteúdo do bucket especificado. Para obter um exemplo de política de bucket que concede acesso a um bucket do S3, consulte [Permitir que um usuário do IAM acesse um dos seus buckets](#). Se você encontrar um erro HTTP — Acesso Negado (403 Proibido), consulte [Políticas de bucket e políticas do IAM](#).

Você pode listar seu bucket usando o console do Amazon S3, o AWS CLI ou os AWS SDKs.

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 Buckets.
3. Na lista Buckets de uso geral, selecione o bucket que você deseja visualizar.

Note

A lista de Buckets de uso geral inclui buckets que estão localizados em todas as Regiões da AWS.

Uso do AWS CLI

Para usar o AWS CLI para acessar um bucket do S3 ou gerar uma lista dos buckets do S3, use o comando `ls`. Ao listar todos os objetos no bucket, observe que você deve ter a permissão `s3:ListBucket`.

Para usar este comando de exemplo, substitua *DOC-EXAMPLE-BUCKET1* pelo nome do bucket.

```
$ aws s3 ls s3://DOC-EXAMPLE-BUCKET1
```

O comando de exemplo a seguir mostra todos os buckets do Amazon S3 em sua conta:

```
$ aws s3 ls
```

Para obter mais informações e exemplos, consulte [Listar buckets e objetos](#).

Uso de AWS SDKs

Você também pode acessar um bucket do Amazon S3 usando a operação de API [ListBuckets](#). Para obter exemplos de como usar essa operação com diferentes AWS SDKs, consulte [Usar ListBuckets com um SDK da AWS ou uma ferramenta de linha de comandos](#).

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](#).

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 enviar um aumento de limite de bucket, consulte [Cotas da AWS service \(Serviço da AWS\)](#) na Referência geral da AWS. Você pode armazenar qualquer número de objetos no bucket.

A Propriedade de objeto do S3 é uma configuração no nível do bucket do Amazon S3 que você pode usar para controlar a propriedade de objetos carregados no bucket e para desabilitar ou habilitar as listas de controle de acesso (ACLs). Por padrão, a Propriedade de Objetos está definida com a configuração Imposto pelo proprietário do bucket e todas as ACLs estão desabilitadas. Com as ACLs desabilitadas, o proprietário do bucket possui cada objeto no bucket e gerencia o acesso aos dados exclusivamente usando políticas.

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

A criptografia no lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) é o nível básico de configuração de criptografia para cada bucket no Amazon S3. Todos os novos objetos carregados em um bucket do S3 são automaticamente criptografados com o SSE-S3 como nível básico da configuração de criptografia. Se quiser usar um tipo diferente de criptografia padrão, você também poderá especificar a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS) ou chaves fornecidas pelo cliente (SSE-C) a fim de criptografar seus dados. Para ter mais informações, consulte [Definir o comportamento padrão da criptografia para os buckets do Amazon S3](#).

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. Na barra de navegação na parte superior da página, escolha o nome da Região da AWS exibida no momento. Em seguida, escolha a região na qual você deseja criar um bucket.

Note

Para reduzir a latência e os custos e atender aos requisitos regulatórios, selecione uma região perto de você. Os objetos armazenados em uma região nunca saem dessa região, a menos que você os transfira para outra região. Para obter uma lista de Regiões

da AWS do Amazon S3, consulte [Endpoints de AWS service \(Serviço da AWS\)](#) no Referência geral da Amazon Web Services.

3. No painel de navegação à esquerda, escolha Buckets.
4. Selecione Create bucket (Criar bucket).

A página Create bucket (Criar bucket) é aberta.

5. Em Configuração geral, visualize a Região da AWS onde o bucket será criado.
6. Em Tipo de bucket, escolha Propósito geral.
7. Para Nome do bucket, insira um nome para o bucket.

O nome do bucket deve:

- Seja exclusivo em 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) Regions).
- Ter entre 3 e 63 caracteres.
- Consistem apenas em letras minúsculas, números, pontos (.) e hífens (-). 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.
- Inicie e termine com uma letra ou um número.

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

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.

8. O AWS Management Console permite que você copie as configurações de um bucket existente para seu novo bucket. Prossiga para a próxima etapa se você não quiser copiar as configurações de um bucket existente.

Note

Essa opção:

- Não está disponível na AWS CLI, mas apenas no console.
- Não está disponível para buckets de diretório.
- Não copia a política do bucket existente para o novo bucket.

Para copiar as configurações de um bucket existente, em Copiar configurações do bucket existente, selecione Escolher bucket. A janela Escolher bucket é aberta. Encontre o bucket com as configurações que você gostaria de copiar e selecione Escolher bucket. A janela Escolher bucket é fechada e a janela Criar bucket é reaberta.

Agora, em Copiar configurações do bucket existente, você verá o nome do bucket selecionado. Você também verá uma opção Restaurar padrões, que pode ser usada para remover as configurações do bucket copiado. Revise as configurações restantes do bucket na página Criar bucket. Você verá que agora elas correspondem às configurações do bucket que você selecionou. Você pode pular para a etapa final.

9. 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

- Imposto pelo proprietário do bucket (padrão): 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 de acesso aos dados no bucket do S3. O bucket usa políticas exclusivamente para definir o controle de acesso.


Por padrão, as ACLs estão desabilitadas. A maioria dos casos de uso modernos no Amazon S3 não exige mais o uso de ACLs. Recomendamos manter as ACLs desabilitadas, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

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 Proprietário do bucket preferido para exigir que todos os uploads do Amazon S3 incluam a ACL `bucket-owner-full-control` pré-configurada, será possível [adicionar uma política de bucket](#) que só permita uploads de objetos que usem 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

A configuração padrão é imposto pelo proprietário do Bucket. Para aplicar a configuração padrão e manter as ACLs desabilitadas, somente a permissão `s3:CreateBucket` é necessária. Para habilitar ACLs, é necessário ter a permissão `s3:PutBucketOwnershipControls`.

10. Em Configurações de bloqueio de acesso público para este bucket, selecione as configurações de bloqueio de acesso público que deseja aplicar ao bucket.

Por padrão, todas as quatro configurações do Bloqueio de Acesso Público são habilitadas. Recomendamos que você mantenha todas as configurações habilitadas, a menos que precise desativar uma ou mais delas para seu caso de uso específico. Para obter mais informações sobre como bloquear o acesso público, consulte [Bloquear o acesso público ao armazenamento do Amazon S3](#).

 Note

Para habilitar todas as configurações do Bloqueio de Acesso Público, somente a permissão `s3:CreateBucket` é necessária. Para desabilitar qualquer configuração do Bloqueio de Acesso Público, é necessário ter a permissão `s3:PutBucketPublicAccessBlock`.

11. (Opcional) Em Bucket Versioning (Versionamento do bucket), você pode escolher se deseja manter variantes de objetos no bucket. Para obter mais informações sobre versionamento, consulte [Usando o versionamento em buckets do S3](#).

Para desabilitar ou habilitar o versionamento em seu bucket, escolha **Disable (Desabilitar)** ou **Enable (Habilitar)**.


12. (Opcional) Em **Tags (Etiquetas)**, você pode optar por adicionar etiquetas ao seu bucket. As etiquetas são pares de chave-valor usados para categorizar o armazenamento.

Para adicionar uma etiqueta, escolha **Add Tag (Adicionar etiqueta)** e insira uma chave e um valor para cada etiqueta.

13. Em **Default encryption (Criptografia padrão)**, escolha **Edit (Editar)**.

14. Para configurar a criptografia padrão, em **Tipo de criptografia**, selecione uma das seguintes opções:

- Chave gerenciada pelo Amazon S3 (SSE-S3)
- Chave do AWS Key Management Service (SSE-KMS)

 **Important**

Se você usar a opção SSE-KMS em sua configuração de criptografia padrão, estará sujeito à cota de solicitações por segundo (RPS) do AWS KMS. Para obter mais informações sobre as cotas do AWS KMS e como solicitar um aumento de cota, consulte [Cotas](#) no Guia do desenvolvedor do AWS Key Management Service.

Os buckets e novos objetos são criptografados com criptografia do lado do servidor com uma chave gerenciada do Amazon S3 como nível básico da configuração de criptografia. Para obter mais informações sobre criptografia padrão, consulte [Definir o comportamento padrão da criptografia para os buckets do Amazon S3](#).

Para obter mais informações sobre como usar a criptografia no lado do servidor do Amazon S3 para criptografar seus dados, consulte [Usar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#).

15. Se você selecionou **Chave do AWS Key Management Service (SSE-KMS)**, faça o seguinte:
- a. Em **Chave do AWS KMS**, especifique sua chave do KMS de uma das seguintes maneiras:
 - Para escolher entre uma lista de chaves do KMS disponíveis, selecione **Escolher entre suas AWS KMS keys** e escolha sua chave do KMS na lista de chaves disponíveis.

As chaves Chave gerenciada pela AWS (aws/s3) e as chaves gerenciadas pelo cliente são exibidas nessa lista. Para ter mais informações sobre chaves gerenciadas pelo cliente, consulte [Chaves de clientes e chaves da AWS](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para inserir o ARN da chave do KMS, escolha Inserir ARN da AWS KMS key e insira o ARN da chave do KMS no campo exibido.
- Para criar uma chave gerenciada pelo cliente no console do AWS KMS, selecione Criar uma chave do KMS.

Para ter mais informações sobre como criar uma AWS KMS key, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

Important

Você só pode usar chaves do KMS disponíveis na mesma Região da AWS que o bucket. O console do Amazon S3 lista somente as primeiras 100 chaves do KMS na mesma região que o bucket. Para usar uma chave do KMS que não esteja listada, você deve inserir o ARN da chave do KMS. Se quiser usar uma chave do KMS que seja de propriedade de outra conta, primeiro você deverá ter permissão para usar a chave e, depois, inserir o ARN da chave do KMS. Para obter mais informações sobre permissões entre contas para chaves do KMS, consulte [Criar chaves do KMS que outras contas podem usar](#) no Guia do desenvolvedor do AWS Key Management Service. Para obter mais informações sobre a SSE-KMS, consulte [Especificando criptografia no lado do servidor com o AWS KMS \(SSE-KMS\)](#).

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 ter mais informações, consulte [Identificar chaves do KMS 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 [Usar criptografia do lado do servidor com o AWS KMS \(SSE-KMS\)](#).

- b. Quando você configura seu bucket para usar a criptografia padrão com o SSE-KMS, também pode habilitar as chaves de bucket do S3. As chaves de bucket do S3 diminuem o custo de criptografia reduzindo o tráfego de solicitações do Amazon S3 para o AWS KMS. Para ter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3](#).

Para usar Chaves de bucket do S3, em Bucket Key (Chave de bucket), escolha Enable (Habilitar).

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


- a. Selecione Advanced settings (Configurações avançadas).

 Important

A ativação do bloqueio de objetos também permite o versionamento para o bucket. Depois de habilitar, é necessário definir as configurações de retenção padrão e retenção legal do bloqueio de objetos para proteger novos objetos contra exclusão ou substituição.

- b. Se quiser habilitar o bloqueio de objetos, escolha Enable (Habilitar), leia o aviso que aparece e confirme-o.

Para ter mais informações, consulte [Usar o bloqueio de objetos do S3](#).

 Note

Para criar um bucket com bloqueio de objetos habilitado, você deve ter as seguintes permissões: `s3:CreateBucket`, `s3:PutBucketVersioning` e `s3:PutBucketObjectLockConfiguration`.

17. 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). Se você quiser restringir a criação do bucket a determinada Região da AWS, use a chave de condição [LocationConstraint](#).

Para criar um cliente a fim de acessar um endpoint de pilha dupla, é necessário especificar uma Região da AWS. Para ter mais informações, consulte [Endpoints de pilha dupla](#). 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](#).

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](#).

```
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,
                // 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](#).

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](#).

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

# Example usage:
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
```

```
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.

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 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](#).
 - 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](#).

- 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](#).
- 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](#)
- 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](#).
- 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](#).
- 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](#).
- 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](#).
- 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 ter mais informações, consulte [Hospedagem de um site estático usando o Amazon S3](#).

Usar a AWS CLI

Você também pode usar a AWS Command Line Interface (AWS CLI) para visualizar as propriedades de um bucket do S3. Para obter mais informações, consulte os comandos a seguir na Referência de comandos da AWS CLI.

- [get-bucket-tagging](#)
- [get-bucket-versioning](#)
- [get-bucket-encryption](#)
- [get-bucket-notification-configuration](#)
- [get-bucket-logging](#)

Para obter informações sobre o AWS CLI, consulte [O que é AWS Command Line Interface?](#) no Guia do usuário do AWS Command Line Interface.

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. Os objetos adicionados ao bucket enquanto a ação de esvaziar estiver em andamento também poderão ser excluídos. Todos os objetos (incluindo todas as versões do objeto e marcadores de exclusão) no bucket devem ser excluídos antes que o próprio bucket possa ser excluído.

Ao esvaziar um bucket com o versionamento do S3 habilitado ou suspenso, todas as versões de todos os objetos no bucket são excluídas. Para ter mais informações, consulte [Trabalhar com objetos em um bucket com versionamento habilitado](#).

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 ter mais informações, consulte [Definir uma configuração de ciclo de vida em um bucket](#). Para esvaziar um bucket grande, recomendamos que você use uma regra de configuração do ciclo de vida do S3. A expiração do ciclo de vida é um processo assíncrono, portanto, a regra pode levar alguns dias para ser executada antes que o bucket fique vazio. Depois da primeira vez que o Amazon S3 executa a regra, todos os objetos que estão qualificados para expiração são marcados para exclusão. Você não recebe mais cobranças pelos objetos marcados para exclusão. Para obter mais informações, consulte [Como esvaziar um bucket do Amazon S3 usando uma regra de configuração do ciclo de vida?](#).

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](#).

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](#). 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

Para esvaziar um bucket grande, recomendamos que você use uma regra de configuração do ciclo de vida do S3. A expiração do ciclo de vida é um processo assíncrono, portanto, a regra pode levar alguns dias para ser executada antes que o bucket fique vazio. Depois da primeira vez que o Amazon S3 executa a regra, todos os objetos que estão qualificados para expiração são marcados para exclusão. Você não recebe mais cobranças pelos objetos marcados para exclusão. Para obter mais informações, consulte [Como esvaziar um bucket do Amazon S3 usando uma regra de configuração do ciclo de vida?](#).

Se você usar uma configuração de ciclo de vida para esvaziar seu bucket, a configuração deverá 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 interrompe 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 [Configurar uma política de ciclo de vida de bucket para anular uploads incompletos de várias partes](#).

Para obter mais informações sobre como usar uma configuração de ciclo de vida para esvaziar um bucket, consulte [Definir uma configuração de ciclo de vida em um bucket](#) e [Expirando objetos](#).

Esvaziar um bucket com o AWS CloudTrail configurado

O AWS CloudTrail rastreia eventos de dados em nível de objeto em um bucket do Amazon S3, como a exclusão de objetos. Se você usa um bucket como destino para registrar os eventos do CloudTrail e está excluindo objetos desse mesmo bucket, você pode estar criando objetos enquanto esvazia o bucket. Para evitar isso, pare as trilhas do AWS CloudTrail. Para obter mais informações sobre como impedir que as trilhas do CloudTrail registrem eventos, consulte [Desativar o registro de uma trilha](#) no Guia 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 política de bucket. Se você quiser armazenar novos objetos no bucket posteriormente, precisará remover essa declaração `s3:PutObject` de negação. Para obter mais informações, consulte [Operações com objetos](#) e [Elementos da política de JSON do IAM: efeito](#) no Guia do usuário do IAM.

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 [Tutorial: Configurar um site estático usando um domínio personalizado registrado no Route 53](#), 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](#).
- 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.
- Verifique se não existe nenhum ponto de acesso associado: só é possível excluir buckets que não tenham nenhum ponto de acesso associado. Exclua todos os pontos de acesso associados ao bucket antes de excluí-lo.
- Políticas de controle de serviços (SCPs) do AWS Organizations: uma política de controle de serviços pode negar a permissão de exclusão em um bucket. Para receber informações sobre SCPs, consulte [Políticas de controle de serviço](#) no Guia do usuário do AWS Organizations.
- Permissões s3:DeleteBucket: se você não puder excluir um bucket, trabalhe com o administrador do IAM para confirmar que você tem permissões s3:DeleteBucket. 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](#).

```
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` (remove 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

Important

O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).

Todos os buckets do Amazon S3 têm a criptografia configurada por padrão, e os objetos são automaticamente criptografados usando criptografia do lado do servidor, com chaves gerenciadas do Amazon S3 (SSE-S3). Essa configuração de criptografia se aplica a todos os objetos em seus buckets do Amazon S3.

Se você precisar de maior controle sobre suas chaves, como gerenciar a alternância de chaves e as concessões de políticas de acesso, poderá optar por usar a criptografia do lado do servidor chaves do AWS Key Management Service (AWS KMS) (SSE-KMS) ou a criptografia de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS). Para ter mais informações sobre edição de chaves do KMS, consulte [Editar chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

Note

Alteramos os buckets para criptografar automaticamente uploads de novos objetos. Se você criou anteriormente um bucket sem criptografia padrão, o Amazon S3 ativará a criptografia por padrão para o bucket usando SSE-S3. Não haverá nenhuma alteração na configuração de criptografia padrão para um bucket existente que já tenha a criptografia SSE-S3 ou SSE-KMS configurada. Se quiser criptografar seus objetos com a SSE-KMS, deverá alterar o

tipo de criptografia nas configurações do bucket. Para ter mais informações, consulte [Usar criptografia do lado do servidor com o AWS KMS \(SSE-KMS\)](#).

Ao configurar seu bucket para usar criptografia padrão com a SSE-KMS, você também pode ativar o recurso de chaves de bucket do S3 para diminuir o tráfego de solicitações do Amazon S3 para o AWS KMS e reduzir o custo de criptografia. Para ter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3](#).

Para identificar buckets que têm SSE-KMS habilitada como criptografia padrão, você pode usar as métricas da Lente de Armazenamento do Amazon S3. A Lente de Armazenamento do S3 é um recurso de análise de armazenamento em nuvem que você pode usar para obter visibilidade em toda a organização sobre o uso e a atividade do armazenamento de objetos. Para obter mais informações, consulte [Como usar a Lente de Armazenamento do S3 para proteger seus dados](#).

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 o baixa. Para ter mais informações sobre como proteger dados usando a criptografia no lado do servidor e o gerenciamento de chaves de criptografia, consulte [Proteger os dados usando criptografia do lado do servidor](#).

Para ter mais informações sobre permissões necessárias para criptografia padrão, consulte [PutBucketEncryption](#) na Referência da API do Amazon Simple Storage Service.

Você pode configurar o comportamento da criptografia padrão do Amazon S3 para um bucket do S3 usando o console do Amazon S3, os AWS SDKs, a API REST do Amazon S3 e a AWS Command Line Interface (AWS CLI).

Criptografar objetos existentes

Para criptografar objetos existentes não criptografados do Amazon S3, você pode usar Operações em Lote do Amazon S3. 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](#) 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 operação de API CopyObject ou o comando copy-object da AWS CLI. 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 definida como SSE-KMS não podem ser usados como buckets de destino para [the section called “Registrando acesso ao servidor”](#). Somente a criptografia padrão SSE-S3 é suportada para buckets de destino do log de acesso do servidor.

Usar criptografia SSE-KMS para operações entre contas

Ao utilizar a criptografia para operações entre contas, esteja ciente do seguinte:

- Se o nome do recurso da Amazon (ARN) da AWS KMS key ou alias não for fornecido no momento da solicitação nem por meio da configuração de criptografia padrão do bucket, a Chave gerenciada pela AWS (aws/s3) será utilizada.
- Se você estiver fazendo upload ou acessando objetos do S3 usando entidades principais do AWS Identity and Access Management (IAM) que estão na mesma Conta da AWS que a sua chave do KMS, poderá usar a Chave gerenciada pela AWS (aws/s3).
- Se quiser conceder acesso entre contas aos seus objetos do S3, use uma chave gerenciada pelo cliente. Você pode configurar a política de uma chave gerenciada pelo cliente para permitir o acesso de outra conta.
- Se estiver especificando sua própria chave do KMS, recomendamos usar um ARN de chave do KMS totalmente qualificado. Se você usar um alias da chave do KMS, o AWS KMS resolverá a chave na conta do solicitante. Isso pode resultar em dados criptografados com uma chave do KMS pertencente ao solicitante, e não ao proprietário do bucket.
- É necessário especificar uma chave para a qual você (o solicitante) recebeu a permissão Encrypt. Para ter 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 ter 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 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. Como resultado, as tags de entidade (ETags) dos objetos de origem diferem dos ETags dos objetos de réplica. Se você tiver aplicações que usam ETags, será necessário atualizar essas aplicações para considerar essa diferença.
- Se os objetos no bucket de origem forem criptografados usando a criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3), a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS) ou a criptografia de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS), os objetos de réplica no bucket de destino usarão o mesmo tipo de criptografia que os objetos 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 \(SSE-C, SSE-S3, SSE-KMS, DSSE-KMS\)](#).

Usar Chaves de bucket do Amazon S3 com criptografia padrão

Quando você configura seu bucket para usar a SSE-KMS como o comportamento de criptografia padrão para novos objetos, você também pode configurar chaves 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 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 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.

Para obter mais informações sobre como usar Chaves de bucket do S3, consulte [Uso de chaves de bucket do Amazon S3](#).

Configurar a criptografia padrão

Important


O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).

Os buckets do Amazon S3 têm a criptografia configurada por padrão, e os novos objetos são automaticamente criptografados usando criptografia do lado do servidor, com chaves gerenciadas do Amazon S3 (SSE-S3). Essa criptografia se aplica a todos os novos objetos em seus buckets do Amazon S3 e não tem nenhum custo para você.

Se você precisar de maior controle sobre suas chaves de criptografia, como gerenciar a alternância de chaves e as concessões de políticas de acesso, poderá optar por usar a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS) ou a criptografia de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS). Para obter mais informações sobre SSE-KMS, consulte [Especificando criptografia no lado do servidor com o AWS KMS \(SSE-KMS\)](#). Para ter mais informações sobre DSSE-KMS, consulte [the section called “Criptografia de camada dupla do lado do servidor \(DSSE-KMS\)”](#).

Se quiser usar uma chave do KMS que seja de propriedade de outra conta, você deverá ter permissão para usar a chave. Para obter mais informações sobre permissões entre contas para chaves do KMS, consulte [Criar chaves do KMS que outras contas podem usar](#) no Guia do desenvolvedor do AWS Key Management Service.

Ao definir a criptografia do bucket padrão como SSE-KMS, você também pode configurar uma chave de bucket do S3 para reduzir os custos de solicitação do AWS KMS. Para ter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3](#).

 Note

Se você usa [PutBucketEncryption](#) para definir a criptografia de bucket padrão como SSE-KMS, é necessário verificar se o ID da chave do KMS está correto. O Amazon S3 não valida o ID da chave do KMS fornecido nas solicitações PutBucketEncryption.

Não há cobranças adicionais para usar a criptografia padrão para buckets do S3. As solicitações para configurar o comportamento de criptografia padrão geram cobranças padrão de solicitação do Amazon S3. Para obter mais informações sobre preços, consulte [Preços do Amazon S3](#). Para a SSE-KMS e DSSE-KMS, tarifas do AWS KMS podem ser aplicadas. Elas estão listadas em [Preços do AWS KMS](#).

A criptografia no lado do servidor com chaves fornecidas pelo cliente (SSE-C) não é compatível para criptografia padrão.

Você pode configurar a criptografia padrão do Amazon S3 para um bucket do S3 usando o console do Amazon S3, os AWS SDKs, a API REST do Amazon S3 e a AWS Command Line Interface (AWS CLI).

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 ou a DSSE-KMS em sua configuração de criptografia padrão, poderão ser cobradas cotas de solicitações por segundo (RPS) do AWS KMS. Para ter mais informações sobre as cotas do AWS KMS e como solicitar um aumento de cota, consulte [Cotas](#) no Guia do desenvolvedor do AWS Key Management Service.

Note

Os objetos carregados antes da ativação da criptografia padrão não serão criptografados. Para ter informações sobre criptografia de objetos existentes, consulte [the section called “Definir criptografia de bucket padrão”](#).

Uso do console do S3

Como configurar 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. No painel de navegação à esquerda, escolha Buckets.
3. Na lista de Buckets, escolha o nome do bucket desejado.
4. Escolha a guia Properties (Propriedades).
5. Em Default encryption (Criptografia padrão), escolha Edit (Editar).
6. Para configurar a criptografia padrão, em Tipo de criptografia, selecione uma das seguintes opções:
 - Criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3)
 - Criptografia no lado do servidor com chaves do AWS Key Management Service (SSE-KMS)
 - Criptografia de camada dupla do lado do servidor com chaves do AWS Key Management Service (DSSE-KMS)

Important

Se você usar a opção SSE-KMS ou a DSSE-KMS em sua configuração de criptografia padrão, poderão ser cobradas cotas de solicitações por segundo (RPS) do AWS KMS. Para obter mais informações sobre as cotas do AWS KMS e como solicitar

um aumento de cota, consulte [Cotas](#) no Guia do desenvolvedor do AWS Key Management Service.

Os buckets e novos objetos são criptografados por padrão com o SSE-S3, a menos que você especifique outro tipo de criptografia padrão para eles. Para obter mais informações sobre criptografia padrão, consulte [Definir o comportamento padrão da criptografia para os buckets do Amazon S3](#).

Para obter mais informações sobre como usar a criptografia no lado do servidor do Amazon S3 para criptografar seus dados, consulte [Usar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#).

7. Se você escolheu Criptografia do lado do servidor com chaves do AWS Key Management Service (SSE-KMS) ou Criptografia de camada dupla do lado do servidor com chaves do AWS Key Management Service (DSSE-KMS), faça o seguinte:

a. Em Chave do AWS KMS, especifique sua chave do KMS de uma das seguintes maneiras:

- Para escolher entre uma lista de chaves do KMS disponíveis, selecione Escolher entre suas AWS KMS keys e escolha sua chave do KMS na lista de chaves disponíveis.

As chaves Chave gerenciada pela AWS (aws/s3) e as chaves gerenciadas pelo cliente são exibidas nessa lista. Para ter mais informações sobre chaves gerenciadas pelo cliente, consulte [Chaves de clientes e chaves da AWS](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para inserir o ARN da chave do KMS, escolha Inserir ARN da AWS KMS key e insira o ARN da chave do KMS no campo exibido.
- Para criar uma chave gerenciada pelo cliente no console do AWS KMS, selecione Criar uma chave do KMS.

Para ter mais informações sobre como criar uma AWS KMS key, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

 **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 Inserir o ARN da AWS KMS key 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. Para obter mais informações sobre essas chaves, consulte [Chaves do KMS de criptografia simétrica](#) no Guia do desenvolvedor do AWS Key Management Service.

Para ter mais informações sobre como usar a SSE-KMS com o Amazon S3, consulte [Usar criptografia do lado do servidor com o AWS KMS \(SSE-KMS\)](#). Para ter mais informações sobre como usar a DSSE-KMS, consulte [the section called “Criptografia de camada dupla do lado do servidor \(DSSE-KMS\)”](#).

- b. Ao configurar seu bucket para usar a criptografia padrão com a SSE-KMS, você também pode ativar a chave de bucket do S3. As chaves de bucket do S3 diminuem o custo de criptografia reduzindo o tráfego de solicitações do Amazon S3 para o AWS KMS. Para ter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3](#).

Para usar Chaves de bucket do S3, em Bucket Key (Chave de bucket), escolha Enable (Habilitar).

 Note

As chaves de bucket do S3 não comportam DSSE-KMS.

8. Escolha Salvar alterações.

Usando a AWS CLI

Esses exemplos mostram como configurar a criptografia padrão usando SSE-S3 ou usando a 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](#). 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 chaves gerenciadas pelo Amazon S3.

```
aws s3api put-bucket-encryption --bucket DOC-EXAMPLE-BUCKET --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 DOC-EXAMPLE-BUCKET --server-side-encryption-configuration '{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "aws:kms",
        "KMSMasterKeyID": "KMS-Key-ARN"
      },
      "BucketKeyEnabled": true
    }
  ]
}'
```

Uso da API REST

Use a operação `PutBucketEncryption` da API REST para habilitar a criptografia padrão e definir o tipo de criptografia do lado do servidor a ser utilizado, ou seja, SSE-S3, SSE-KMS ou DSSE-KMS.

Para obter mais informações, consulte [PutBucketEncryption](#) na Referência da API do Amazon Simple Storage Service.

Monitorar a criptografia padrão com o AWS CloudTrail e o Amazon EventBridge

Important

O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).

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 regras do EventBridge que correspondam aos eventos do CloudTrail para essas chamadas de API. Para obter mais informações sobre eventos do CloudTrail, consulte [Habilitar o registro em log de objetos em um bucket usando o console](#). Para ter mais informações sobre eventos do EventBridge, consulte [Eventos de Serviços da AWS](#).

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 um dos seguintes campos como o par de nome-valor: "SSEApplied": "Default_SSE_S3", "SSEApplied": "Default_SSE_KMS" ou "SSEApplied": "Default_DSSE_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", "SSEApplied": "DSSE_KMS" ou "SSEApplied": "SSE_C".

Para multipart uploads, essas informações estão incluídas nas solicitações de operação de API `InitiateMultipartUpload`. Para obter mais informações sobre como usar o CloudTrail e o CloudWatch, consulte [Monitorar o Amazon S3](#).

Trabalhar com o Mountpoint para Amazon S3

O Mountpoint para Amazon S3 é um cliente de arquivos de código aberto com alta taxa de throughput para montar um bucket do Amazon S3 como um sistema de arquivos local. Com o Mountpoint, as aplicações podem acessar objetos armazenados no Amazon S3 por meio de operações do sistema de arquivo, como abertura e leitura. O Mountpoint traduz automaticamente essas operações em chamadas de API de objetos do S3, dando aos aplicativos acesso ao armazenamento elástico e ao throughput do Amazon S3 por meio de uma interface de arquivo.

O Mountpoint para Amazon S3 está [disponível ao público em geral](#) para uso na produção, em aplicações em grande escala com uso intenso de leitura: data lakes, treinamento de machine learning, renderização de imagens, simulação de veículos autônomos, extração, transformação e carregamento (ETL) e muito mais.

O Mountpoint permite operações básicas de sistema de arquivos e pode ler arquivos de até 5 TB. Ele pode listar e ler arquivos existentes e criar outros. Ele não pode modificar arquivos existentes nem excluir diretórios e não é compatível com links simbólicos ou bloqueio de arquivos. O Mountpoint é ideal para aplicações que não precisam de todos os recursos de um sistema de arquivos compartilhado e de permissões no estilo POSIX, mas exigem o throughput elástico do Amazon S3 para ler e gravar grandes conjuntos de dados do S3. Para obter detalhes, consulte [Comportamento do sistema de arquivos do Mountpoint](#) no GitHub. Quanto a workloads que exigem compatibilidade total com o POSIX, recomendamos o [Amazon FSx para Lustre](#), que [permite vincular buckets do S3](#).

O Mountpoint para Amazon S3 está disponível somente para sistemas operacionais Linux. Você pode usar o Mountpoint para acessar objetos do S3 em todas as classes de armazenamento, exceto S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, S3 Intelligent-Tiering Archive Access Tier e S3 Intelligent-Tiering Deep Archive Access Tier.

Tópicos

- [Instalar o Mountpoint](#)
- [Configurar e usar o Mountpoint](#)

Instalar o Mountpoint

Você pode baixar e instalar pacotes pré-compilados do Mountpoint para Amazon S3 usando a linha de comando. As instruções para baixar e instalar o Mountpoint variam, dependendo de qual sistema operacional Linux você está usando.

Tópicos

- [Distribuições baseadas em RPM \(Amazon Linux, Fedora, CentOS, RHEL\)](#)
- [Distribuições baseadas em DEB \(Debian, Ubuntu\)](#)
- [Outras distribuições do Linux](#)
- [Conferir a assinatura do pacote do Mountpoint para Amazon S3](#)

Distribuições baseadas em RPM (Amazon Linux, Fedora, CentOS, RHEL)

1. Copie o URL de download a seguir referente à sua arquitetura.

x86_64:

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

ARM64 (Graviton):

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

2. Baixe o pacote do Mountpoint para Amazon S3. Substitua *download-link* pelo URL de download apropriado da etapa anterior.

```
wget download-link
```

3. (Opcional) Verifique a integridade e a autenticidade do arquivo baixado. Primeiro, copie o URL de assinatura apropriado para a sua arquitetura.

x86_64:

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

ARM64 (Graviton):

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

Depois, consulte [Verificar a assinatura do pacote do Mountpoint para Amazon S3](#).

4. Instale o pacote usando o seguinte comando:

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

5. Confira se o Mountpoint foi instalado com êxito digitando o seguinte comando:

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

Você deve ver saída semelhante a:

```
mount-s3 1.3.1
```

Distribuições baseadas em DEB (Debian, Ubuntu)

1. Copie o URL de download para a arquitetura.

x86_64:

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

ARM64 (Graviton):

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

2. Baixe o pacote do Mountpoint para Amazon S3. Substitua *download-link* pelo URL de download apropriado da etapa anterior.

```
wget download-link
```


3. (Opcional) Verifique a integridade e a autenticidade do arquivo baixado. Primeiro, copie o URL de assinatura para a arquitetura.

x86_64:

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

ARM64 (Graviton):

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

Depois, consulte [Verificar a assinatura do pacote do Mountpoint para Amazon S3](#).

4. Instale o pacote usando o seguinte comando:

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

5. Confira se o Mountpoint para Amazon S3 foi instalado com êxito digitando o seguinte comando:

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

Você deve ver saída semelhante a:

```
mount-s3 1.3.1
```

Outras distribuições do Linux

1. Consulte a documentação do sistema operacional para instalar os pacotes FUSE e libfuse2, que são obrigatórios.
2. Copie o URL de download para a arquitetura.

x86_64:

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

ARM64 (Graviton):

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

3. Baixe o pacote do Mountpoint para Amazon S3. Substitua *download-link* pelo URL de download apropriado da etapa anterior.

```
wget download-link
```

4. (Opcional) Verifique a integridade e a autenticidade do arquivo baixado. Primeiro, copie o URL de assinatura para a arquitetura.

x86_64:

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

ARM64 (Graviton):

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

Depois, consulte [Verificar a assinatura do pacote do Mountpoint para Amazon S3](#).

5. Instale o pacote usando o seguinte comando:

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

6. Adicione o `mount-s3` binário à variável de ambiente `PATH`. No arquivo `$HOME/.profile`, anexe a seguinte linha:

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

Salve o arquivo `.profile` e execute o seguinte comando:

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

7. Confira se o Mountpoint para Amazon S3 foi instalado com êxito digitando o seguinte comando:

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

Você deve ver saída semelhante a:

```
mount-s3 1.3.1
```

Conferir a assinatura do pacote do Mountpoint para Amazon S3

1. Instale GnuPG (o comando `gpg`). É necessário verificar a autenticidade e a integridade de um pacote do Mountpoint para Amazon S3 baixado. GnuPG é instalado por padrão em imagens de máquina da Amazon (AMIs) do Linux Amazon. Depois de instalar GnuPG, vá para a etapa 2.
2. Baixe a chave pública do Mountpoint executando o seguinte comando:

```
wget https://s3.amazonaws.com/mountpoint-s3-release/public_keys/KEYS
```

3. Importe a chave pública do Mountpoint para o token de autenticação executando o seguinte comando:

```
gpg --import KEYS
```

4. Verifique a impressão digital da chave pública do Mountpoint executando o seguinte comando:

```
gpg --fingerprint mountpoint-s3@amazon.com
```

Confirme se a string de impressões digitais exibida corresponde ao seguinte:

```
673F E406 1506 BB46 9A0E F857 BE39 7A52 B086 DA5A
```

Se a string de impressões digitais não coincidir, não conclua a instalação do Mountpoint. Nesse caso, entre em contato com o [AWS Support](#).

5. Faça download do arquivo SIGNATURE do pacote. Substitua *signature-link* pelo link de assinatura apropriado das seções anteriores.

```
wget signature-link
```

6. Verifique a assinatura do pacote baixado executando o comando a seguir. Substitua *signature-filename* pelo nome do arquivo da etapa anterior.

```
gpg --verify signature-filename
```

Por exemplo, em distribuições com base em RPM, incluindo o Amazon Linux, execute o seguinte comando:

```
gpg --verify mount-s3.rpm.asc
```

7. A saída deve incluir a frase `Good signature`. Se a saída incluir a frase `BAD signature`, baixe novamente o arquivo do pacote do Mountpoint e repita essas etapas. Se o problema persistir, não conclua a instalação do Mountpoint. Nesse caso, entre em contato com o [AWS Support](#).

A saída pode incluir um aviso sobre uma assinatura confiável. Isso não indica um problema. Significa apenas que você não verificou de forma independente a chave pública do Mountpoint.

Configurar e usar o Mountpoint

Para usar o Mountpoint para Amazon S3, o host precisa de credenciais válidas da AWS, com acesso ao bucket ou aos buckets que você gostaria de montar. Para diferentes formas de autenticação, consulte [Credenciais da AWS](#) para o Mountpoint no GitHub.

Por exemplo, você pode criar um usuário e um perfil do AWS Identity and Access Management (IAM) para essa finalidade. Esse perfil deve ter acesso ao bucket ou aos buckets que você gostaria de montar. É possível [transmitir o perfil do IAM](#) à instância do Amazon EC2 com um perfil de instância.

Usar o Mountpoint para Amazon S3

Use o Mountpoint para Amazon S3 para fazer o seguinte:

1. Montar buckets com o comando `mount-s3`.

No exemplo a seguir, substitua `DOC-EXAMPLE-BUCKET` pelo nome do bucket do S3 e substitua `~/mnt` pelo diretório no host em que você deseja montar o bucket do S3.

```
mkdir ~/mnt
mount-s3 DOC-EXAMPLE-BUCKET ~/mnt
```

Como o cliente do Mountpoint é executado em segundo plano por padrão, o diretório `~/mnt` agora fornece acesso aos objetos no bucket do S3.

2. Acesse os objetos no bucket por meio do Mountpoint.

Depois de montar o bucket localmente, você pode usar os comandos Linux comuns, como `cat` ou `ls`, para trabalhar com os objetos do S3. O Mountpoint para Amazon S3 interpreta

as chaves no bucket do S3 como caminhos do sistema de arquivos dividindo-as com o caractere de barra (/). Por exemplo, se você tiver a chave do objeto `Data/2023-01-01.csv` no bucket, terá um diretório chamado `Data` no sistema de arquivos do Mountpoint, o qual contém um arquivo chamado `2023-01-01.csv`.

O Mountpoint para Amazon S3 não implementa intencionalmente a especificação padrão completa [POSIX](#) para sistemas de arquivos. O Mountpoint é otimizado para workloads que, embora precisem de alta taxa de throughput para acesso de leitura e gravação aos dados armazenados no Amazon S3 por meio de uma interface de sistema de arquivos, de outra forma não dependem dos atributos do sistema de arquivos. Para receber mais informações, consulte [Comportamento do sistema de arquivos](#) do Mountpoint para Amazon S3 no GitHub. Os clientes que precisam de uma semântica mais completa do sistema de arquivos devem considerar outros serviços de arquivos AWS, como [Amazon Elastic File System \(Amazon EFS\)](#) ou [Amazon FSx](#).

3. Desmonte o bucket usando o comando `umount`. Esse comando desmonta o bucket do S3 e sai do Mountpoint.

Para usar o comando de exemplo a seguir, substitua `~/mnt` pelo diretório no host em que o bucket do S3 está montado.

```
umount ~/mnt
```

Note

Para receber uma lista de opções para esse comando, execute `umount --help`.

Para receber detalhes adicionais sobre a configuração do Mountpoint, consulte [Configuração do bucket do S3](#) e [Configuração do sistema de arquivos](#) no GitHub.

Configurar o armazenamento em cache no Mountpoint

Ao usar o Mountpoint para Amazon S3, você pode configurá-lo para armazenar em cache os dados acessados mais recentemente dos buckets do S3 no armazenamento de instâncias do Amazon EC2 ou em um volume anexado do Amazon EBS. Armazenar esses dados em cache pode ajudar a acelerar a performance e reduzir o custo do acesso repetido aos dados. O armazenamento em cache no Mountpoint é ideal para casos de uso em que você lê repetidamente os mesmos dados

que não mudam durante as várias leituras. Por exemplo, você pode usar o armazenamento em cache com trabalhos de treinamento de machine learning que precisam ler um conjunto de dados de treinamento várias vezes para melhorar a precisão do modelo.

Ao montar um bucket do S3, você também pode habilitar o armazenamento em cache por meio de sinalizadores. É possível configurar a localização e o tamanho do cache de dados e a quantidade de tempo em que os metadados são retidos no cache. Quando você monta um bucket e o armazenamento em cache é habilitado, o Mountpoint cria um subdiretório vazio no local do cache configurado, caso esse subdiretório ainda não exista. Quando você monta um bucket pela primeira vez e quando desmonta, o Mountpoint exclui o conteúdo da localização do cache. Para obter mais informações sobre como configurar e usar o armazenamento em cache no Mountpoint, consulte [Mountpoint for Amazon S3 Caching configuration](#) no GitHub.

Ao montar um bucket do S3, você pode habilitar o armazenamento em cache com o sinalizador `--cache` `CACHE_PATH`. No exemplo a seguir, substitua `CACHE_PATH` pelo caminho de arquivo do diretório no qual você deseja armazenar os dados em cache. Substitua `DOC-EXAMPLE-BUCKET` pelo nome do bucket do S3 e substitua `~/mnt` pelo diretório no host em que você deseja montar o bucket do S3.

```
mkdir ~/mnt
mount-s3 --cache CACHE_PATH DOC-EXAMPLE-BUCKET ~/mnt
```

Important

Se você habilitar o armazenamento em cache, o Mountpoint manterá o conteúdo de objeto não criptografado do bucket do S3 no local de armazenamento em cache configurado na montagem. Para proteger os dados, recomendamos que você restrinja o acesso ao local do cache de dados.

Solução de problemas do Mountpoint

O Mountpoint para Amazon S3 conta com o suporte do AWS Support. Se precisar de ajuda, entre em contato com o [AWS Support Center](#).

Você também pode examinar e enviar [Problemas](#) do Mountpoint no GitHub.

Se você descobrir um possível problema de segurança nesse projeto, pedimos que você notifique a Segurança da AWS por meio de nossa página [Relatório de vulnerabilidade](#). Não crie um problema público no GitHub.

Se a aplicação se comportar inesperadamente com o Mountpoint, você poderá inspecionar as informações de log para diagnosticar o problema.

Registro em log

Por padrão, o Mountpoint emite informações de log de alta gravidade para [syslog](#).

Para visualizar os logs sobre as distribuições do Linux mais modernas, inclusive o Amazon Linux, execute o seguinte comando `journalctl`:

```
journalctl -e SYSLOG_IDENTIFIER=mount-s3
```

Em outros sistemas Linux, as entradas `syslog` provavelmente são gravadas em um arquivo como `/var/log/syslog`.

Você pode usar esses logs para solucionar problemas da aplicação. Por exemplo, se a aplicação tentar sobrescrever um arquivo existente, a operação falhará, e você verá uma linha semelhante à seguinte no log:

```
[WARN] open{req=12 ino=2}: mountpoint_s3::fuse: open failed: inode error: inode 2 (full key "README.md") is not writable
```

Para receber mais informações, consulte [Registro em log](#) do Mountpoint para Amazon S3 no GitHub.

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](#).
- 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](#).

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 é compatível com buckets localizados nas seguintes regiões:

- Ásia Pacific (Tóquio) (ap-northeast-1)
- Ásia-Pacífico (Seul) (ap-northeast-2)

- Ásia-Pacífico (Mumbai) (ap-south-1)
- Ásia-Pacífico (Singapura) (ap-southeast-1)
- Ásia-Pacífico (Sydney) (ap-southeast-2)
- Canadá (Central) (ca-central-1)
- Europa (Frankfurt) (eu-central-1)
- Europa (Irlanda) (eu-west-1)
- Europa (Londres) (eu-west-2)
- Europa (Paris) (eu-west-3)
- América do Sul (São Paulo) (sa-east-1)
- 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)

- 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 pode continuar usando os endpoints regulares para a transferência de dados padrão.
- 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`.

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](#)
- [Habilitar e usar o S3 Transfer Acceleration](#)

- [Usar a ferramenta de comparação de velocidade do Amazon S3 Transfer Acceleration](#)

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](#).

Para obter mais informações, consulte [Habilitar e usar o S3 Transfer Acceleration](#).

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](#) e [Usar endpoints de pilha dupla do Amazon S3](#).

 Note

A aplicação de transferência de dados deve usar um dos dois tipos de endpoints a seguir para acessar o bucket e agilizar a transferência de dados: `.s3-accelerate.amazonaws.com` ou `.s3-accelerate.dualstack.amazonaws.com` para o endpoint de pilha dupla. Se quiser usar a transferência de dados padrão, você poderá continuar usando os endpoints regulares.

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](#).

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 a Aceleração de Transferências usando o console do S3, a AWS Command Line Interface (AWS CLI), a API 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](#).

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.

Usando 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](#).

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 dos AWS SDKs

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](#). Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3](#).

```
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](#).

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](#).

Uso da API REST

Use a operação `PutBucketAccelerateConfiguration` da API REST para acelerar a configuração em um bucket existente.

Para obter mais informações, consulte [PutBucketAccelerateConfiguration](#) na Referência da API do Amazon Simple Storage Service.

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 qualquer 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-comparison.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 mostrar que entendem que serão cobrados pela solicitação e pelo download dos dados. Para mostrar que aceitam as cobranças, os solicitantes devem incluir `x-amz-request-payer` como um cabeçalho na solicitação de API para as solicitações DELETE, GET, HEAD, POST e PUT, ou adicionar o parâmetro `RequestPayer` na solicitação REST. Para solicitações da CLI, os solicitantes podem usar o parâmetro `--request-payer`.

Example : usar a opção Pagamento pelo solicitante ao excluir um objeto

Para usar o exemplo de API [DeleteObjectVersion](#) a seguir, substitua *user input placeholders* por suas próprias informações.

```
DELETE /Key+?versionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-mfa: MFA
x-amz-request-payer: RequestPayer
x-amz-bypass-governance-retention: BypassGovernanceRetention
x-amz-expected-bucket-owner: ExpectedBucketOwner
```

Se o solicitante restaurar objetos usando a API [RestoreObject](#), a opção Pagamento pelo solicitante será compatível desde que o cabeçalho `x-amz-request-payer` ou o parâmetro `RequestPayer` estejam na solicitação. No entanto, o solicitante paga apenas pelo custo da solicitação. O proprietário do bucket paga pelas cobranças de recuperação.

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](#)
- [Recuperar a configuração requestPayment usando a API REST](#)
- [Fazer download de objetos de buckets de Pagamento pelo solicitante](#)

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 de 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](#).

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, será possível aumentar a cota de bucket da conta para no máximo 1.000 enviando uma solicitação de aumento de cota. Não há diferença no desempenho ao usar muitos buckets ou somente alguns.

Note

Você não precisa enviar várias solicitações de aumento de cota para cada Região da AWS. A cota de bucket é aplicada à Conta da AWS.

Para receber informações sobre como aumentar a cota do bucket, acesse [AWS Service Quotas](#) 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](#).

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](#).

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.

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Important

No console do Amazon S3, quando você escolhe Open (Abrir) ou Download As (Baixar como) para um objeto, essas operações criam URLs pré-assinados. Durante cinco minutos, seu objeto estará acessível a qualquer pessoa que tenha acesso a esses URLs pré-assinados. Para obter mais informações sobre URLs pré-assinados, consulte [Usar URLs pré-assinados](#).

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](#)
- [Criar nomes de chave de objeto](#)
- [Trabalhar com metadados de objeto](#)

- [Fazer upload de objetos](#)
- [Carregar e copiar objetos usando multipart upload](#)
- [Copiar e mover objetos](#)
- [Baixar objetos](#)
- [Verificar a integridade do objeto](#)
- [Excluir objetos do Amazon S3](#)
- [Organizar, listar e trabalhar com seus objetos](#)
- [Trabalhar com URLs pré-assinados](#)
- [Transformar objetos com o S3 Object Lambda](#)

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](#).

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](#).

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](#).

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](#).

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](#).

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](#)
- [Gerenciamento de identidade e acesso para o Amazon S3](#)
- [Configurar ACLs](#)

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 com URLs pré-assinados](#).

Tags

Para categorizar objetos armazenados para a finalidade de controle de acesso ou alocação de custos, você pode usar tags. Para ter mais informações, consulte [Categorizando seu armazenamento usando tags](#).

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) .

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](#).

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. Os nomes das chaves de objeto diferenciam letras maiúsculas de minúsculas.

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](#).

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

Recomendamos que você não use os caracteres a seguir no nome de uma chave, devido ao tratamento de caracteres especiais significativos, que não é consistente em todas as aplicações.

- 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\r\n (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](#).

Há dois tipos de metadados no Amazon S3: definidos pelo sistema e definidos pelo usuário. As seções a seguir 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](#).

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:

- Controlados pelo sistema: os metadados, como a data de criação do objeto, são controlados pelo sistema e somente o Amazon S3 pode modificar o valor.
- Controlados pelo usuário: 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](#).

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 [Proteger dados com criptografia](#).

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?
Date	A 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	O tamanho do objeto em bytes.	Não
Content-Type	O tipo do objeto.	Sim
Last-Modified	A data de criação do objeto ou data da última modificação, o que aconteceu por último. Para carregamentos	Não

Nome	Descrição	O usuário pode modificar o valor?
	fracionados, a data de criação do objeto é a data de início do respectivo carregamento.	
ETag	Uma tag de entidade (ETag) que 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	Um cabeçalho que indica se a criptografia do lado do servidor está ativada para o objeto e se essa criptografia está usando as chaves do AWS Key Management Service (AWS KMS) (SSE-KMS) ou usando as chaves de criptografia gerenciadas pelo Amazon S3 (SSE-S3). Para ter mais informações, consulte Proteger os dados usando criptografia do lado do servidor .	Sim
x-amz-checksum-crc32 , x-amz-checksum-crc32c , x-amz-checksum-sha1 , x-amz-checksum-sha256	Cabeçalhos que contêm 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 instrui o Amazon S3 a usar. Para obter mais informações sobre a escolha do algoritmo de soma de verificação, consulte Verificar a integridade do objeto .	Não
x-amz-version-id	A versão do objeto. Quando você ativa o versionamento em um bucket, o Amazon S3 atribui um ID de versão aos objetos adicionados ao bucket. Para ter mais informações, consulte Usando o versionamento em buckets do S3 .	Não

Nome	Descrição	O usuário pode modificar o valor?
<code>x-amz-delete-marker</code>	Um marcador booleano que indica se o objeto é um marcador de exclusão. Esse marcador é usado somente em buckets com versionamento ativado.	Não
<code>x-amz-storage-class</code>	A classe de armazenamento usada para armazenar o objeto. Para ter mais informações, consulte Uso de classes de armazenamento do Amazon S3 .	Sim
<code>x-amz-website-redirect-location</code>	Um cabeçalho que redireciona solicitações do objeto associado para outro objeto no mesmo bucket ou para um URL externo. Para ter mais informações, consulte (Opcional) Configurar um redirecionamento de uma página da Web .	Sim
<code>x-amz-server-side-encryption-aws-kms-key-id</code>	Um cabeçalho que indica o ID da chave do KMS de criptografia simétrica do AWS KMS que foi usada para criptografar o objeto. Esse cabeçalho é usado somente quando o cabeçalho <code>x-amz-server-side-encryption</code> está presente e tem o valor de <code>aws:kms</code> .	Sim
<code>x-amz-server-side-encryption-customer-algorithm</code>	Um cabeçalho que indica se a criptografia do lado do servidor com as chaves de criptografia fornecidas pelo cliente (SSE-C) está ativada. Para ter mais informações, consulte Como usar criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C) .	Sim
<code>x-amz-tagging</code>	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 não US-ASCII em seus valores de metadados, a string Unicode fornecida é examinada quanto a caracteres não US-ASCII. 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: DOC-EXAMPLE-BUCKET1.s3.amazonaws.com
x-amz-meta-nonascii: ÄMÄZÖÑ S3

HEAD /Key HTTP/1.1
Host: DOC-EXAMPLE-BUCKET1.s3.amazonaws.com
x-amz-meta-nonascii: =?UTF-8?B?w4PChE3Dg8KEWsODwpXDg8KRIFMz?=?

PUT /Key HTTP/1.1
Host: DOC-EXAMPLE-BUCKET1.s3.amazonaws.com
x-amz-meta-ascii: AMAZONS3

HEAD /Key HTTP/1.1
Host: DOC-EXAMPLE-BUCKET1.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 objeto depois do upload criando uma cópia, modificando e substituindo o objeto antigo ou criando outra versão dele, consulte [Editar metadados de objeto no console do Amazon S3](#).

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 e Last-Modified são campos de metadados de objetos definidos pelo sistema que não podem ser modificados por um usuário.

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 ao perfil 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](#).

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](#).

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 [Gerenciamento de identidade e acesso para o Amazon S3](#).

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 Command Line Interface (AWS CLI), AWS SDKs 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](#).

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 operação de API de upload de várias partes, é possível fazer upload de um único objeto grande, com até 5 TB.

A operação de API de upload de várias partes 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 ter mais informações, consulte [Carregar e copiar objetos usando multipart upload](#).

Quando você faz upload de um objeto, ele é automaticamente criptografado usando criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) por padrão. Quando você faz download, o objeto é descriptografado. Para obter mais informações, consulte [Definir o comportamento padrão da criptografia para os buckets do Amazon S3](#) e [Proteger dados com criptografia](#).

Ao fazer upload de um objeto, se quiser usar um tipo diferente de criptografia padrão, você também pode especificar a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS) nas solicitações PUT do S3 ou definir a configuração de criptografia padrão no bucket de destino para usar o SSE-KMS para criptografar os dados. Para obter mais informações sobre SSE-KMS, consulte [Especificando criptografia no lado do servidor com o AWS KMS \(SSE-KMS\)](#). Se quiser usar uma chave do KMS que seja de propriedade de outra conta, você deverá ter permissão para usar a chave. Para obter mais informações sobre permissões entre contas para chaves do KMS, consulte [Criar chaves do KMS que outras contas podem usar](#) no Guia do desenvolvedor do AWS Key Management Service.

Se você encontrar um erro de Acesso negado (403 Proibido) no Amazon S3, consulte [Solucionar erros de acesso negado \(403 proibido\) no Amazon S3](#) para saber mais sobre as causas comuns.

Uso do console do S3

Este procedimento explica como fazer upload de objetos e pastas para um bucket do Amazon 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](#).

Note

Se você renomear um objeto ou alterar qualquer uma das propriedades no console do Amazon S3, por exemplo Classe de armazenamento, Criptografia ou Metadados, será criado outro objeto 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. O perfil 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 `images`, os objetos `images/sample1.jpg` e `images/sample2.jpg` são exibidos como `sample1.jpg` e `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. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, escolha o nome do bucket no qual você deseja carregar suas pastas ou arquivos.
4. Escolha Upload (Fazer upload).
5. Na janela Upload (Fazer upload), siga um destes procedimentos:
 - Arraste e solte arquivos e pastas para a janela Upload (Fazer upload) .

- Escolha Adicionar arquivo ou Adicionar pasta, escolha arquivos ou pastas para fazer upload e depois escolha Abrir.
6. Para habilitar o versionamento, em Destination (Destino), escolha Enable Bucket Versioning (Ativar versionamento de bucket).
 7. 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 upload 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](#). 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 Classe de armazenamento escolha a classe de armazenamento para os arquivos que você está carregando.

Para obter mais informações sobre classes de armazenamento, consulte [Uso de classes de armazenamento do Amazon S3](#).

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. Em Configurações de criptografia, escolha Usar configurações de bucket para criptografia padrão ou Substituir configurações de bucket para criptografia padrão.
 - c. Se você escolher Substituir configurações do bucket para criptografia padrão, deverá definir as configurações de criptografia a seguir.

- Para criptografar os arquivos carregados usando chaves gerenciadas pelo Amazon S3, escolha Chave gerenciada pelo Amazon S3 (SSE-S3).

Para ter mais informações, consulte [Usar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#).

- Para criptografar os arquivos carregados usando chaves armazenadas no AWS Key Management Service (AWS KMS), selecione Chave do AWS Key Management Service (SSE-KMS). Depois, escolha uma das seguintes opções para a chave do AWS KMS:
 - Para escolher entre uma lista de chaves do KMS disponíveis, selecione Escolher de sua AWS KMS keys e escolha a chave do KMS na lista de chaves disponíveis.

As chaves Chave gerenciada pela AWS (aws/s3) e as chaves gerenciadas pelo cliente são exibidas nessa lista. Para ter mais informações sobre chaves gerenciadas pelo cliente, consulte [Chaves de clientes e chaves da AWS](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para inserir o ARN da chave do KMS, selecione Inserir ARN da AWS KMS key e insira o ARN da chave do KMS no campo exibido.
- Para criar uma chave gerenciada pelo cliente no console do AWS KMS, selecione Criar uma chave do KMS.

Para ter mais informações sobre como criar uma AWS KMS key, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

Important

Você só pode usar chaves do KMS disponíveis na mesma Região da AWS que o bucket. O console do Amazon S3 lista somente as primeiras 100 chaves do KMS na mesma região que o bucket. Para usar uma chave do KMS que não esteja listada, você deve inserir o ARN da chave do KMS. Se quiser usar uma chave do KMS que seja de propriedade de outra conta, primeiro você deverá ter permissão para usar a chave e, depois, inserir o ARN da chave do KMS.

O Amazon S3 só é compatível com chaves do KMS de criptografia simétrica, e não com chaves assimétricas do KMS. Para ter mais informações, consulte [Identificar chaves do KMS simétricas e assimétricas](#) no Guia do desenvolvedor do AWS Key Management Service.

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](#).

7. Para adicionar tags a todos os objetos que você está carregando, escolha Add tag (Adicionar tag). Insira um nome de tag no campo Chave. Insira 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. Uma 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](#).

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](#). 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](#).

- 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 os AWS SDKs para fazer upload de objetos no Amazon S3. Os SDKs fornecem 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](#).

```
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 criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3](#).

```
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 { PutObjectCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new PutObjectCommand({
    Bucket: "test-bucket",
    Key: "hello-s3.txt",
    Body: "Hello S3!",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

PHP

Este exemplo orienta você quanto ao 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 operação de API de upload de várias partes. Para ter mais informações, consulte [Carregar e copiar objetos usando multipart upload](#).

Este exemplo considera que você já está seguindo as instruções para [Usar o AWS SDK for PHP e executar exemplos do PHP](#) e tem o AWS SDK for PHP devidamente instalado.

Exemplo 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](#).

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

use Aws\S3\Exception\S3Exception;
```

```
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

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

try {
    // Upload data.
    $result = $s3->putObject([
        'Bucket' => $bucket,
        'Key'    => $keyname,
        'Body'   => 'Hello, world!',
        'ACL'    => 'public-read'
    ]);

    // Print the URL to the object.
    echo $result['ObjectURL'] . PHP_EOL;
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Ruby

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

# Example usage:
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

# Example usage:
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 carregar 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.

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#). Para obter mais informações sobre como usar multipart upload com a classe S3 Express One Zone e buckets de diretório, consulte [Usar uploads multiparte com buckets de diretório](#).

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.

Quando você faz upload de uma parte, o Amazon S3 retorna um cabeçalho de etiqueta de entidade (ETag) para a parte 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. Cada parte terá sua própria ETag no momento do upload. No entanto, quando o upload multipart estiver concluído e todas as partes estiverem consolidadas, todas as partes estarão sob uma ETag como uma soma de verificação das somas de verificação.

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 carregamentos fracionados em andamento, você precisará enviar solicitações adicionais para recuperar os carregamentos 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`. Se você quiser recuperar os valores da soma de verificação para partes individuais de carregamentos fracionados ainda em andamento, você pode usar `ListParts`.

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](#).

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.

Essas partes são cobradas de acordo com a classe de armazenamento especificada quando as peças foram carregadas. Uma exceção a isso são partes carregadas no S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive. As partes fracionadas em andamento para um PUT na classe de armazenamento S3 Glacier Flexible Retrieval são cobradas como S3 Glacier Flexible Retrieval Staging Storage de acordo com as taxas de armazenamento da classe S3 Standard até que o carregamento seja concluído. Além disso, tanto o `CreateMultipartUpload` quanto o `UploadPart` são cobrados de acordo com as taxas do S3 Standard. Somente a solicitação `CompleteMultipartUpload` é cobrada de acordo com a taxa do S3 Glacier Flexible Retrieval. De maneira semelhante, as partes fracionadas em andamento para um PUT na classe de armazenamento S3 Glacier Deep Archive são cobradas como S3 Glacier Flexible Retrieval Staging Storage de acordo com as taxas de armazenamento da classe S3 Standard até que o carregamento seja concluído. Apenas a solicitação `CompleteMultipartUpload` cobrada de acordo com as taxas de S3 Glacier Deep Archive.

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. Não há cobrança de exclusão antecipada pela exclusão de carregamentos fracionados incompletos, independentemente da classe de armazenamento especificada. Para obter mais informações sobre a definição de preços, consulte [Definição de preços do Amazon S3](#).

Note

Para minimizar os custos de armazenamento, recomendamos que você configure uma regra de ciclo de vida para excluir uploads incompletos de várias partes após um determinado número de dias usando a ação `AbortIncompleteMultipartUpload`. Para obter mais informações sobre como criar uma regra de ciclo de vida para excluir uploads incompletos de várias partes, consulte [Configurando uma política de ciclo de vida de bucket para anular multipart uploads incompletos](#).

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.

Para conferir um passo a passo sobre upload multiparte que utiliza funções do AWS Lambda, consulte [Uploading large objects to Amazon S3 using multipart upload and transfer acceleration](#).

- [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

Ação	Permissões obrigatórias
	<p>es 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.</p>
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>

Ação	Permissões obrigatórias
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 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>

Ação	Permissões obrigatórias
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](#). Para obter informações sobre os usuários do IAM e as práticas recomendadas, consulte [Identidades do IAM \(usuários, grupos de usuários e perfis\)](#) no Guia do usuário do IAM.

Tópicos

- [Configurar uma política de ciclo de vida de bucket para anular uploads incompletos de várias partes](#)
- [Fazer upload de um objeto usando multipart upload](#)
- [Carregar um diretório usando a classe TransferUtility .NET de alto nível](#)
- [Listar multipart uploads](#)
- [Monitorar um multipart upload](#)
- [Abortar um multipart upload](#)
- [Copiar um objeto usando multipart upload](#)
- [Limites do multipart upload do Amazon S3](#)

Configurar uma política de ciclo de vida de bucket para anular uploads incompletos de várias partes

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](#).

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 forem concluídos dentro de um número especificado de dias após a inicialização. Quando um multipart upload não é concluído dentro do prazo especificado, ele se torna elegível para uma operação de anulação. O Amazon S3 interrompe o multipart upload e exclui todas as partes associadas a ele.

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` (o [prefixo do nome da chave de objeto](#)). Portanto, a regra 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 elegíveis 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](#).

Note

Se o multipart upload for concluído durante o período 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. Além disso, você não incorrerá em cobranças de exclusão antecipada do ciclo de vida do S3 ao remover qualquer parte incompleta do upload de várias partes.

Uso do console do S3

Para gerenciar multipart uploads incompletos automaticamente, use o console do S3 para criar uma regra de ciclo de vida para expirar bytes de multipart uploads incompletos em um bucket após determinado número de dias. O procedimento a seguir mostra como adicionar uma regra de ciclo de vida para excluir multipart uploads incompletos depois de 7 dias. Para obter mais informações sobre como adicionar regras de ciclo de vida, consulte [Definir uma configuração de ciclo de vida em um bucket](#).

Como adicionar uma regra de ciclo de vida para anular multipart uploads incompletos há mais de 7 dias

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 criar uma regra de ciclo de vida para todos os objetos com um prefixo específico, escolha Limit the scope of this rule using one or more filters (Limitar o escopo desta regra usando um ou mais filtros) e insira o prefixo no campo Prefix (Prefixo).
 - Para criar uma regra de ciclo de vida para todos os objetos no bucket, escolha This rule applies to all objects in the bucket (Esta regra se aplica a todos os objetos no 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. Em Lifecycle rule actions (Ações da regra de ciclo de vida), selecione Delete expired object delete markers or incomplete multipart uploads (Excluir marcadores de exclusão de objetos expirados ou multipart uploads incompletos).
7. Em Delete expired object delete markers or incomplete multipart uploads (Excluir marcadores de exclusão de objetos expirados ou multipart uploads incompletos), escolha Delete incomplete multipart uploads (Excluir multipart uploads incompletos).
8. No campo Number of days (Número de dias), insira o número de dias depois dos quais os multipart uploads incompletos devem ser excluídos (por exemplo, 7 dias).
9. Escolha Create rule.

Usando a AWS CLI

O comando `put-bucket-lifecycle-configuration` da AWS Command Line Interface (AWS CLI) a seguir adiciona a configuração de ciclo de vida para o bucket especificado. Para usar esse comando, substitua *user input placeholders* por suas informações.

```
aws s3api put-bucket-lifecycle-configuration \
  --bucket DOC-EXAMPLE-BUCKET1 \
  --lifecycle-configuration filename-containing-lifecycle-configuration
```

O exemplo a seguir mostra como adicionar uma regra de ciclo de vida para anular multipart uploads incompletos usando a AWS CLI. Ele inclui um exemplo de configuração de ciclo de vida JSON para anular multipart uploads incompletos há mais de 7 dias.

Para usar os comandos da CLI neste exemplo, substitua *user input placeholders* pelas suas informações.

Como adicionar uma regra de ciclo de vida para anular multipart uploads incompletos

1. Configure o AWS CLI. Para obter instruções, consulte [Desenvolvimento com o Amazon S3 usando a AWS CLI](#).
2. Salve o seguinte exemplo de configuração de ciclo de vida em um arquivo (por exemplo, *lifecycle.json*). Esse exemplo de configuração especifica um prefixo vazio, portanto aplica-se a todos os objetos no bucket. Para restringir a configuração a um subconjunto de objetos, especifique um prefixo.

```
{
```

```
"Rules": [  
  {  
    "ID": "Test Rule",  
    "Status": "Enabled",  
    "Filter": {  
      "Prefix": ""  
    },  
    "AbortIncompleteMultipartUpload": {  
      "DaysAfterInitiation": 7  
    }  
  }  
]
```

3. Execute o comando da CLI a seguir para definir essa configuração de ciclo de vida em seu bucket.

```
aws s3api put-bucket-lifecycle-configuration \  
--bucket DOC-EXAMPLE-BUCKET1 \  
--lifecycle-configuration file:///lifecycle.json
```

4. Para verificar se a configuração de ciclo de vida foi definida no bucket, recupere a configuração de ciclo de vida usando o comando `get-bucket-lifecycle` a seguir.

```
aws s3api get-bucket-lifecycle \  
--bucket DOC-EXAMPLE-BUCKET1
```

5. Para excluir a configuração de ciclo de vida, use o comando `delete-bucket-lifecycle` a seguir.

```
aws s3api delete-bucket-lifecycle \  
--bucket DOC-EXAMPLE-BUCKET1
```

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](#).

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 \(API de baixo nível\)](#).

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](#).

```
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](#).

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadFileMPUHighLevelAPITest
    {
        private const string bucketName = "**** provide bucket name ****";
        private const string keyName = "**** provide a name for the uploaded object ****";
        private const string filePath = "**** provide the full path name of the file to upload ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
        RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

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

```
        UploadFileAsync().Wait();
    }

    private static async Task UploadFileAsync()
    {
        try
        {
            var fileTransferUtility =
                new TransferUtility(s3Client);

            // Option 1. Upload a file. The file name is used as the object key
name.

            await fileTransferUtility.UploadAsync(filePath, bucketName);
            Console.WriteLine("Upload 1 completed");

            // Option 2. Specify object key name explicitly.
            await fileTransferUtility.UploadAsync(filePath, bucketName,
keyName);
            Console.WriteLine("Upload 2 completed");

            // Option 3. Upload data from a type of System.IO.Stream.
            using (var fileToUpload =
                new FileStream(filePath, FileMode.Open, FileAccess.Read))
            {
                await fileTransferUtility.UploadAsync(fileToUpload,
                    bucketName, keyName);
            }
            Console.WriteLine("Upload 3 completed");

            // Option 4. Specify advanced settings.
            var fileTransferUtilityRequest = new TransferUtilityUploadRequest
            {
                BucketName = bucketName,
                FilePath = filePath,
                StorageClass = S3StorageClass.StandardInfrequentAccess,
                PartSize = 6291456, // 6 MB.
                Key = keyName,
                CannedACL = S3CannedACL.PublicRead
            };
            fileTransferUtilityRequest.Metadata.Add("param1", "Value1");
            fileTransferUtilityRequest.Metadata.Add("param2", "Value2");

            await fileTransferUtility.UploadAsync(fileTransferUtilityRequest);
            Console.WriteLine("Upload 4 completed");
        }
    }
}
```

```
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
}
}
```

JavaScript

Example

Faça upload de um arquivo grande.

```
import {
  CreateMultipartUploadCommand,
  UploadPartCommand,
  CompleteMultipartUploadCommand,
  AbortMultipartUploadCommand,
  S3Client,
} from "@aws-sdk/client-s3";

const twentyFiveMB = 25 * 1024 * 1024;

export const createString = (size = twentyFiveMB) => {
  return "x".repeat(size);
};

export const main = async () => {
  const s3Client = new S3Client({});
  const bucketName = "test-bucket";
  const key = "multipart.txt";
  const str = createString();
  const buffer = Buffer.from(str, "utf8");

  let uploadId;
```

```
try {
  const multipartUpload = await s3Client.send(
    new CreateMultipartUploadCommand({
      Bucket: bucketName,
      Key: key,
    }),
  );

  uploadId = multipartUpload.UploadId;

  const uploadPromises = [];
  // Multipart uploads require a minimum size of 5 MB per part.
  const partSize = Math.ceil(buffer.length / 5);

  // Upload each part.
  for (let i = 0; i < 5; i++) {
    const start = i * partSize;
    const end = start + partSize;
    uploadPromises.push(
      s3Client
        .send(
          new UploadPartCommand({
            Bucket: bucketName,
            Key: key,
            UploadId: uploadId,
            Body: buffer.subarray(start, end),
            PartNumber: i + 1,
          }),
        )
        .then((d) => {
          console.log("Part", i + 1, "uploaded");
          return d;
        }),
    );
  }

  const uploadResults = await Promise.all(uploadPromises);

  return await s3Client.send(
    new CompleteMultipartUploadCommand({
      Bucket: bucketName,
      Key: key,
      UploadId: uploadId,
```

```

    MultipartUpload: {
      Parts: uploadResults.map(({ ETag }, i) => ({
        ETag,
        PartNumber: i + 1,
      })),
    },
  })),
);

// Verify the output by downloading the file from the Amazon Simple Storage
Service (Amazon S3) console.
// Because the output is a 25 MB string, text editors might struggle to open the
file.
} catch (err) {
  console.error(err);

  if (uploadId) {
    const abortCommand = new AbortMultipartUploadCommand({
      Bucket: bucketName,
      Key: key,
      UploadId: uploadId,
    });

    await s3Client.send(abortCommand);
  }
}
};

```

Example

Baixe um arquivo grande.

```

import { GetObjectCommand, S3Client } from "@aws-sdk/client-s3";
import { createWriteStream } from "fs";

const s3Client = new S3Client({});
const oneMB = 1024 * 1024;

export const getObjectRange = ({ bucket, key, start, end }) => {
  const command = new GetObjectCommand({
    Bucket: bucket,
    Key: key,
    Range: `bytes=${start}-${end}`,
  });
};

```

```
    return s3Client.send(command);
};

/**
 * @param {string | undefined} contentRange
 */
export const getRangeAndLength = (contentRange) => {
  const [range, length] = contentRange.split("/");
  const [start, end] = range.split("-");
  return {
    start: parseInt(start),
    end: parseInt(end),
    length: parseInt(length),
  };
};

export const isComplete = ({ end, length }) => end === length - 1;

// When downloading a large file, you might want to break it down into
// smaller pieces. Amazon S3 accepts a Range header to specify the start
// and end of the byte range to be downloaded.
const downloadInChunks = async ({ bucket, key }) => {
  const writeStream = createWriteStream(
    fileURLToPath(new URL(`./${key}`, import.meta.url)),
  ).on("error", (err) => console.error(err));

  let rangeAndLength = { start: -1, end: -1, length: -1 };

  while (!isComplete(rangeAndLength)) {
    const { end } = rangeAndLength;
    const nextRange = { start: end + 1, end: end + oneMB };

    console.log(`Downloading bytes ${nextRange.start} to ${nextRange.end}`);

    const { ContentRange, Body } = await getObjectRange({
      bucket,
      key,
      ...nextRange,
    });

    writeStream.write(await Body.transformToByteArray());
    rangeAndLength = getRangeAndLength(ContentRange);
  }
}
```



```
};

export const main = async () => {
  await downloadInChunks({
    bucket: "my-cool-bucket",
    key: "my-cool-object.txt",
  });
};
```

Go

Example

Faça upload de um objeto grande usando um gerenciador de uploads para dividir os dados em partes e fazer upload delas simultaneamente.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3) actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform bucket
// and object actions.
type BucketBasics struct {
  S3Client *s3.Client
}
```

```
// UploadLargeObject uses an upload manager to upload data to an object in a bucket.
// The upload manager breaks large data into parts and uploads the parts
// concurrently.
func (basics BucketBasics) UploadLargeObject(bucketName string, objectKey string,
  largeObject []byte) error {
  largeBuffer := bytes.NewReader(largeObject)
  var partMiBs int64 = 10
  uploader := manager.NewUploader(basics.S3Client, func(u *manager.Uploader) {
    u.PartSize = partMiBs * 1024 * 1024
  })
  _, err := uploader.Upload(context.TODO(), &s3.PutObjectInput{
    Bucket: aws.String(bucketName),
    Key:    aws.String(objectKey),
    Body:   largeBuffer,
  })
  if err != nil {
```

```
    log.Printf("Couldn't upload large object to %v:%v. Here's why: %v\n",
        bucketName, objectKey, err)
}

return err
}
```

Example

Faça download de um objeto grande usando um gerenciador de downloads para obter os dados em partes e baixá-las simultaneamente.

```
// DownloadLargeObject uses a download manager to download an object from a bucket.
// The download manager gets the data in parts and writes them to a buffer until all
// of
// the data has been downloaded.
func (basics BucketBasics) DownloadLargeObject(bucketName string, objectKey string)
([]byte, error) {
    var partMiBs int64 = 10
    downloader := manager.NewDownloader(basics.S3Client, func(d *manager.Downloader) {
        d.PartSize = partMiBs * 1024 * 1024
    })
    buffer := manager.NewWriteAtBuffer([]byte{})
    _, err := downloader.Download(context.TODO(), buffer, &s3.GetObjectInput{
        Bucket: aws.String(bucketName),
        Key:    aws.String(objectKey),
    })
    if err != nil {
        log.Printf("Couldn't download large object from %v:%v. Here's why: %v\n",
            bucketName, objectKey, err)
    }
    return buffer.Bytes(), err
}
```

PHP

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](#) 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](#).

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

use Aws\Exception\MultipartUploadException;
use Aws\S3\MultipartUploader;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

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

// Prepare the upload parameters.
$uploader = new MultipartUploader($s3, '/path/to/large/file.zip', [
    'bucket' => $bucket,
    'key'    => $keyname
]);

// Perform the upload.
try {
    $result = $uploader->upload();
    echo "Upload complete: {$result['ObjectURL']}" . PHP_EOL;
} catch (MultipartUploadException $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Python

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](#)). 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 \(API de alto nível\)](#)).

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](#).

```
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](#).

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](#).

```
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(initWithRequest);

    // 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](#) 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](#).

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

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

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';
$filename = '*** Path to and Name of the File to Upload ***';

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

$result = $s3->createMultipartUpload([
    'Bucket'      => $bucket,
    'Key'         => $keyname,
    'StorageClass' => 'REDUCED_REDUNDANCY',
    'Metadata'   => [
        'param1' => 'value 1',
        'param2' => 'value 2',
        'param3' => 'value 3'
    ]
]);
$uploadId = $result['UploadId'];

// Upload the file in parts.
try {
    $file = fopen($filename, 'r');
    $partNumber = 1;
    while (!feof($file)) {
        $result = $s3->uploadPart([
            'Bucket'      => $bucket,
            'Key'         => $keyname,
            'UploadId'   => $uploadId,
            'PartNumber' => $partNumber,
```

```
        'Body'      => fread($file, 5 * 1024 * 1024),
    ]);
    $parts['Parts'][$partNumber] = [
        'PartNumber' => $partNumber,
        'ETag' => $result['ETag'],
    ];
    $partNumber++;

    echo "Uploading part $partNumber of $filename." . PHP_EOL;
}
fclose($file);
} catch (S3Exception $e) {
    $result = $s3->abortMultipartUpload([
        'Bucket' => $bucket,
        'Key' => $keyname,
        'UploadId' => $uploadId
    ]);

    echo "Upload of $filename failed." . PHP_EOL;
}

// Complete the multipart upload.
$result = $s3->completeMultipartUpload([
    'Bucket' => $bucket,
    'Key' => $keyname,
    'UploadId' => $uploadId,
    'MultipartUpload' => $parts,
]);
$url = $result['Location'];

echo "Uploaded $filename to $url." . PHP_EOL;
```

Usando a 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](#).

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](#)

Usando a 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](#). Para obter mais informações sobre os SDKs, consulte [Fazer upload de um objeto usando multipart upload](#).

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](#).

```
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](#).

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](#).

```
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](#) 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 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](#).

```
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](#).

```
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](#). 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 upload incompleto de várias partes usando uma configuração de ciclo de vida do 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 `TransferManager` .
- 2 Execute o método `TransferManager.abortMultipartUploads` passando o nome do bucket e um valor de `Date`.

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](#).

```
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 \(API de baixo nível\)](#).

.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](#).

```
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 \(API de baixo nível\)](#).

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](#).

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 \(API de alto nível\)](#).

.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 \(API de baixo nível\)](#).

```
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 \(API de alto nível\)](#).

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](#) 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](#).

```
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.

Usando a 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 [Copiar e mover objetos](#).

Uso da SDKs AWS

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](#).

```
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 dos AWS SDKs](#). Para obter mais informações sobre multipart uploads do Amazon S3, consulte [Carregar e copiar objetos usando multipart upload](#).

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](#).

```
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 próprias solicitações REST ou use um dos SDKs que fornecemos. Para obter mais informações sobre como usar o multipart upload com a AWS CLI, consulte [Usando a AWS CLI](#). Para obter mais informações sobre os SDKs, consulte [Suporte do AWS SDK para upload fracionado](#).

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](#).

Item	Especificação
Tamanho máximo do objeto	5 TiB
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.

Item	Especificação
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

Copiar e mover objetos

A operação `CopyObject` cria uma cópia de um objeto que já está armazenado no Amazon S3.

Você pode criar uma cópia do objeto de até 5 GB em uma única operação atômica. Contudo, para copiar um objeto maior do que 5 GB, você deve usar um carregamento fracionado. Para ter mais informações, consulte [the section called “Copiar um objeto”](#).

Usando a operação `CopyObject` você pode:

- Criar cópias adicionais de objetos.
- Renomear objetos copiando-os e excluindo os originais.
- Copiar ou mover objetos de um bucket para outro, inclusive entre Regiões da AWS (por exemplo, de `us-west-1` para `eu-west-2`). Ao mover um objeto, o Amazon S3 o copia para o destino especificado e, em seguida, exclui o objeto de origem.

Note

A cópia ou movimentação de objetos entre Regiões da AWS gera cobranças de largura de banda. Para obter mais informações, consulte [Preços do Amazon S3](#).

- Alterar metadados do objeto. Cada objeto do Amazon S3 tem metadados. Esses metadados são um conjunto de pares de nome-valor. Você pode definir metadados de objeto no momento em que carregar um objeto. Depois que você carregá-lo, não poderá modificar os respectivos metadados. A única forma de modificar metadados de objeto é fazer uma cópia do objeto e definir os metadados. Para isso, na operação de cópia, defina o mesmo objeto como origem e destino.

Alguns metadados de objeto são metadados de sistema e outros são definidos pelo usuário. É possível controlar alguns dos metadados do sistema. Por exemplo, é possível controlar a classe de armazenamento e o tipo de criptografia do lado do servidor a ser usado para o objeto. 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 de metadados controlados pelo sistema 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](#).

Copiar objetos arquivados e restaurados

Se o objeto de origem estiver arquivado em S3 Glacier Flexible Retrieval ou 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 \(arquivamento de objetos\)](#).

No console do Amazon S3, não é possível usar a operação Copiar para objetos restaurados nas classes de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive. Para copiar esses objetos restaurados, use a AWS Command Line Interface (AWS CLI), os SDKs da AWS ou a API REST do Amazon S3.

Copiar objetos criptografados

O Amazon S3 criptografa automaticamente todos os novos objetos que são copiados para um bucket do S3. Se você não especificar informações de criptografia na solicitação de cópia, a configuração de

criptografia do objeto de destino será definida como a configuração de criptografia padrão do bucket de destino. Por padrão, todos os buckets têm um nível básico de configuração de criptografia que usa criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3). Se o bucket de destino tiver uma configuração de criptografia padrão que usa criptografia do lado do servidor com uma chave do AWS Key Management Service (AWS KMS) (SSE-KMS) ou uma chave de criptografia fornecida pelo cliente (SSE-C), o Amazon S3 usará a chave do KMS correspondente ou uma chave fornecida pelo cliente para criptografar a cópia do objeto de destino.

Ao copiar um objeto, se quiser usar um tipo diferente de configuração de criptografia para o objeto de destino, você poderá solicitar que o Amazon S3 criptografe o objeto de destino com uma chave do KMS, uma chave gerenciada pelo Amazon S3 ou uma chave fornecida pelo cliente. Se a configuração de criptografia na solicitação for diferente da configuração de criptografia padrão do bucket de destino, a configuração de criptografia na solicitação terá precedência. Se o objeto de origem da cópia for criptografado com SSE-C, você deverá fornecer as informações de criptografia necessárias na solicitação, de maneira que o Amazon S3 possa descriptografar o objeto para a cópia. Para ter mais informações, consulte [Proteger dados com criptografia](#).

Usar somas de verificação ao copiar objetos

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](#).

Copiar vários objetos em uma única solicitação

Para copiar mais de um objeto do Amazon S3 com uma única solicitação, você pode 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 operação de 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. É possível usar as operações em lote do S3 por meio do console do Amazon S3, da AWS CLI, de AWS SDKs ou da API REST. Para ter mais informações, consulte [the section called “Conceitos básicos do Batch Operations”](#).

Copiar objetos para buckets de diretório

Para obter informações sobre como copiar um objeto em um bucket de diretório, consulte [Copiar um objeto para um bucket de diretório](#). Para obter informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Para copiar um objeto

Para copiar um objeto, use os métodos a seguir.

Uso do console do S3

Note

- Ao copiar um objeto usando o console do Amazon S3, você deve ter a permissão `s3:ListAllMyBuckets`. O console precisa dessa permissão para validar a operação Copiar. Para ver exemplos das políticas que concedem essas permissões, consulte [the section called “Exemplos de políticas baseadas em identidade”](#).
- Objetos criptografados com chaves de criptografia fornecidas pelo cliente (SSE-C) não podem ser copiados usando o console do S3. Para copiar objetos criptografados com SSE-C, use a AWS CLI, o SDK da AWS ou a API REST do Amazon S3.
- O console do Amazon S3 não permite a cópia entre regiões de objetos criptografados com SSE-KMS. Para copiar objetos criptografados com SSE-KMS, use a AWS CLI, o SDK da AWS 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. No painel de navegação esquerdo, escolha Buckets e selecione a guia Buckets de uso geral. 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. No menu Ações, escolha Copiar na lista de opções exibida.

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 (Somadas 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](#).

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.

Uso dos 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 um carregamento fracionado. Para ter mais informações, consulte [Copiar um objeto usando multipart upload](#).

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](#).

```
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](#).

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 for .NET e instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET](#).

```
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 exemplo fornece orientações sobre como usar classes da versão 3 do AWS SDK for PHP para copiar um ou vários objetos no Amazon S3 de um bucket para outro ou no mesmo bucket.

Este exemplo considera que você já está seguindo as instruções para [Usar o AWS SDK for PHP e executar exemplos do PHP](#) e tem o AWS SDK for PHP devidamente instalado.

O exemplo do PHP a seguir mostra o uso do método `copyObject()` para copiar um único objeto no Amazon S3. Ele também demonstra como fazer várias cópias de um objeto usando um lote de chamadas para `CopyObject` com o método `getCommand()`.

Cópia de objetos

- 1 Crie uma instância de um cliente do Amazon S3 usando o construtor da classe `Aws\S3\S3Client`.
- 2 Para fazer várias cópias de um objeto, execute um lote de chamadas para o método `getCommand()` do cliente do Amazon S3, que é herdado da classe `Aws\CommandInterface`. Você fornece o comando `CopyObject` 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\CommandPool;
use Aws\Exception\AwsException;
use Aws\ResultInterface;
use Aws\S3\S3Client;

$sourceBucket = '*** Your Source Bucket Name ***';
$sourceKeyname = '*** Your Source Object Key ***';
$targetBucket = '*** Your Target Bucket Name ***';

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

// Copy an object.
$s3->copyObject([
    'Bucket' => $targetBucket,
    'Key' => "$sourceKeyname-copy",
    'CopySource' => "$sourceBucket/$sourceKeyname",
]);
```



```
// Perform a batch of CopyObject operations.
$batch = array();
for ($i = 1; $i <= 3; $i++) {
    $batch[] = $s3->getCommand('CopyObject', [
        'Bucket' => $targetBucket,
        'Key' => "{targetKeyname}-$i",
        'CopySource' => "$sourceBucket/$sourceKeyname",
    ]);
}
try {
    $results = CommandPool::batch($s3, $batch);
    foreach ($results as $result) {
        if ($result instanceof ResultInterface) {
            // Result handling here
        }
        if ($result instanceof AwsException) {
            // AwsException handling here
        }
    }
} catch (Exception $e) {
    // General error handling here
}
```

Python

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource in
        Boto3
                           that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key
```

```
def copy(self, dest_object):
    """
    Copies the object to another bucket.
```

```
        :param dest_object: The destination object initialized with a bucket and
key.

        This is a Boto3 Object resource.

    """
    try:
        dest_object.copy_from(
            CopySource={"Bucket": self.object.bucket_name, "Key":
self.object.key}
        )
        dest_object.wait_until_exists()
        logger.info(
            "Copied object from %s:%s to %s:%s.",
            self.object.bucket_name,
            self.object.key,
            dest_object.bucket_name,
            dest_object.key,
        )
    except ClientError:
        logger.exception(
            "Couldn't copy object from %s/%s to %s/%s.",
            self.object.bucket_name,
            self.object.key,
            dest_object.bucket_name,
            dest_object.key,
        )
        raise
```

Ruby

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](#).
- 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
end

# Example usage:
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__

```

Uso da API REST

Este exemplo descreve como copiar um objeto usando a API REST do S3. Para obter mais informações sobre a API REST, consulte [CopyObject](#).

Este exemplo copia o objeto `flotsam` do bucket `DOC-EXAMPLE-BUCKET1` para o objeto `jetsam` do bucket `DOC-EXAMPLE-BUCKET2`, preservando seus metadados.

```

PUT /jetsam HTTP/1.1
Host: DOC-EXAMPLE-BUCKET2.s3.amazonaws.com
x-amz-copy-source: /DOC-EXAMPLE-BUCKET1/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:/DOC-EXAMPLE-BUCKET1/flotsam\r\n
/DOC-EXAMPLE-BUCKET2/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: Vyaxt7qEbvz34BnSu5hctyyNSlHTYZFMWK4Ftz0+iX8JQNyaLdTshL0Kxatba0Zt
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>
```

Usando a AWS CLI

Você também pode usar a AWS Command Line Interface (AWS CLI) para copiar um objeto do S3. Para obter mais informações, consulte [copy-object](#) 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.

Como mover um objeto

Para mover um objeto, use os métodos a seguir.

Uso do console do S3

Note

- Objetos criptografados com chaves de criptografia fornecidas pelo cliente (SSE-C) não podem ser movidos usando o console do S3. Para mover objetos criptografados com SSE-C, use a AWS CLI, SDKs da AWS ou a API REST do Amazon S3.
- Ao mover pastas, aguarde a conclusão da operação Mover antes de fazer alterações adicionais nas pastas.
- No console do Amazon S3, não é possível usar aliases de ponto de acesso do S3 como origem ou destino para operações Mover.

Como mover um objeto

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 Buckets e selecione a guia Buckets de uso geral. 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. No menu Ações, escolha Mover.
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.
- Essa ação atualiza metadados para versionamento de bucket, criptografia, recursos de bloqueio de objetos e objetos arquivados.

Usando a AWS CLI

Você também pode usar a AWS Command Line Interface (AWS CLI) para mover um objeto do S3. Para obter mais informações, consulte [mv](#) 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.

Baixar objetos

Esta seção explica como baixar objetos de um bucket do Amazon S3. Com o Amazon S3, é possível armazenar objetos em um ou mais buckets e cada objeto pode ter até 5 TB de tamanho. Qualquer objeto do Amazon S3 que não esteja arquivado pode ser acessado em tempo real. Os objetos arquivados, no entanto, devem ser restaurados antes de serem baixados. Para obter informações sobre como baixar objetos arquivados, consulte [the section called “Baixar objetos arquivados”](#).

É possível baixar um único objeto usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI), AWS SDKs ou a API REST do Amazon S3. Para baixar um objeto do S3 sem escrever nenhum código ou executar nenhum comando, use o console do S3. Para ter mais informações, consulte [the section called “Fazer download de um objeto”](#).

Para baixar vários objetos, use o AWS CloudShell, a AWS CLI ou os AWS SDKs. Para ter mais informações, consulte [the section called “Baixar vários objetos”](#).

Se precisar baixar parte de um objeto, use parâmetros extras com a AWS CLI ou a API REST para especificar somente os bytes que você deseja baixar. Para ter mais informações, consulte [the section called “Baixar parte de um objeto”](#).

Se você precisar baixar um objeto que não seja seu, peça ao proprietário do objeto para gerar um URL pré-assinado que permita baixá-lo. Para ter mais informações, consulte [the section called “Baixar um objeto de outra Conta da AWS”](#).

Quando você baixa objetos fora da rede da AWS, aplicam-se taxas de transferência de dados. A transferência de dados dentro da rede da AWS é gratuita dentro da mesma Região da AWS, mas haverá cobrança por qualquer solicitação GET. Para obter mais informações sobre custos de transferência de dados e cobranças de recuperação de dados, consulte [Preços do Amazon S3](#).

Tópicos

- [Fazer download de um objeto](#)
- [Baixar vários objetos](#)
- [Baixar parte de um objeto](#)
- [Baixar um objeto de outra Conta da AWS](#)
- [Baixar objetos arquivados](#)
- [Solução de problemas de download de objetos](#)

Fazer download de um objeto

É possível baixar um objeto usando o console do Amazon S3, a AWS CLI, AWS SDKs ou a API REST.

Uso do console do S3

Esta seção explica como usar o console do Amazon S3 para fazer baixar um objeto de um bucket do S3.

Note

- Só é possível baixar um objeto por vez.
- Se você usar o console do Amazon S3 para baixar um objeto cujo nome da chave termine com um ponto (.), o ponto será removido do nome da chave do objeto baixado. Para reter o ponto no final do nome do objeto baixado, você deve usar a AWS Command Line Interface (AWS CLI), AWS SDKs ou a API REST do Amazon S3.

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:
 - Marque a caixa de seleção ao lado do objeto e escolha Fazer download. Se você quiser baixar o objeto em uma pasta específica, no menu Ações, escolha Fazer download como.
 - Se você quiser baixar uma versão específica do objeto, ative Mostrar versões (ao lado da caixa de pesquisa). Marque a caixa de seleção ao lado da versão do objeto que você deseja e escolha Fazer Download. Se você quiser baixar o objeto em uma pasta específica, no menu Ações, escolha Fazer download como.

Usando a AWS CLI

O exemplo de `get-object` a seguir mostra como você pode usar a AWS CLI para baixar um objeto do Amazon S3. Esse comando obtém o objeto `folder/my_image` do bucket `DOC-EXAMPLE-BUCKET1`. O objeto será baixado em um arquivo chamado `my_downloaded_image`.

```
aws s3api get-object --bucket DOC-EXAMPLE-BUCKET1 --key folder/  
my_image my_downloaded_image
```

Para obter mais informações e exemplos, consulte [get-object](#) na Referência de comandos da AWS CLI.

Uso dos AWS SDKs

Para ver exemplos de como baixar um objeto com os AWS SDKs, consulte [Usar GetObject com um AWS SDK ou ferramenta de linha de comando](#).

Para obter informações gerais sobre como usar AWS SDKs diferentes, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers](#).

Uso da API REST

É possível usar a API REST para recuperar objetos do Amazon S3. Para obter mais informações, consulte [GetObject](#) na Referência da API do Amazon Simple Storage Service.

Baixar vários objetos

Você pode baixar vários objetos usando o AWS CloudShell, a AWS CLI ou os AWS SDKs.

Usar AWS CloudShell no AWS Management Console

O AWS CloudShell é um shell pré-autenticado baseado em navegador que você pode iniciar diretamente do AWS Management Console.

Para obter mais informações sobre o AWS CloudShell, consulte [O que é o CloudShell?](#) no Guia do usuário do AWS CloudShell.

Important

Com o AWS CloudShell, o diretório inicial tem armazenamento de até 1 GB por Região da AWS. Portanto, você não pode sincronizar buckets com objetos cujo total ultrapasse

esse valor. Para consultar mais limitações, acesse [Cotas e restrições de serviço](#) no Guia do usuário do AWS CloudShell.

Para baixar objetos usando o AWS CloudShell

1. Faça login no AWS Management Console e abra o console do CloudShell em <https://console.aws.amazon.com/cloudshell/>.
2. Os comandos a seguir são executados para sincronizar objetos no bucket com o CloudShell. O comando a seguir sincroniza objetos do bucket chamado *DOC-EXAMPLE-BUCKET1* e cria uma pasta chamada *temp* no CloudShell. O CloudShell sincroniza os objetos com essa pasta. Para usar esse comando, substitua *user input placeholders* por suas informações.

```
aws s3 sync s3://DOC-EXAMPLE-BUCKET1 ./temp
```

Note

Para realizar a correspondência de padrões e excluir ou incluir objetos específicos, você pode usar os parâmetros `--exclude "value"` e `--include "value"` com o comando `sync`.

3. Execute o comando a seguir para compactar objetos na pasta chamada *temp* para um arquivo chamado *temp.zip*.

```
zip temp.zip -r temp/
```

4. Escolha Ações e Baixar arquivo.
5. Insira um nome para o arquivo **temp.zip** e escolha Baixar.
6. (Opcional) Exclua o arquivo *temp.zip* e os objetos que estão sincronizados com a pasta *temp* no CloudShell. Com o AWS CloudShell, você tem armazenamento persistente de até 1 GB para cada Região da AWS.

Você pode usar os exemplos de comandos a seguir para excluir o arquivo `.zip` e a pasta. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
rm temp.zip && rm -rf temp/
```

Usando a AWS CLI

Os exemplos a seguir mostram como você pode usar a AWS CLI para baixar todos os arquivos ou objetos no diretório ou prefixo especificado. Esse comando copia todos os objetos do bucket *DOC-EXAMPLE-BUCKET1* para o diretório atual. Para usar esse exemplo de comando, use o nome do seu bucket no lugar de *DOC-EXAMPLE-BUCKET1*.

```
aws s3 cp s3://DOC-EXAMPLE-BUCKET1 . --recursive
```

O comando a seguir baixa todos os objetos com o prefixo *logs* no bucket *DOC-EXAMPLE-BUCKET1* para o diretório atual. Ele também usa os parâmetros `--exclude` e `--include` para copiar somente objetos com o sufixo *.log*. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3 cp s3://DOC-EXAMPLE-BUCKET1/logs/ . --recursive --exclude "*" --include "*.log"
```

Para obter mais informações e exemplos, consulte [cp](#) na Referência de comandos da AWS CLI.

Uso dos AWS SDKs

Para ver exemplos de como baixar todos os objetos em um bucket do Amazon S3 com os AWS SDKs, acesse [Fazer download de todos os objetos de um bucket do Amazon Simple Storage Service \(Amazon S3\) em um diretório local](#).

Para obter informações gerais sobre como usar AWS SDKs diferentes, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers](#).

Baixar parte de um objeto

É possível baixar parte de um objeto usando a AWS CLI ou a API REST. Para fazer isso, use parâmetros adicionais a fim de especificar qual parte de um objeto você deseja baixar.

Usando a AWS CLI

O exemplo de comando a seguir executa uma solicitação GET para um intervalo de bytes no objeto chamado *folder/my_data* no bucket denominado *DOC-EXAMPLE-BUCKET1*. Na solicitação, o intervalo de bytes deve ter `bytes=` como prefixo. O objeto parcial é baixado para o arquivo de saída chamado *my_data_range*. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3api get-object --bucket DOC-EXAMPLE-BUCKET1 --key folder/my_data --range
bytes=0-500 my_data_range
```

Para obter mais informações e exemplos, consulte [get-object](#) na Referência de comandos da AWS CLI.

Para obter mais informações sobre o cabeçalho HTTP Range, consulte [RFC 9110](#) no site do RFC Editor.

Note

O Amazon S3 não comporta a recuperação de vários intervalos de dados em uma única solicitação GET.

Uso da API REST

Você pode usar os parâmetros `partNumber` e `Range` na API REST para recuperar partes de objetos do Amazon S3. Para obter mais informações, consulte [GetObject](#) na Referência da API do Amazon Simple Storage Service.

Baixar um objeto de outra Conta da AWS

É possível usar um URL pré-assinado para conceder acesso por tempo limitado aos objetos sem atualizar a política de bucket.

O URL pré-assinado pode ser inserido em um navegador ou usado por um programa para baixar um objeto. As credenciais usadas pelo URL são as do usuário da AWS que gerou o URL. Depois que o URL for criado, qualquer pessoa com o URL pré-assinado poderá baixar o objeto correspondente até que o URL expire.

Usar um URL pré-assinado no console do S3

É possível usar o console do Amazon S3 para gerar um URL pré-assinado para compartilhar um objeto seguindo estas etapas. Ao usar o console, 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 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. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, selecione o nome do bucket que contém os objetos para os quais você deseja obter um URL pré-assinado.
4. Na lista Objects (Objetos), selecione o objeto para o qual deseja criar um URL pré-assinado.
5. No menu Ações de objeto, escolha Compartilhar com um URL pré-assinado.
6. Especifique por quanto tempo deseja que o URL pré-assinado seja válido.
7. Escolha Create presigned URL (Criar URL pré-assinado).
8. Quando uma mensagem de 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.
9. Para baixar o objeto, cole o URL em qualquer navegador para que o objeto tente fazer download.

Para obter mais informações sobre URLs pré-assinados e outros métodos para criá-los, consulte [Trabalhar com URLs pré-assinados](#).

Baixar objetos arquivados

Para reduzir os custos de armazenamento de objetos acessados com pouca frequência, você pode arquivar esses objetos. Quando você arquiva um objeto, ele é movido para um armazenamento de baixo custo, o que significa que não é possível acessá-lo em tempo real. Para baixar um objeto arquivado, você deve restaurá-lo primeiro.

Os objetos arquivados podem ser restaurados em minutos ou horas, dependendo da classe de armazenamento. Você pode restaurar um objeto arquivado usando o console do Amazon S3, Operações em Lote do S3, a API REST do Amazon S3, os AWS SDKs e a AWS Command Line Interface (AWS CLI).

Para obter instruções, consulte [Restaurar um objeto arquivado](#). Depois de restaurar o objeto arquivado, você poderá baixá-lo.

Solução de problemas de download de objetos

Permissões insuficientes ou políticas incorretas de bucket ou de usuário do AWS Identity and Access Management (IAM) podem causar erros ao tentar baixar objetos do Amazon S3. Esses problemas geralmente podem provocar erros de acesso negado (403 proibido), nos quais o Amazon S3 não consegue permitir o acesso a um recurso.

Para ver as causas comuns de erros de acesso negado (403 proibido), consulte [Solucionar erros de acesso negado \(403 proibido\) no Amazon S3](#).

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. Para saber mais sobre como usar somas de verificação adicionais para verificar a integridade dos dados, consulte [Tutorial: Verificar a integridade dos dados no Amazon S3 com somas de verificação adicionais](#).

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 cíclica (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](#).

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](#).

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](#) e [Tutorial: verificar a integridade dos dados no Amazon S3 com somas de verificação adicionais](#).

Uso dos 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](#).

```
import software.amazon.awssdk.auth.credentials.AwsCredentials;
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;
import software.amazon.awssdk.core.ResponseInputStream;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.AbortMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.ChecksumAlgorithm;
import software.amazon.awssdk.services.s3.model.ChecksumMode;
import software.amazon.awssdk.services.s3.model.CompleteMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.CompleteMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.CompletedMultipartUpload;
import software.amazon.awssdk.services.s3.model.CompletedPart;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.GetObjectAttributesRequest;
import software.amazon.awssdk.services.s3.model.GetObjectAttributesResponse;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.GetObjectTaggingRequest;
import software.amazon.awssdk.services.s3.model.ObjectAttributes;
import software.amazon.awssdk.services.s3.model.PutObjectTaggingRequest;
import software.amazon.awssdk.services.s3.model.Tag;
import software.amazon.awssdk.services.s3.model.Tagging;
import software.amazon.awssdk.services.s3.model.UploadPartRequest;
import software.amazon.awssdk.services.s3.model.UploadPartResponse;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
```



```
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.nio.ByteBuffer;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.Base64;
import java.util.List;

public class LargeObjectValidation {
    private static String FILE_NAME = "sample.file";
    private static String BUCKET = "sample-bucket";
    //Optional, if you want a method of storing the full multipart object
checksum in S3.
    private static String CHECKSUM_TAG_KEYNAME = "fullObjectChecksum";
    //If you have existing full-object checksums that you need to validate
against, you can do the full object validation on a sequential upload.
    private static String SHA256_FILE_BYTES = "htCM5g7ZNdoSw8bN/
mkgiAhXt5MFoVowVg+LE9aIQmI=";
    //Example Chunk Size - this must be greater than or equal to 5MB.
    private static int CHUNK_SIZE = 5 * 1024 * 1024;

    public static void main(String[] args) {
        S3Client s3Client = S3Client.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(new AwsCredentialsProvider() {
                @Override
                public AwsCredentials resolveCredentials() {
                    return new AwsCredentials() {
                        @Override
                        public String accessKeyId() {
                            return Constants.ACCESS_KEY;
                        }

                        @Override
                        public String secretAccessKey() {
                            return Constants.SECRET;
                        }
                    };
                }
            })
            .build();
        uploadLargeFileBracketedByChecksum(s3Client);
    }
}
```

```

        downloadLargeFileBracketedByChecksum(s3Client);
        validateExistingFileAgainstS3Checksum(s3Client);
    }

    public static void uploadLargeFileBracketedByChecksum(S3Client s3Client) {
        System.out.println("Starting uploading file validation");
        File file = new File(FILE_NAME);
        try (InputStream in = new FileInputStream(file)) {
            MessageDigest sha256 = MessageDigest.getInstance("SHA-256");
            CreateMultipartUploadRequest createMultipartUploadRequest =
CreateMultipartUploadRequest.builder()
                .bucket(BUCKET)
                .key(FILE_NAME)
                .checksumAlgorithm(ChecksumAlgorithm.SHA256)
                .build();
            CreateMultipartUploadResponse createdUpload =
s3Client.createMultipartUpload(createMultipartUploadRequest);
            List<CompletedPart> completedParts = new ArrayList<CompletedPart>();
            int partNumber = 1;
            byte[] buffer = new byte[CHUNK_SIZE];
            int read = in.read(buffer);
            while (read != -1) {
                UploadPartRequest uploadPartRequest =
UploadPartRequest.builder()
                    .partNumber(partNumber).uploadId(createdUpload.uploadId()).key(FILE_NAME).bucket(BUCKET).ch
                        UploadPartResponse uploadedPart =
s3Client.uploadPart(uploadPartRequest,
RequestBody.fromByteBuffer(ByteBuffer.wrap(buffer, 0, read)));
                CompletedPart part =
CompletedPart.builder().partNumber(partNumber).checksumSHA256(uploadedPart.checksumSHA256())
                    completedParts.add(part);
                sha256.update(buffer, 0, read);
                read = in.read(buffer);
                partNumber++;
            }
            String fullObjectChecksum =
Base64.getEncoder().encodeToString(sha256.digest());
            if (!fullObjectChecksum.equals(SHA256_FILE_BYTES)) {
                //Because the SHA256 is uploaded after the part is uploaded; the
upload is bracketed and the full object can be fully validated.
            }

            s3Client.abortMultipartUpload(AbortMultipartUploadRequest.builder().bucket(BUCKET).key(FILE

```

```

        throw new IOException("Byte mismatch between stored checksum and
upload, do not proceed with upload and cleanup");
    }
    CompletedMultipartUpload completedMultipartUpload =
CompletedMultipartUpload.builder().parts(completedParts).build();
    CompleteMultipartUploadResponse completedUploadResponse =
s3Client.completeMultipartUpload(

CompleteMultipartUploadRequest.builder().bucket(BUCKET).key(FILE_NAME).uploadId(createdUplo

    Tag checksumTag =
Tag.builder().key(CHECKSUM_TAG_KEYNAME).value(fullObjectChecksum).build();
    //Optionally, if you need the full object checksum stored with the
file; you could add it as a tag after completion.

s3Client.putObjectTagging(PutObjectTaggingRequest.builder().bucket(BUCKET).key(FILE_NAME).t
    } catch (IOException | NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
    GetObjectAttributesResponse
        objectAttributes =
s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_N
        .objectAttributes(ObjectAttributes.OBJECT_PARTS,
ObjectAttributes.CHECKSUM).build());
    System.out.println(objectAttributes.objectParts().parts());
    System.out.println(objectAttributes.checksum().checksumSHA256());
}

    public static void downloadLargeFileBracketedByChecksum(S3Client s3Client) {
        System.out.println("Starting downloading file validation");
        File file = new File("DOWNLOADED_" + FILE_NAME);
        try (OutputStream out = new FileOutputStream(file)) {
            GetObjectAttributesResponse
                objectAttributes =
s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_N
                .objectAttributes(ObjectAttributes.OBJECT_PARTS,
ObjectAttributes.CHECKSUM).build());
            //Optionally if you need the full object checksum, you can grab a
tag you added on the upload
            List<Tag> objectTags =
s3Client.getObjectTagging(GetObjectTaggingRequest.builder().bucket(BUCKET).key(FILE_NAME).b
            String fullObjectChecksum = null;
            for (Tag objectTag : objectTags) {
                if (objectTag.key().equals(CHECKSUM_TAG_KEYNAME)) {
                    fullObjectChecksum = objectTag.value();
                }
            }
        }
    }
}

```

```

        break;
    }
}
    MessageDigest sha256FullObject =
MessageDigest.getInstance("SHA-256");
    MessageDigest sha256ChecksumOfChecksums =
MessageDigest.getInstance("SHA-256");

    //If you retrieve the object in parts, and set the ChecksumMode to
enabled, the SDK will automatically validate the part checksum
    for (int partNumber = 1; partNumber <=
objectAttributes.objectParts().totalPartsCount(); partNumber++) {
        MessageDigest sha256Part = MessageDigest.getInstance("SHA-256");
        ResponseInputStream<GetObjectResponse> response =
s3Client.getObject(GetObjectRequest.builder().bucket(BUCKET).key(FILE_NAME).partNumber(part
GetObjectResponse getObjectResponse = response.getResponse();
        byte[] buffer = new byte[CHUNK_SIZE];
        int read = response.read(buffer);
        while (read != -1) {
            out.write(buffer, 0, read);
            sha256FullObject.update(buffer, 0, read);
            sha256Part.update(buffer, 0, read);
            read = response.read(buffer);
        }
        byte[] sha256PartBytes = sha256Part.digest();
        sha256ChecksumOfChecksums.update(sha256PartBytes);
        //Optionally, you can do an additional manual validation again
the part checksum if needed in addition to the SDK check
        String base64PartChecksum =
Base64.getEncoder().encodeToString(sha256PartBytes);
        String base64PartChecksumFromObjectAttributes =
objectAttributes.objectParts().parts().get(partNumber - 1).checksumSHA256();
        if (!
base64PartChecksum.equals(getObjectResponse.checksumSHA256()) || !
base64PartChecksum.equals(base64PartChecksumFromObjectAttributes)) {
            throw new IOException("Part checksum didn't match for the
part");
        }
        System.out.println(partNumber + " " + base64PartChecksum);
    }
    //Before finalizing, do the final checksum validation.
    String base64FullObject =
Base64.getEncoder().encodeToString(sha256FullObject.digest());

```

```

        String base64ChecksumOfChecksums =
Base64.getEncoder().encodeToString(sha256ChecksumOfChecksums.digest());
        if (fullObjectChecksum != null && !
fullObjectChecksum.equals(base64FullObject)) {
            throw new IOException("Failed checksum validation for full
object");
        }
        System.out.println(fullObjectChecksum);
        String base64ChecksumOfChecksumFromAttributes =
objectAttributes.checksum().checksumSHA256();
        if (base64ChecksumOfChecksumFromAttributes != null && !
base64ChecksumOfChecksums.equals(base64ChecksumOfChecksumFromAttributes)) {
            throw new IOException("Failed checksum validation for full
object checksum of checksums");
        }
        System.out.println(base64ChecksumOfChecksumFromAttributes);
        out.flush();
    } catch (IOException | NoSuchAlgorithmException e) {
        //Cleanup bad file
        file.delete();
        e.printStackTrace();
    }
}

public static void validateExistingFileAgainstS3Checksum(S3Client s3Client)
{
    System.out.println("Starting existing file validation");
    File file = new File("DOWNLOADED_" + FILE_NAME);
    GetObjectAttributesResponse
        objectAttributes =
s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_N
        .objectAttributes(ObjectAttributes.OBJECT_PARTS,
ObjectAttributes.CHECKSUM).build());
    try (InputStream in = new FileInputStream(file)) {
        MessageDigest sha256ChecksumOfChecksums =
MessageDigest.getInstance("SHA-256");
        MessageDigest sha256Part = MessageDigest.getInstance("SHA-256");
        byte[] buffer = new byte[CHUNK_SIZE];
        int currentPart = 0;
        int partBreak =
objectAttributes.objectParts().parts().get(currentPart).size();
        int totalRead = 0;
        int read = in.read(buffer);
        while (read != -1) {

```

```

        totalRead += read;
        if (totalRead >= partBreak) {
            int difference = totalRead - partBreak;
            byte[] partChecksum;
            if (totalRead != partBreak) {
                sha256Part.update(buffer, 0, read - difference);
                partChecksum = sha256Part.digest();
                sha256ChecksumOfChecksums.update(partChecksum);
                sha256Part.reset();
                sha256Part.update(buffer, read - difference,
difference);
            } else {
                sha256Part.update(buffer, 0, read);
                partChecksum = sha256Part.digest();
                sha256ChecksumOfChecksums.update(partChecksum);
                sha256Part.reset();
            }
            String base64PartChecksum =
Base64.getEncoder().encodeToString(partChecksum);
            if (!
base64PartChecksum.equals(objectAttributes.objectParts().parts().get(currentPart).checksumSH
{
                throw new IOException("Part checksum didn't match S3");
            }
            currentPart++;
            System.out.println(currentPart + " " + base64PartChecksum);
            if (currentPart <
objectAttributes.objectParts().totalPartsCount()) {
                partBreak +=
objectAttributes.objectParts().parts().get(currentPart - 1).size();
            }
        } else {
            sha256Part.update(buffer, 0, read);
        }
        read = in.read(buffer);
    }
    if (currentPart != objectAttributes.objectParts().totalPartsCount())
{
        currentPart++;
        byte[] partChecksum = sha256Part.digest();
        sha256ChecksumOfChecksums.update(partChecksum);
        String base64PartChecksum =
Base64.getEncoder().encodeToString(partChecksum);
        System.out.println(currentPart + " " + base64PartChecksum);
    }
}

```

```
    }

    String base64CalculatedChecksumOfChecksums =
Base64.getEncoder().encodeToString(sha256ChecksumOfChecksums.digest());
    System.out.println(base64CalculatedChecksumOfChecksums);
    System.out.println(objectAttributes.checksum().checksumSHA256());
    if (!
base64CalculatedChecksumOfChecksums.equals(objectAttributes.checksum().checksumSHA256()))
    {
        throw new IOException("Full object checksum of checksums don't
match S3");
    }

} catch (IOException | NoSuchAlgorithmException e) {
    e.printStackTrace();
}
}
}
```

Uso da API REST

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](#).

Usando a 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.

Para obter informações, consulte [Amazon S3 CLI FAQ](#) no Guia do usuário da AWS Command Line Interface.

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 `PutObject`, `PostObject` ou `CopyObject`, 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 `PutObject`, `PostObject` ou `CopyObject`, 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](#).

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](#).

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”](#).

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 operação de API DELETE (`DeleteObject`) que pode ser usada para excluir um objeto em uma única solicitação HTTP.
- Excluir vários objetos: o Amazon S3 fornece a operação de API para exclusão de vários objetos (`DeleteObjects`) 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 com versionamento ativado, as operações de API de exclusão permitem 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](#).
- 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 operação de API de exclusão de vários objetos em um bucket habilitado para MFA, se alguma exclusão for uma solicitação de exclusão versionada (se você especificar uma chave de objeto e um 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 que especifica 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](#).

Tópicos

- [Excluir um único objeto](#)
- [Excluir vários objetos](#)

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](#).

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”](#).

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.

Warning

Quando você excluir permanentemente um objeto ou uma versão especificada do objeto no console do Amazon S3, a exclusão não poderá ser desfeita.

Como excluir um objeto com versionamento habilitado ou suspenso

Note

Se o ID da versão de um objeto em um bucket com versionamento suspenso estiver marcado como NULL, o S3 excluirá permanentemente o objeto, pois não existirão versões anteriores. No entanto, se um ID de versão válido estiver listado para o objeto em um bucket com versionamento suspenso, o S3 criará um marcador de exclusão para o objeto excluído, mantendo as versões anteriores do 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. Selecione o objeto e escolha Excluir.
4. Para confirmar a exclusão da lista de objetos em Objetos especificados, insira **delete** na caixa de texto Excluir objetos?.

Como excluir permanente uma versão específica do objeto em um bucket habilitado para versionamento


Warning

Quando você excluir permanentemente uma versão específica do objeto no Amazon S3, a exclusão não poderá ser desfeita.

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. Selecione o objeto que deseja excluir.
4. Escolha a opção Mostrar versões.
5. Selecione a versão do objeto e escolha Excluir.


6. Para confirmar a exclusão permanente das versões específicas do objeto em Objetos especificados, insira Excluir permanentemente na caixa de texto Excluir objetos?. O Amazon S3 excluirá permanentemente a versão específica do objeto.

Como excluir permanentemente um objeto de um bucket do Amazon S3 que não tenha o versionamento habilitado

 Warning

Quando você excluir permanentemente um objeto no Amazon S3, a exclusão não poderá ser desfeita. Além disso, as exclusões são permanentes para todos os buckets que não tenham o versionamento habilitado.

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. Selecione o objeto e escolha Excluir.
4. Para confirmar a exclusão permanente do objeto listado em Objetos especificados, insira Excluir permanentemente na caixa de texto Excluir objetos?.

 Note

Se você enfrentar problemas ao excluir um objeto, consulte [Quero excluir permanentemente objetos com versionamento](#).

Uso da SDKs AWS

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](#).

Java

Exemplo 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](#).

```
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](#).

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](#).

```
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](#).

```
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](#).

Este exemplo considera que você já está seguindo as instruções para [Usar o AWS SDK for PHP e executar exemplos do PHP](#) 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](#).

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](#).

```
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”](#).

Para obter informações sobre os recursos e a definição de preço do Amazon S3, consulte a [definição de preço do Amazon S3](#).

É possível usar o console do Amazon S3, AWS SDKs ou a API REST para excluir simultaneamente vários objetos 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.

Warning

- A exclusão de um objeto especificado não pode 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.
- Ao excluir objetos em um bucket sem o versionamento habilitado, o Amazon S3 exclui permanentemente os objetos.

- Ao excluir objetos em um bucket com o versionamento de buckets habilitado ou suspenso, o Amazon S3 cria marcadores de exclusão. Para obter mais informações, consulte [Trabalhar com marcadores de exclusão](#).

Como excluir objetos que têm versionamento habilitado ou suspenso

Note

Se os IDs da versão do objeto em um bucket com versionamento suspenso estiverem marcados como NULL, o S3 excluirá permanentemente os objetos, já que não há versões anteriores. No entanto, se um ID de versão válido estiver listado para os objetos em um bucket com versionamento suspenso, o S3 criará marcadores de exclusão para os objetos excluídos, mantendo as versões anteriores dos objetos.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Nome do bucket, escolha o nome do bucket do qual você deseja excluir objetos.
3. Selecione os objetos e escolha Excluir.
4. Para confirmar a exclusão da lista de objetos em Objetos especificados, insira **delete** na caixa de texto Excluir objetos?.

Como excluir permanente versões específicas de objetos em um bucket habilitado para versionamento


Warning

Quando você excluir permanentemente versões específicas de objetos no Amazon S3, a exclusão não poderá ser desfeita.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Nome do bucket, escolha o nome do bucket do qual você deseja excluir objetos.
3. Selecione os objetos do que deseja excluir.


4. Escolha a opção **Mostrar versões**.
5. Selecione as versões de objetos e escolha **Excluir**.
6. Para confirmar a exclusão permanente das versões específicas do objeto em **Objetos especificados**, insira **Excluir permanentemente** na caixa de texto **Excluir objetos?**. O Amazon S3 exclui permanentemente as versões específicas de objetos.

Como excluir permanentemente os objetos em um bucket do Amazon S3 que não tenham o versionamento habilitado

 **Warning**

Quando você excluir permanentemente um objeto no Amazon S3, a exclusão não poderá ser desfeita. Além disso, as exclusões são permanentes para todos os buckets que não tenham o versionamento habilitado.

1. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista **Nome do bucket**, escolha o nome do bucket do qual você deseja excluir objetos.
3. Selecione os objetos e escolha **Excluir**.
4. Para confirmar a exclusão permanente dos objetos listados em **Objetos especificados**, insira **excluir permanentemente** na caixa de texto **Excluir objetos?**.

 **Note**

Se você enfrentar problemas ao excluir os objetos, consulte [Quero excluir permanentemente objetos com versionamento](#).

Uso da SDKs AWS

Para ver exemplos de como excluir vários objetos com os AWS SDKs, consulte [Usar DeleteObjects com um AWS SDK ou ferramenta de linha de comando](#).

Para obter informações gerais sobre como usar AWS SDKs diferentes, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers](#).

Uso da API REST

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](#)
- [Listar chaves de objeto programaticamente](#)
- [Organizar objetos no console do Amazon S3 usando pastas](#)
- [Exibir uma visão geral do objeto no console do Amazon S3](#)
- [Exibir propriedades do objeto no console do Amazon S3](#)

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 um delimitador de prefixo muito comum. Em seguida, crie seus nomes de chave concatenando todos os níveis de contenção 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/'.

Para obter mais informações sobre delimitadores, prefixos e pastas aninhadas, consulte [Diferença entre prefixos e pastas aninhadas](#).

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>75cc57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</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](#).

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 delimite 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 cima pelo prefixo comum para fins de listagem. Isso permite que aplicativos organizem e naveguem por suas chaves hierarquicamente, de maneira muito similar a como você organizaria seus arquivos em diretórios em um sistema de arquivos.

Por exemplo, para estender o bucket de dicionário para conter mais do que apenas palavras inglesas, você poderia formar chaves prefixando cada palavra com seu idioma e um delimitador, tal como "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 prefixos](#).

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. AWS As bibliotecas wrapper do SDK fornecem a mesma paginação.

Exemplos

Os exemplos de código a seguir mostram como usar o `ListObjects`.

CLI

AWS CLI

O seguinte exemplo usa o comando `list-objects` para exibir os nomes de todos os objetos presentes no bucket especificado:

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size: Size}'
```

O exemplo usa o argumento `--query` para filtrar a saída de `list-objects` até encontrar o valor e o tamanho da chave de cada objeto

Para obter mais informações sobre objetos, consulte [Working with Amazon S3 Buckets](#) no Guia do desenvolvedor do Amazon S3.

- Para obter detalhes da API, consulte [ListObjects](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando recupera as informações sobre todos os itens no bucket “test-files”.

```
Get-S3Object -BucketName test-files
```

Exemplo 2: este comando recupera as informações sobre o item “sample.txt” do bucket “test-files”.

```
Get-S3Object -BucketName test-files -Key sample.txt
```

Exemplo 3: este comando recupera as informações sobre todos os itens com prefixo “sample” do bucket “test-files”.

```
Get-S3Object -BucketName test-files -KeyPrefix sample
```

- Consulte detalhes da API em [ListObjects](#) em AWS Tools for PowerShell Cmdlet Reference.

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. O console faz isso usando um prefixo de nome compartilhado para os objetos agrupados. Em outras palavras, os objetos agrupados têm nomes que começam com uma string comum. Essa string comum, ou prefixo compartilhado, é o nome da pasta. 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 cria uma pasta no Amazon S3, o 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 Command Line Interface (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](#)
- [Tornar as pastas públicas](#)
- [Calcular o tamanho da pasta](#)
- [Excluir pastas](#)

Criar uma pasta

Esta seção descreve como usar o console do Amazon S3 para criar uma pasta.

Important

Se a política de bucket não permitir o carregamento de objetos para esse bucket sem tags, metadados ou beneficiados da lista de controle de acesso (ACL), você não poderá criar uma pasta usando o procedimento a seguir. Em vez disso, carregue uma pasta vazia e especifique as configurações a seguir na configuração de carregamento.

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. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, selecione o nome do bucket no qual você deseja criar uma pasta.
4. 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).
5. Selecione Criar pasta.
6. 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 aos dados por prefixo. Para ter mais informações, consulte [Gerenciamento de identidade e acesso para o Amazon S3](#).

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 ter mais informações, consulte [Configurar ACLs](#).

Tópicos

- [Calcular o tamanho da pasta](#)
- [Excluir pastas](#)

Calcular o tamanho da pasta

Esta seção descreve como usar o console do Amazon S3 para calcular o tamanho de uma pasta.

Como calcular o tamanho de uma pasta

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 no qual a pasta está armazenada.
4. Na lista Objects (Objetos), marque a caixa de seleção ao lado do nome da pasta.
5. Escolha Actions (Ações), depois escolha Calculate total size (Calcular tamanho total).

Note

Ao sair da página, as informações da pasta (incluindo o tamanho total) não estarão mais disponíveis. É necessário calcular o tamanho total novamente se quiser vê-lo outra vez.

Important

- Quando você usa a ação Calculate total size (Calcular tamanho total) em objetos ou pastas especificados em seu bucket, o Amazon S3 calcula o número total de objetos e o tamanho total do armazenamento. No entanto, carregamentos fracionados incompletos ou em andamento e versões anteriores ou desatualizadas não são calculados no número total de objetos ou no tamanho total. Essa ação calcula somente o número total de objetos e o tamanho total da versão atual ou mais recente de cada objeto armazenado no bucket.

Por exemplo, se houver duas versões de um objeto em seu bucket, a calculadora de armazenamento no Amazon S3 as contará como apenas um objeto. Como resultado, o número total de objetos que é calculado no console do Amazon S3 pode ser diferente da métrica Object Count (Contagem de objetos) mostrada no S3 Storage Lens e do número informado pela métrica do Amazon CloudWatch, `NumberOfObjects`. Da mesma forma, o tamanho total do armazenamento também pode ser diferente da métrica Total Storage (Armazenamento total) mostrada no S3 Storage Lens e da métrica `BucketSizeBytes` mostrada no CloudWatch.

- Se o tempo para calcular o tamanho total de uma pasta grande estiver demorando muito, considere a possibilidade de usar o Amazon S3 Inventory e o Amazon S3 Select como alternativa. Primeiro, crie uma configuração do S3 Inventory para incluir os metadados de Tamanho de cada objeto da pasta grande em um relatório de inventário. O primeiro relatório do S3 Inventory pode levar até 48 horas para ser entregue. Quando o relatório de inventário for publicado, consulte o relatório de inventário com uma expressão SUM do S3 Select para agregar os tamanhos dos objetos na pasta. Para obter mais informações, consulte [Configurar um inventário usando o console do S3](#) e [Exemplo de SUM](#).

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 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. O console fornece todas as informações essenciais de um objeto em um só lugar.

Como abrir a página de detalhes 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 página de detalhes 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.

ℹ 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](#).

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 \(SSE-KMS\)](#) ou [Especificar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#).
- 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](#).
- d. Tags: você categoriza o armazenamento adicionando tags a um objeto do S3. Para ter mais informações, consulte [Categorizando seu armazenamento usando tags](#).
- e. Object lock legal hold and retention (Retenção legal e retenção do Bloqueio de objetos): você pode impedir que um objeto seja excluído. Para ter mais informações, consulte [Usar o bloqueio de objetos do S3](#).

Trabalhar com URLs pré-assinados

É possível usar URLs pré-assinados para conceder acesso por tempo limitado a objetos no Amazon S3 sem atualizar a política de bucket. Um URL pré-assinado pode ser inserido em um navegador ou usado por um programa para baixar um objeto. As credenciais usadas pelo URL pré-assinado são as do usuário da AWS que gerou o URL.

Também é possível usar URLs pré-assinados para permitir que alguém carregue um objeto específico para o bucket do Amazon S3. Isso permite um carregamento sem exigir que outra parte tenha credenciais ou permissões de segurança da AWS. Se um objeto com a mesma chave já existe no bucket como especificado no URL pré-assinado, o Amazon S3 substitui o objeto existente pelo objeto carregado.

É possível usar o pre-signed URL várias vezes, até a data e a hora de expiração.

Ao criar um URL pré-assinado, você deve fornecer suas credenciais de segurança e especificar o seguinte:

- Um bucket do Amazon S3
- Uma chave de objeto (se o download desse objeto for no bucket do Amazon S3, se estiver carregando, esse é o nome do arquivo a ser carregado)
- Um método HTTP (GET para baixar objetos ou PUT para carregar)
- Um intervalo de tempo de validade

No momento, os URLs pré-assinados do Amazon S3 não comportam o uso dos algoritmos de soma de verificação de integridade de dados (CRC32, CRC32C, SHA-1, SHA-256) a seguir quando você carrega objetos. Para verificar a integridade do objeto após o carregamento, forneça um resumo MD5 do objeto ao carregá-lo com um URL pré-assinado. Para ter mais informações sobre integridade de objeto, consulte [Verificar a integridade do objeto](#).

Tópicos

- [Quem pode criar um URL pré-assinado](#)
- [Tempo de validade para URLs pré-assinados](#)
- [Limitar recursos de pre-signed URLs](#)
- [Compartilhar objetos com URLs pré-assinados](#)
- [Carregar objetos com URLs pré-assinados](#)

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.

Veja a seguir os tipos de credencial que podem ser usados para criar um URL pré-assinado:

- Perfil de instância do IAM: válido por até 6 horas.
- AWS Security Token Service: válido até no máximo 36 horas quando assinado com credenciais de segurança de longo prazo ou com a duração da credencial temporária, o que terminar primeiro.
- 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 sete dias, primeiro delegue credenciais de usuário do IAM (a chave de acesso e a chave secreta) ao método usado para criar o URL pré-assinado.

Note

Se você criou um URL pré-assinado usando uma credencial temporária, o URL expirará quando a credencial expirar. Isso se aplica mesmo que o URL tenha sido criado com um prazo de validade posterior. Para receber informações sobre a permanência de credenciais de segurança temporárias, consulte [Comparação das operações de API do AWS STS](#) no Guia do usuário do IAM.

Tempo de validade para URLs pré-assinados

Um URL pré-assinado permanece válido pelo período de tempo especificado quando o URL é gerado. Se você criar um URL pré-assinado com o console do Amazon S3, o tempo de validade pode ser definido entre 1 minuto e 12 horas. Se você usar a AWS CLI ou AWS SDKs, o tempo de validade poderá ser definido como até sete dias.

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. Para obter mais informações sobre como as credenciais que você usa afetam o tempo de validade, consulte [Quem pode criar 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 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.

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. Veja a seguir alguns métodos que podem ser usados para restringir o uso de URLs pré-assinados.

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, a política de bucket a seguir usará a condição `s3:signatureAge` para negar qualquer solicitação de URL pré-assinado do Amazon S3 em objetos no bucket `DOC-EXAMPLE-BUCKET1` 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 min
old",
      "Effect": "Deny",
      "Principal": {"AWS": "*"},
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*",
      "Condition": {
        "NumericGreaterThan": {
          "s3:signatureAge": 600000
        }
      }
    }
  ]
}
```

Para obter mais informações sobre as chaves de política relacionadas ao AWS Signature versão 4, consulte [Autenticação do AWS Signature versão 4](#) na Referência de API do Amazon Simple Storage Service.

Restrição de caminho de rede

Se quiser restringir o uso de URLs pré-assinados e todo o acesso do Amazon S3 a caminhos de rede específicos, você poderá escrever políticas do AWS Identity and Access Management (IAM). 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 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 Amazon S3, use `aws:SourceIp`. Se você estiver usando um endpoint de nuvem privada virtual (VPC) 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, exige-se que qualquer acesso tenha origem nesse intervalo. Isso inclui o caso de alguém que esteja usando um URL pré-assinado para o Amazon S3. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
{
  "Sid": "NetworkRestrictionForIAMPrincipal",
  "Effect": "Deny",
  "Action": "*",
  "Resource": "*",
  "Condition": {
    "NotIpAddressIfExists": {"aws:SourceIp": "IP-address-range"},
    "BoolIfExists": {"aws:ViaAWSService": "false"}
  }
}
```

Para ver mais um exemplo de políticas de bucket que usam a chave de condição global `aws:SourceIp` da AWS para restringir o acesso a um bucket do Amazon S3 a um intervalo específico de rede, consulte [Gerenciar o acesso com base em endereços IP específicos](#).

Compartilhar objetos com URLs pré-assinados

Por padrão, todos os objetos do Amazon S3 são privados, somente o proprietário dos objetos tem permissão para acessá-los. No entanto, o proprietário dos objetos pode compartilhá-los com outras pessoas criando um URL pré-assinado. Um URL pré-assinado usa credenciais de segurança para conceder permissão por tempo limitado para baixar objetos. O URL pode ser inserido em um navegador ou usado por um programa para baixar o objeto. As credenciais usadas pelo URL pré-assinado são as do usuário da AWS que gerou o URL.

Para obter informações gerais sobre URLs pré-assinados, consulte [Trabalhar com URLs pré-assinados](#).

É possível criar um URL pré-assinado para compartilhar um objeto sem escrever nenhum código usando o console do Amazon S3, o AWS Explorer for Visual Studio (Windows) ou AWS Toolkit for Visual Studio Code. Também é possível gerar um URL pré-assinado programaticamente usando a AWS Command Line Interface (AWS CLI) ou AWS SDKs.

Uso do console do S3

É possível usar o console do Amazon S3 para gerar um URL pré-assinado para compartilhar um objeto seguindo estas etapas. Ao usar o console, 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 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. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, selecione o nome do bucket que contém os objetos para os quais você deseja obter um URL pré-assinado.
4. Na lista Objects (Objetos), selecione o objeto para o qual deseja criar um URL pré-assinado.
5. No menu Ações de objeto, escolha Compartilhar com um URL pré-assinado.
6. Especifique por quanto tempo deseja que o URL pré-assinado seja válido.
7. Escolha Create presigned URL (Criar URL pré-assinado).
8. 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.

Usando a AWS CLI

O exemplo do comando da AWS CLI a seguir gera um URL pré-assinado para compartilhar um objeto de um bucket do Amazon S3. Ao usar a AWS CLI, o tempo máximo de validade de um URL pré-assinado é de sete dias a partir do momento da criação. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
aws s3 presign s3://DOC-EXAMPLE-BUCKET1/mydoc.txt --expires-in 604800
```

Note

Para todas as Regiões da AWS lançadas após 20 de março de 2019, você precisa especificar o `endpoint-url` e a Região da AWS com a solicitação. Para obter uma lista de todos os endpoints e as regiões do Amazon S3, consulte [Regiões e endpoints](#) na Referência geral da AWS.

```
aws s3 presign s3://DOC-EXAMPLE-BUCKET1/mydoc.txt --expires-in 604800 --region af-south-1 --endpoint-url https://s3.af-south-1.amazonaws.com
```

Para obter mais informações, consulte [presign](#) na Referência de comandos da AWS CLI.

Uso dos AWS SDKs

Para obter exemplos de uso de AWS SDKs para gerar um URL pré-assinado para compartilhar um objeto, consulte [Criar um URL pré-assinado para o Amazon S3 usando um AWS SDK](#).

Ao usar os AWS SDKs para gerar um URL pré-assinado, o tempo máximo de expiração é de sete dias a partir do momento da criação.

Note

Para todas as Regiões da AWS lançadas após 20 de março de 2019, você precisa especificar o `endpoint-url` e a Região da AWS com a solicitação. Para obter uma lista de todos os endpoints e as regiões do Amazon S3, consulte [Regiões e endpoints](#) na Referência geral da AWS.

Note

Ao usar AWS SDKs, o atributo de marcação deve ser um cabeçalho e não um parâmetro de consulta. Todos os outros atributos podem ser transmitidos como um parâmetro para o URL pré-assinado.

Usar o AWS Toolkit for Visual Studio (Windows)

Note

No momento, o AWS Toolkit for Visual Studio não é compatível com o Visual Studio para Mac.

1. Instale o AWS Toolkit for Visual Studio usando as instruções a seguir, [Installing and setting up the Toolkit for Visual Studio](#) no Guia do usuário do AWS Toolkit for Visual Studio.
2. Conecte-se à AWS usando as etapas a seguir, [Conectar-se ao AWS](#) no Guia do usuário do AWS Toolkit for Visual Studio.
3. No painel do lado esquerdo chamado AWS Explorer, clique duas vezes no bucket que contém seu objeto.
4. Clique com o botão direito do mouse no objeto para o qual você deseja gerar um URL pré-assinado e selecione Criar URL pré-assinado....
5. Na janela pop-up, defina a data e hora de validade para o URL pré-assinado.
6. A Chave de objeto deve ser preenchida previamente com base no objeto selecionado.
7. Escolha GET para especificar que esse URL pré-assinado será usado para baixar um objeto.
8. Escolha o botão Generate (Gerar).
9. Para copiar o URL para a área de transferência, escolha Copy (Copiar).
10. Para usar o URL pré-assinado gerado, cole o URL em qualquer navegador.

Usar o AWS Toolkit for Visual Studio Code

Se estiver usando o Visual Studio Code, você poderá gerar um URL pré-assinado para compartilhar um objeto sem escrever nenhum código usando o AWS Toolkit for Visual Studio Code. Para obter informações gerais, consulte [AWS Toolkit for Visual Studio Code](#) no Guia do usuário do AWS Toolkit for Visual Studio Code.

Para obter instruções sobre como instalar o AWS Toolkit for Visual Studio Code, consulte [Instalar o AWS Toolkit for Visual Studio Code](#) no Guia do usuário do AWS Toolkit for Visual Studio Code.

1. Conecte-se à AWS usando as etapas a seguir, [Conectar-se ao AWS Toolkit for Visual Studio Code](#) no Guia do usuário do AWS Toolkit for Visual Studio Code.
2. Selecione o logotipo da AWS no painel esquerdo do Visual Studio Code.

3. Em EXPLORER, selecione S3.
4. Escolha um bucket e um arquivo e abra o menu de contexto (clique com o botão direito do mouse).
5. Escolha Gerar URL pré-assinado e defina o tempo de validade (em minutos).
6. Pressione Enter e o URL pré-assinado será copiado para a área de transferência.

Carregar objetos com URLs pré-assinados

Você pode usar URLs pré-assinados para permitir que alguém carregue um objeto no seu bucket do Amazon S3. O uso de um URL pré-assinado permitirá um carregamento sem exigir que outra pessoa tenha credenciais ou permissões de segurança da AWS. Um URL pré-assinado é limitado pelas permissões do usuário que o criou. Isto é, se você receber um URL pré-assinado para carregar um objeto, só poderá fazer isso se o criador do URL pré-assinado tiver as permissões necessárias para carregar esse objeto.

Quando alguém usa o URL para carregar 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. Após o carregamento, o proprietário do bucket será o proprietário do objeto.

Para obter informações gerais sobre URLs pré-assinados, consulte [Trabalhar com URLs pré-assinados](#).

É possível criar um URL pré-assinado para carregar um objeto sem escrever nenhum código usando o AWS Explorer for Visual Studio. Também é possível gerar um URL pré-assinado programaticamente usando AWS SDKs.

Usar o AWS Toolkit for Visual Studio (Windows)

Note

No momento, o AWS Toolkit for Visual Studio não é compatível com o Visual Studio para Mac.

1. Instale o AWS Toolkit for Visual Studio usando as instruções a seguir, [Installing and setting up the Toolkit for Visual Studio](#) no Guia do usuário do AWS Toolkit for Visual Studio.

2. Conecte-se à AWS usando as etapas a seguir, [Conectar-se ao AWS](#) no Guia do usuário do AWS Toolkit for Visual Studio.
3. No painel do lado esquerdo chamado AWS Explorer, clique com o botão direito do mouse no bucket para o qual você deseja que um objeto seja carregado.
4. Escolha Criar URL pré-assinado....
5. Na janela pop-up, defina a data e hora de validade para o URL pré-assinado.
6. Em Chave de objeto, defina o nome do arquivo a ser carregado. O arquivo que você está carregando deve corresponder exatamente a esse nome. Se já existir um objeto com a mesma chave de objeto no bucket, o Amazon S3 substituirá o objeto existente pelo objeto recém-carregado.
7. Escolha PUT para especificar que esse URL pré-assinado será usado para carregar um objeto.
8. Escolha o botão Generate (Gerar).
9. Para copiar o URL para a área de transferência, escolha Copy (Copiar).
10. Para usar esse URL, você pode enviar uma solicitação PUT com o comando `curl`. Inclua o caminho completo para o arquivo e o próprio URL pré-assinado.

```
curl -X PUT -T "/path/to/file" "presigned URL"
```

Uso dos AWS SDKs

Para obter exemplos de como usar AWS SDKs para gerar um URL pré-assinado para o carregamento de um objeto, consulte [Criar um URL pré-assinado para o Amazon S3](#) usando um AWS SDK.

Ao usar os AWS SDKs para gerar um URL pré-assinado, o tempo máximo de expiração é de sete dias a partir do momento da criação.

Note

Para todas as Regiões da AWS lançadas após 20 de março de 2019, você precisa especificar o `endpoint-url` e a Região da AWS com a solicitação. Para obter uma lista de todos os endpoints e as regiões do Amazon S3, consulte [Regiões e endpoints](#) na Referência geral da AWS.

Transformar objetos com o S3 Object Lambda

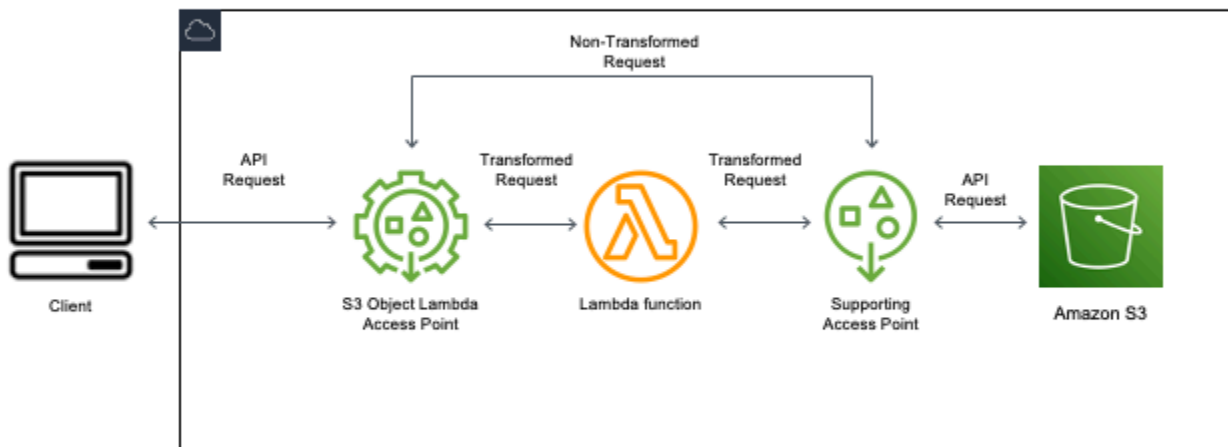
Com o Amazon S3 Object Lambda, você pode adicionar seu próprio código às solicitações GET, LIST e HEAD do Amazon S3 para modificar e processar dados, conforme eles são retornados para uma aplicação. Você pode usar o código personalizado para modificar os dados retornados por solicitações GET do S3 para filtrar linhas, redimensionar imagens e marcas d'água dinamicamente, editar dados confidenciais e muito mais. Você também pode usar o S3 Object Lambda para modificar a saída das solicitações LIST do S3 para criar uma visualização personalizada de todos os objetos em um bucket e as solicitações HEAD do S3 para modificar os metadados do objeto, como nome e tamanho do objeto. Você pode usar o S3 Object Lambda como origem para sua distribuição do Amazon CloudFront a fim de personalizar dados para usuários finais, como redimensionar imagens automaticamente, transcodificar formatos mais antigos (como de JPEG para WebP) ou remover metadados. Para obter mais informações, consulte a publicação do blog da AWS [Usar o Amazon S3 Object Lambda com o Amazon CloudFront](#). Alimentado pelas funções do AWS Lambda, seu código é executado em uma infraestrutura totalmente gerenciada pela AWS. O uso do S3 Object Lambda reduz a necessidade de criar e armazenar cópias derivadas de seus dados ou de executar proxies, tudo sem a necessidade de alterar suas aplicações.

Como o S3 Object Lambda funciona

O S3 Object Lambda usa funções do AWS Lambda para processar automaticamente a saída de solicitações GET, LIST e HEAD 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.

Depois de configurar uma função do Lambda, anexe-a a um endpoint de serviço do S3 Object Lambda, conhecido como ponto de acesso do Object Lambda. O ponto de acesso do Object Lambda usa um ponto de acesso padrão do S3, conhecido como ponto de acesso de suporte, para acessar o Amazon S3.

Quando você envia uma solicitação para seu ponto de acesso do Object Lambda, o Amazon S3 chama automaticamente a função do Lambda. Qualquer dado recuperado usando uma solicitação GET, LIST ou HEAD do S3 por meio do ponto de acesso do Object Lambda retorna um resultado transformado de volta à aplicação. Todas as outras solicitações são processadas como normais, conforme ilustrado no diagrama a seguir.



Os tópicos nesta seção descrevem como trabalhar com o S3 Object Lambda.

Tópicos

- [Criar pontos de acesso do Object Lambda](#)
- [Usar pontos de acesso do Amazon S3 Object Lambda](#)
- [Considerações sobre segurança para pontos de acesso do S3 Object Lambda](#)
- [Escrever funções do Lambda para pontos de acesso do S3 Object Lambda](#)
- [Usar funções do Lambda criadas pela AWS](#)
- [Práticas recomendadas e diretrizes para o S3 Object Lambda](#)
- [Tutoriais do S3 Object Lambda](#)
- [Depuração do S3 Object Lambda](#)

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:

- Um bucket do Amazon S3. Para obter mais informações sobre a criação de buckets, consulte [the section called “Criação de um bucket”](#).
- Um ponto de acesso do S3 padrão. Quando você trabalha com pontos de acesso do Object Lambda, esse ponto de acesso padrão é conhecido como ponto de acesso de suporte. Para obter informações sobre a criação de pontos de acesso padrão, consulte [the section called “Criar pontos de acesso”](#).
- Uma função do AWS Lambda. Você pode criar sua própria função do Lambda ou usar uma função pré-criada. Para obter mais informações sobre como criar funções do Lambda, consulte [the section called “Escrever funções do Lambda”](#). Para obter mais informações sobre funções pré-construídas, consulte [Usar funções do Lambda criadas pela AWS](#).
- (Opcional) Uma política do AWS Identity and Access Management (IAM). Os pontos de acesso do Amazon S3 são compatíveis com políticas de recursos do IAM que você pode usar para controlar o uso do ponto de acesso por recurso, usuário ou outras condições. Para obter mais informações sobre como criar essas políticas, consulte [the section called “Configurar políticas do IAM”](#).

As seções a seguir descrevem como criar um ponto de acesso do Object Lambda usando:

- A AWS Management Console
- A AWS Command Line Interface (AWS CLI)
- Um modelo do AWS CloudFormation
- A AWS Cloud Development Kit (AWS CDK)

Para obter informações sobre como criar um ponto de acesso do Object Lambda usando a API REST, consulte [CreateAccessPointForObjectLambda](#) na Referência de API do Amazon Simple Storage Service.

Criar um ponto de acesso do Object Lambda

Use um dos procedimentos a seguir para criar o ponto de acesso do Object Lambda.

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. Na barra de navegação, escolha o nome da Região da AWS exibida no momento. Em seguida, escolha a região para a qual você deseja mudar.
3. No painel de navegação à esquerda, escolha Object Lambda access points (Pontos de acesso do Object Lambda).
4. 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).
5. 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 de pontos de acesso do Object Lambda. Para ter mais informações, consulte [Regras para nomear pontos de acesso do Amazon S3](#).

6. 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. Para obter informações sobre a criação de pontos de acesso padrão, consulte [the section called “Criar pontos de acesso”](#).
7. Em Configuração de transformação, você pode adicionar uma função que transforma os dados para o ponto de acesso do Object Lambda. Execute um destes procedimentos:
 - Se você já tem uma função do AWS Lambda em sua conta, pode escolhê-la em Invoke Lambda function (Invocar função do Lambda). Aqui você pode inserir o nome do recurso da Amazon (ARN) de uma função do Lambda em sua Conta da AWS ou escolher uma função do Lambda no menu suspenso.
 - Se você quiser usar uma função integrada à AWS, escolha o nome da função em Função criada pela AWS e selecione Criar função do Lambda. Isso levará você ao console do Lambda, onde poderá implantar uma função integrada à sua Conta da AWS. Para obter mais informações sobre funções integradas, consulte [Usar funções do Lambda criadas pela AWS](#).

Em S3 APIs (APIs do S3), escolha uma ou mais operações de API para invocar. Para cada API selecionada, você deve especificar uma função do Lambda para invocar.

8. (Opcional) Em Payload (Carga útil), adicione o texto JSON que você quer fornecer para a 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 função do Lambda.

⚠ Important

Ao usar os pontos de acesso do Object Lambda, verifique se a carga útil não contém informações sigilosas.

9. (Opcional) Em Range and part number (Intervalo e número de peça), você deve habilitar essa opção a fim processar solicitações GET e HEAD com cabeçalhos de intervalo e número de peça. Habilitar 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](#).
10. (Opcional) Em Métricas de solicitação, escolha Ativar ou Desativar para adicionar o monitoramento do Amazon S3 ao ponto de acesso do Object Lambda. As métricas de solicitação são cobradas na taxa padrão do Amazon CloudWatch.
11. (Opcional) Em Object Lambda Access Point policy (Política de ponto de acesso do Object Lambda), defina uma política de recursos. As políticas de recursos concedem permissões para o ponto de acesso do Object Lambda especificado e podem controlar o uso do ponto de acesso por recurso, usuário ou outras condições. Para obter mais informações sobre as políticas de recursos de ponto de acesso do Object Lambda, consulte [Configurar políticas do IAM para pontos de acesso do Object Lambda](#).
12. Em Block Public Access settings for this Object Lambda Access Point (Configurações de bloqueio de acesso público para este ponto de acesso do Object Lambda), selecione as configurações de bloqueio de acesso público que você deseja aplicar. Todas as configurações de bloqueio de acesso público são habilitadas por padrão para novos pontos de acesso do Object Lambda, e recomendamos que você deixe as configurações padrão habilitadas. Atualmente, o Amazon S3 não oferece suporte à alteração das configurações de bloqueio de acesso público a pontos de acesso do Object Lambda após a criação de pontos de acesso do Object Lambda.

Para obter mais informações sobre como usar o bloqueio de acesso público do Amazon S3, consulte [Gerenciar o acesso público a pontos de acesso](#).
13. Escolha Create Object Lambda Access Point (Criar ponto de acesso do Object Lambda).

Usando a AWS CLI

Para criar um ponto de acesso do Object Lambda usando um modelo do AWS CloudFormation

Note

Para usar os comandos a seguir, substitua *user input placeholders* por suas próprias informações.

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. Execute o comando `put-object` a seguir para fazer upload do pacote em um bucket do Amazon S3.

```
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. Execute o comando `deploy` a seguir para implantar o modelo 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
```

Você pode configurar esse modelo do AWS CloudFormation para invocar o Lambda para as operações de API GET, HEAD e LIST. Para obter mais informações sobre como modificar a configuração padrão do modelo, consulte [the section called “Automatizar a configuração do S3 Object Lambda com AWS CloudFormation”](#).

Para criar um ponto de acesso do Object Lambda usando a AWS CLI

Note

Para usar os comandos a seguir, substitua *user input placeholders* por suas próprias informações.

O exemplo a seguir cria um ponto de acesso do Object Lambda chamado *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”](#).

Este exemplo usa a função pré-criada *decompress* da AWS. Para obter mais informações sobre funções pré-construídas, consulte [the section called “Usar funções criadas pela AWS”](#).

1. Criará 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”](#).
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”](#).
3. Execute um destes procedimentos:
 - Crie na conta uma função do Lambda que você deseja usar para transformar o objeto do Simple Storage Service (Amazon S3). Para obter mais informações sobre como criar funções do Lambda, consulte [the section called “Escrever funções do Lambda”](#). Para usar sua função personalizada com a AWS CLI, consulte [Como usar o Lambda com a AWS CLI](#) no Guia do desenvolvedor do AWS Lambda.
 - Use uma função do Lambda pré-criada pela AWS. Para obter mais informações sobre funções pré-construídas, consulte [Usar funções do Lambda criadas pela AWS](#).
4. Crie um arquivo de configuração JSON chamado *my-olap-configuration.json*. Nessa configuração, forneça o ponto de acesso de suporte e o nome do recurso da Amazon (ARN) para a função do Lambda que você criou nas etapas anteriores ou o ARN da função pré-criada que você está usando.

Example


```
{
  "SupportingAccessPoint" : "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
  "TransformationConfigurations": [{
    "Actions" : ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
    "ContentTransformation" : {
      "AwsLambda": {
        "FunctionPayload" : "{\"compressionType\":\"gzip\"}",
        "FunctionArn" : "arn:aws:lambda:us-east-1:111122223333:function/
compress"
      }
    }
  }]
}
```

5. Execute o comando `create-access-point-for-object-lambda` para criar o 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
```

6. (Opcional) Crie um arquivo de política JSON chamado `my-olap-policy.json`.

Adicionar uma política de recursos de ponto de acesso do Object Lambda pode controlar o uso do ponto de acesso por recurso, usuário ou outras condições. Essa política de recursos concede a permissão `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"
    }
  ]
}
```

```

    }
  ]
}

```

7. (Opcional) Execute o comando `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

```

8. (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 função do Lambda.

A configuração do ponto de acesso do Object Lambda a seguir mostra uma carga útil com dois parâmetros.

```

{
  "SupportingAccessPoint": "AccessPointArn",
  "CloudWatchMetricsEnabled": false,
  "TransformationConfigurations": [{
    "Actions": ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
    "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`, `GetObject-PartNumber`, `HeadObject-Range` e `HeadObject-PartNumber` ativados.

```

{
  "SupportingAccessPoint": "AccessPointArn",
  "CloudWatchMetricsEnabled": false,

```

```
"AllowedFeatures": ["GetObject-Range", "GetObject-PartNumber", "HeadObject-Range", "HeadObject-PartNumber"],
"TransformationConfigurations": [{
  "Action": ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
  "ContentTransformation": {
    "AwsLambda": {
      "FunctionArn": "FunctionArn",
      "FunctionPayload": "{\"compression-amount\": \"5\"}"
    }
  }
}]
}
```

Important

Ao usar os pontos de acesso do Object Lambda, verifique se a carga útil não contém 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 modelo do AWS CloudFormation e o código-fonte da função do Lambda do [repositório do GitHub](#) e implantar esses recursos para configurar um ponto de acesso funcional do Object Lambda.

Para obter informações sobre como modificar a configuração padrão do template do AWS CloudFormation, consulte [the section called “Automatizar a configuração do S3 Object Lambda com AWS CloudFormation”](#).


Para obter informações sobre como configurar pontos de acesso do Object Lambda usando o AWS CloudFormation sem o modelo, consulte [AWS::S3ObjectLambda::AccessPoint](#) no Guia do usuário do AWS CloudFormation.

Para carregar o pacote de implantação da função Lambda

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. Faça upload do pacote em um bucket do Amazon S3.


Para criar um ponto de acesso do Object Lambda usando o console do AWS CloudFormation

1. Baixe o modelo `s3objectlambda_defaultconfig.yaml` do AWS CloudFormation 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. Execute um destes procedimentos:
 - Se você nunca usou o AWS CloudFormation, na página inicial do AWS CloudFormation, escolha Create stack (Criar pilha).
 - Se você já usou o AWS CloudFormation, no painel de navegação à esquerda, selecione Stacks (Pilhas). Selecione Create stack (Criar pilha), depois escolha With new resources (standard) (Com novos recursos (padrão)).
4. Em Prerequisite - Prepare template (Pré-requisito - Preparar template), escolha Template is ready (O template está pronto).
5. Em Specify template (Especificar modelo), escolha Upload a template file (Fazer upload de um arquivo de modelo) e faça upload de `s3objectlambda_defaultconfig.yaml`.
6. Escolha Próximo.
7. Na página Specify stack details (Especificar detalhes da pilha), insira um nome para a pilha.
8. Na seção Parameters (Parâmetros), especifique os parâmetros que são definidos no modelo da pilha:
 - a. Em CreateNewSupportingAccessPoint, siga um destes procedimentos:
 - Se você já tem um ponto de acesso de suporte para o bucket do S3 em que fez upload do modelo, escolha false (falso).
 - Se quiser criar um ponto de acesso para esse bucket, selecione true (verdadeiro).
 - b. Para EnableCloudWatchMonitoring, escolha true (verdadeiro) ou false (falso), dependendo se deseja habilitar as métricas e os alarmes de solicitação do Amazon CloudWatch.
 - c. (Opcional) Em LambdaFunctionPayload, adicione o texto JSON que você quer fornecer para a 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 função do Lambda.

 Important

Ao usar os pontos de acesso do Object Lambda, verifique se a carga útil não contém informações sigilosas.

- d. Em `LambdaFunctionRuntime`, insira seu tempo de execução preferido para a função do Lambda. As opções disponíveis são `nodejs14.x`, `python3.9` e `java11`.
- e. Em `LambdaFunctionS3BucketName`, insira o nome do bucket do Amazon S3 para o qual você fez upload do pacote de implantação.
- f. Em `LambdaFunctionS3Key`, insira o nome da chave de objeto do Amazon S3 para o qual você fez upload do pacote de implantação.
- g. Em `LambdaFunctionS3Key`, insira a versão do objeto do Amazon S3 no qual você fez upload do pacote de implantação.
- h. Em `ObjectLambdaAccessPointName`, insira um nome para o ponto de acesso do Object Lambda.
- i. Em `S3BucketName`, insira o nome do bucket do Amazon S3 que será associado ao ponto de acesso do Object Lambda.
- j. Em `SupportingAccessPointName`, insira o nome de seu ponto de acesso de suporte.

 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 modelo para criar um selecionando `true` (verdadeiro) em `CreateNewSupportingAccessPoint`.

- 9. Escolha Próximo.
- 10. Na página Configurar opções de pilha, selecione Avançar.

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.

- 11. Na página Review (Revisão), escolha Create stack (Criar pilha).

Usando a AWS Cloud Development Kit (AWS CDK)

Para obter mais informações sobre como configurar pontos de acesso do Object Lambda usando o AWS CDK, consulte [Biblioteca de estruturas do AWS::S3ObjectLambda](#) na Referência de API do AWS Cloud Development Kit (AWS CDK).

Automatizar a configuração do S3 Object Lambda com um modelo do CloudFormation

Você pode usar um modelo do AWS CloudFormation para criar rapidamente um ponto de acesso do Amazon S3 Object Lambda. O modelo do CloudFormation cria automaticamente recursos relevantes, configura perfis do AWS Identity and Access Management (IAM) e configura uma função do AWS Lambda que lida automaticamente com solicitações por meio do ponto de acesso do Object Lambda. Com o modelo do CloudFormation, é possível implementar práticas recomendadas, melhorar seu procedimento de segurança e reduzir erros causados por processos manuais.

Esse [repositório GitHub](#) contém o modelo do CloudFormation e o código-fonte da função do Lambda. Para obter instruções sobre como usar o modelo, consulte [the section called “Criar pontos de acesso do Object Lambda”](#).

A função do Lambda fornecida no modelo não executa nenhuma transformação. Em vez disso, ele retorna seus objetos na condição em que estão do bucket do 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. Para obter mais informações sobre como modificar a função, consulte [the section called “Modificar a função Lambda”](#) e [the section called “Escrever funções do Lambda”](#).

Modificar o template

Criar um ponto de acesso de suporte

O S3 Object Lambda usa dois pontos de acesso: um ponto de acesso do Object Lambda e um ponto de acesso padrão do S3, chamado de 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. É possível criar um ponto de acesso de suporte transmitindo o seguinte parâmetro como parte do comando `aws cloudformation deploy` ao implantar o modelo.

```
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=json"
```

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
```

Esse parâmetro habilita o ponto de acesso do Object Lambda para métricas de solicitação do Amazon S3 e cria dois alarmes do CloudWatch para monitorar erros no lado do cliente e no 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 de pontos de acesso](#).

Para obter detalhes sobre os preços, consulte [Definição de preço do CloudWatch](#).

Configurar a simultaneidade provisionada

Para reduzir a latência, é possível configurar a simultaneidade provisionada para a função do Lambda com base no ponto de acesso do Object Lambda editando o modelo 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 sobre simultaneidade provisionada, consulte [Gerenciar a simultaneidade provisionada do Lambda](#) no Guia do desenvolvedor do AWS Lambda. Para obter detalhes de preço, consulte [Definição de preço do AWS Lambda](#).

Modificar a função Lambda

Alterar os valores do cabeçalho de uma solicitação **GetObject**

Por padrão, a função do 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 do Lambda para enviar novos valores de cabeçalho, transmitindo-os na operação de 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 mostrado 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 do Lambda para enviar novos valores de cabeçalho, transmitindo-os na solicitação de operação de 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 solicitação de operação de 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 os parâmetros **Range** e **partNumber** no objeto de origem

Por padrão, o ponto de acesso do Object Lambda criado pelo modelo do CloudFormation pode lidar com os parâmetros Range e partNumber. A função Lambda aplica o intervalo ou o número da parte solicitado ao objeto transformado. Para isso, a função precisa baixar o objeto inteiro 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. Essa abordagem reduz a latência geral da função e a memória necessárias. Para ter mais informações, consulte [the section called “Trabalhar com cabeçalhos Range e partNumber”](#).

Desabilitar o tratamento de **Range** e **partNumber**

Por padrão, o ponto de acesso do Object Lambda criado pelo modelo do CloudFormation pode lidar com os parâmetros Range e partNumber. Se você não precisar desse comportamento, poderá desabilitá-lo removendo as seguintes linhas do modelo:

```
AllowedFeatures:  
- GetObject-Range  
- GetObject-PartNumber  
- HeadObject-Range  
- HeadObject-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 do Lambda. Para obter um exemplo de implementação, consulte o [Exemplo de conteúdo compactado de transmissão](#).

Usar pontos de acesso do Amazon S3 Object Lambda

Fazer solicitações via Pontos de Acesso do Amazon 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](#). Você pode fazer solicitações via pontos de acesso do Object Lambda usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI), AWS SDKs ou API REST do Amazon S3.

Important

Os nomes dos recursos da Amazon (ARNs) para pontos de acesso do Object Lambda usam um nome de serviço de `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 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 console ou a AWS CLI do Amazon S3.

Uso do console do S3

Para encontrar o ARN do seu 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 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 você deseja copiar.

4. Escolha Copy ARN (Copiar ARN).

Usando a AWS CLI

Para encontrar o ARN do ponto de acesso do Object Lambda usando a 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"
    },
    ...
  ]
}
```


Como usar um alias no estilo do bucket para o ponto de acesso do Object Lambda do bucket do S3

Quando você cria um ponto de acesso do Object Lambda, o Amazon S3 gera automaticamente um alias exclusivo para seu ponto de acesso do Object Lambda. É possível usar esse alias em vez de um nome de bucket do Amazon S3 ou o nome do recurso da Amazon (ARN) do ponto de acesso do Object Lambda em uma solicitação para operações de plano de dados do ponto de acesso. Para obter uma lista dessas operações, consulte [Compatibilidade com ponto de acesso com produtos da AWS](#).

Um nome de alias do ponto de acesso do Object Lambda é criado dentro do mesmo namespace de um bucket do Amazon S3. Esse nome de alias é gerado automaticamente, e não é possível alterá-lo. Para um ponto de acesso do Object Lambda existente, um alias é automaticamente atribuído para

uso. Um nome de alias de ponto de acesso do Object Lambda atende a todos os requisitos de um nome de bucket válido do Amazon S3 e consiste nas seguintes partes:

Object Lambda Access Point name prefix-metadata--o1-s3

 Note

O sufixo `--o1-s3` é reservado para nomes de alias de ponto de acesso e não pode ser usado para nomes de bucket ou de ponto de acesso do Object Lambda. Para obter mais informações sobre as regras para nomes de bucket do Amazon S3, consulte [Regras de nomeação de bucket](#).

Os exemplos a seguir mostram o ARN e o alias de ponto de acesso do Object Lambda para um ponto de acesso do Object Lambda chamado *my-object-lambda-access-point*:

- ARN: `arn:aws:s3-object-lambda:region:account-id:accesspoint/my-object-lambda-access-point`
- Alias de ponto de acesso do Object Lambda: `my-object-lambda-acc-1a4n8yjrb3kda96f67zwrwiuse1a--o1-s3`

Ao usar um ponto de acesso do Object Lambda, você pode usar o nome do alias do ponto de acesso do Object Lambda sem exigir grandes alterações no código.

Quando você exclui um ponto de acesso do Object Lambda, o nome do alias do ponto de acesso do Object Lambda se torna inativo e não provisionado.

Como encontrar o alias do ponto de acesso do Object Lambda

Uso do console do S3

Para encontrar o alias do 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 à esquerda, escolha Object Lambda access points (Pontos de acesso do Object Lambda).
3. Para o ponto de acesso do Object Lambda que você deseja usar, copie o valor do alias do ponto de acesso do Object Lambda.

Usando a AWS CLI

Quando você cria um ponto de acesso do Object Lambda, o Amazon S3 gera automaticamente um nome de alias de ponto de acesso do Object Lambda, conforme mostrado no exemplo de comando a seguir. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações. Para obter informações sobre como criar um ponto de acesso do Object Lambda usando a AWS CLI, consulte [Para criar um ponto de acesso do Object Lambda usando a AWS CLI](#).

```
aws s3control create-access-point-for-object-lambda --account-id 111122223333 --
name my-object-lambda-access-point --configuration file://my-olap-configuration.json
{
  "ObjectLambdaAccessPointArn": "arn:aws:s3:region:111122223333:accesspoint/my-
access-point",
  "Alias": {
    "Value": "my-object-lambda-acc-1a4n8yjr3kda96f67zwrwiiuse1a--ol-s3",
    "Status": "READY"
  }
}
```

O nome do alias do ponto de acesso do Object Lambda gerado tem dois campos:

- O campo `Value` é o valor do alias do ponto de acesso do Object Lambda.
- O campo `Status` é o status do alias do ponto de acesso do Object Lambda. Se o status for `PROVISIONING`, o Amazon S3 estará provisionando o alias do ponto de acesso do Object Lambda, e o alias ainda não estará pronto para uso. Se o status for `READY`, o alias do ponto de acesso do Object Lambda terá sido provisionado com êxito e estará pronto para uso.

Para obter mais informações sobre o tipo de dados `ObjectLambdaAccessPointAlias` na API REST, consulte [CreateAccessPointForObjectLambda](#) e [ObjectLambdaAccessPointAlias](#) na Referência da API do Amazon Simple Storage Service.

Como usar o alias do ponto de acesso do Object Lambda

É possível usar um alias de ponto de acesso do Object Lambda em vez de um nome de bucket do Amazon S3 para as operações listadas em [Compatibilidade com ponto de acesso com produtos da AWS](#).

O exemplo da AWS CLI a seguir para o comando `get-bucket-location` usa o alias do ponto de acesso do bucket para retornar a Região da AWS em que o bucket está. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3api get-bucket-location --bucket my-object-lambda-acc-w7i37nq6xuzgax3jw3oqtifiusw2a--ol-s3
```

```
{  
  "LocationConstraint": "us-west-2"  
}
```

Se o alias do ponto de acesso do Object Lambda em uma solicitação não for válido, o código de erro `InvalidAccessPointAliasError` será retornado. Para obter mais informações sobre `InvalidAccessPointAliasError`, consulte [Lista de códigos de erro](#) na Referência de API do Amazon Simple Storage Service.

As limitações de um alias de um ponto de acesso do Object Lambda são as mesmas de um alias de ponto de acesso. Para obter mais informações sobre os limites de um alias do ponto de acesso, consulte [Limitações](#).

Considerações sobre segurança para pontos de acesso do S3 Object Lambda

Com o Amazon S3 Object Lambda, você pode realizar transformações personalizadas em dados à medida que deixa o S3 usando a escala e a flexibilidade do AWS Lambda como uma plataforma de computação. O S3 e o Lambda permanecem seguros por padrão, mas, para manter a segurança, é necessária uma consideração especial do autor da função do Lambda. O S3 Object Lambda exige que todo o acesso seja feito por entidades autenticadas (sem acesso anônimo) e por HTTPS.

Para mitigar os riscos de segurança, recomendamos o seguinte:

- Defina o escopo do perfil de execução do Lambda com o menor conjunto de permissões possível.
- Sempre que possível, verifique se a função do Lambda acessa o Amazon S3 por meio do URL pré-assinado fornecido.

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 [Configurar políticas do IAM para pontos de acesso do Object Lambda](#).

Comportamento de criptografia

Como os pontos de acesso do Object Lambda usam o Amazon S3 e o AWS 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](#).

- Quando você usa a criptografia do lado do servidor do S3 com pontos de acesso do Object Lambda, o objeto é descriptografado antes de ser enviado para o Lambda. Depois que o objeto é enviado para o Lambda, ele é processado sem criptografia (no caso de uma solicitação GET ou HEAD).
- Para evitar que a chave de criptografia seja registrada em log, o S3 rejeitará solicitações GET e HEAD para objetos criptografados usando criptografia no servidor com chaves fornecidas pelo cliente (SSE-C). No entanto, 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 de criptografia 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 do Object Lambda e um ponto de acesso padrão do S3, chamado de 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 uma solicitação, o S3 gera um URL pré-assinado para seu objeto em seu nome por meio do ponto de acesso de suporte. Sua função do Lambda receberá esse URL como entrada quando a função for chamada.

É possível definir sua função do Lambda para usar esse URL pré-assinado ao recuperar o objeto original, em vez de chamar o S3 diretamente. Usando esse modelo, você pode 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. Essa abordagem também protege suas funções do Lambda de estarem sujeitas ao [problema Confused Deputy](#), no qual uma função mal configurada com permissões diferentes das permissões de seu chamador 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 por um ponto de acesso do Object Lambda, é necessário ter a permissão `lambda:InvokeFunction` para a função do Lambda configurada. Da mesma forma, ao chamar outras operações de API 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 com erros HTTP 403 (Proibido). Todo o acesso deve ser feito por entidades autenticadas. Se precisar de acesso público, use 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.

Endereços IP de ponto de acesso do Object Lambda

As sub-redes de `describe-managed-prefix-lists` são compatíveis com endpoints de nuvem privada virtual (VPC) do gateway e são relacionadas à tabela de roteamento de endpoints da VPC. Como o ponto de acesso do Object Lambda não é compatível com a VPC do gateway, seus intervalos de IP estão ausentes. Os intervalos ausentes pertencem ao Amazon S3, mas não são compatíveis com endpoints da VPC de gateway. Para obter mais informações sobre `describe-managed-prefix-lists`, consulte [DescribeManagedPrefixLists](#) na Referência da API do Amazon EC2 e [Intervalos de endereços IP da AWS](#) na Referência geral da AWS.

Configurar políticas do IAM para pontos de acesso do Object Lambda

Os pontos de acesso do 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. Você pode controlar o acesso por meio de uma política de recursos opcional no ponto de acesso do Object Lambda ou uma política de recursos no ponto de acesso de suporte. Para obter um exemplo detalhado, consulte [Tutorial: Como transformar dados para sua aplicação com o S3 Object Lambda](#) e [Tutorial: Detectar e editar dados PII com o S3 Object Lambda e o Amazon Comprehend](#).

Os quatro recursos a seguir devem ter permissões concedidas para trabalhar com pontos de acesso do Object Lambda:

- A identidade do IAM, como um usuário ou um perfil. Para obter mais informações sobre as identidades do IAM e as práticas recomendadas, consulte [Identidades do IAM \(usuários, grupos de usuários e perfis\)](#) no Guia do usuário do IAM.

- O bucket e o ponto de acesso padrão associado. Quando você trabalha com pontos de acesso do Object Lambda, esse ponto de acesso padrão é conhecido como ponto de acesso de suporte.
- O ponto de acesso do Object Lambda.
- A função do AWS Lambda.

Important

Antes de salvar a política, solucione os avisos de segurança, os erros, os avisos gerais e as sugestões de AWS Identity and Access Management Access Analyzer. 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 do 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](#).

Os seguintes exemplos de política 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
```

- 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 em sua conta, você deverá incluir a versão da função na declaração da política. No seguinte exemplo de ARN, a versão é indicada por **1**:
`arn:aws:lambda:us-`

`east-1:111122223333:function:MyObjectLambdaFunction:1`

O Lambda não permite adicionar políticas do IAM à versão \$LATEST. Para obter mais informações sobre as versões de funções do Lambda, consulte [Versões de funções do Lambda](#) no Guia do desenvolvedor do AWS Lambda.

Example : Política de buckets que delega controle de acesso a pontos de acesso

O exemplo de política de buckets do S3 a seguir delega o controle de acesso para um bucket aos pontos de acesso padrão 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 de um ponto de acesso, o que significa que as operações só podem ser invocadas por meio dos pontos de acesso. Para ter mais informações, consulte [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" }
      }
    }
  ]
}
```

Example : política do IAM que concede ao usuário as permissões necessárias para usar um ponto de acesso do Object Lambda

A seguinte política do IAM concede a um usuário permissões para a função do Lambda, o ponto de acesso padrão e o ponto de acesso do Object Lambda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowLambdaInvocation",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:lambda:us-
east-1:111122223333:function:MyObjectLambdaFunction:1",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": [
            "s3-object-lambda.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "AllowStandardAccessPointAccess",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:us-east-1:111122223333:accesspoint/my-access-point/*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": [
            "s3-object-lambda.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "AllowObjectLambdaAccess",
```

```
"Action": [
  "s3-object-lambda:Get*",
  "s3-object-lambda:List*"
],
"Effect": "Allow",
"Resource": "arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/my-
object-lambda-ap"
}
]
}
```

Habilitar permissões para funções de execução do Lambda

Quando são feitas solicitações GET a um ponto de acesso do Object Lambda, a função do Lambda precisa de permissão para enviar dados ao ponto de acesso do S3 Object Lambda. Isso é fornecido habilitando a permissão `s3-object-lambda:WriteGetObjectResponse` na função de execução da sua função do Lambda. Crie uma nova função de execução ou atualize uma existente.

Note

Sua função precisa da permissão `s3-object-lambda:WriteGetObjectResponse` somente se você estiver fazendo uma solicitação GET.

Como criar uma função de execução no console do IAM

1. Abra o console IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação à esquerda, escolha Roles.
3. Selecione Create role (Criar função).
4. Em Common use cases (Casos de uso comuns), selecione Lambda.
5. Escolha Próximo.
6. Na página Add permissions (Adicionar permissões), pesquise a política [AmazonS3ObjectLambdaExecutionRolePolicy](#) gerenciada pela AWS e marque a caixa de seleção ao lado do nome da política.

Essa política deve conter a ação `s3-object-lambda:WriteGetObjectResponse`.

7. Escolha Próximo.

8. Na página Name, review, and create (Nomear, revisar e criar), para Role name (Nome da função), insira **s3-object-lambda-role**.
9. (Opcional) Adicione uma descrição e etiquetas para esse perfil.
10. Selecione Criar função.
11. Aplique a **s3-object-lambda-role** recém-criada como a função de execução de sua função Lambda. Isso pode ser feito durante ou após a criação da função do Lambda no console do Lambda.

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

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).

Suporte ao CORS de ponto de acesso do Object Lambda

Ao receber uma solicitação de um navegador ou quando a solicitação inclui um cabeçalho Origin, o S3 Object Lambda sempre adiciona um campo de cabeçalho "AllowedOrigins": "*".

Para ter mais informações, consulte [Usar o compartilhamento de recursos de origem cruzada \(CORS\)](#).

Escrever funções do Lambda para pontos de acesso do S3 Object Lambda

Essa seção apresenta detalhes de como escrever funções do AWS Lambda para usar com pontos de acesso do Amazon S3 Object Lambda.

Para saber mais sobre procedimentos completos para algumas tarefas do S3 Object Lambda, consulte o seguinte:

- [Tutorial: Como transformar dados para sua aplicação com o S3 Object Lambda](#)
- [Tutorial: Detectar e editar dados PII com o S3 Object Lambda e o Amazon Comprehend](#)
- [Tutorial: Using S3 Object Lambda to dynamically watermark images as they are retrieved](#) (Tutorial: Como usar o S3 Object Lambda para colocar marca d'água em imagens de maneira dinâmica à medida que são recuperadas)

Tópicos

- [Trabalhar com solicitações GetObject no Lambda](#)
- [Trabalhar com solicitações HeadObject no Lambda](#)
- [Trabalhar com solicitações ListObjects no Lambda](#)
- [Trabalhar com solicitações ListObjectsV2 no Lambda](#)
- [Formato e uso de contexto de evento](#)
- [Trabalhar com cabeçalhos Range e partNumber](#)

Trabalhar com solicitações **GetObject** no Lambda

Esta seção pressupõe que seu ponto de acesso do Object Lambda está configurado para chamar a função do Lambda para `GetObject`. O S3 Object Lambda inclui a operação de API do Amazon S3, `WriteGetObjectResponse`, que permite que a função do Lambda apresente dados personalizados e cabeçalhos de resposta para o chamador de `GetObject`.

`WriteGetObjectResponse` oferece controle extensivo sobre código de status, cabeçalhos de resposta e corpo de resposta, com base nas 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 código de status HTTP 403 (Proibido)
- Exemplo 2: responda com uma imagem transformada
- Exemplo 3: Transmitir conteúdo compactado

Exemplo 1: Responder com código de status HTTP 403 (Proibido)

Você pode usar `WriteGetObjectResponse` para responder com o código de status HTTP 403 (Proibido) com base no conteúdo do objeto.

Java

```
package com.amazon.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
```

```
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import java.io.ByteArrayInputStream;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example1 {

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
    Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();

        // Check to see if the request contains all of the necessary information.
        // If it does not, send a 4XX response and a custom error code and message.
        // Otherwise, retrieve the object from S3 and stream it
        // to the client unchanged.
        var tokenIsNotPresent = !
event.getUserRequest().getHeaders().containsKey("requiredToken");
        if (tokenIsNotPresent) {
            s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
                .withRequestRoute(event.outputRoute())
                .withRequestToken(event.outputToken())
                .withStatusCode(403)
                .withContentLength(0L).withInputStream(new
ByteArrayInputStream(new byte[0]))
                .withErrorCode("MissingRequiredToken")
                .withErrorMessage("The required token was not present in the
request."));
            return;
        }

        // Prepare the presigned URL for use and make the request to S3.
        HttpClient httpClient = HttpClient.newBuilder().build();
        var presignedResponse = httpClient.send(
            HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
            HttpResponse.BodyHandlers.ofInputStream());

        // Stream the original bytes back to the caller.
        s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
            .withRequestRoute(event.outputRoute()))
```



```
        .withRequestToken(event.outputToken())
        .withInputStream(presignedResponse.body()));
    }
}
```

Python

```
import boto3
import requests

def handler(event, context):
    s3 = boto3.client('s3')

    """
    Retrieve the operation context object from the event. This object indicates
    where the WriteGetObjectResponse request
    should be delivered and contains a presigned URL in 'inputS3Url' where we can
    download the requested object from.
    The 'userRequest' object has information related to the user who made this
    'GetObject' request to
    S3 Object Lambda.
    """
    get_context = event["getObjectContext"]
    user_request_headers = event["userRequest"]["headers"]

    route = get_context["outputRoute"]
    token = get_context["outputToken"]
    s3_url = get_context["inputS3Url"]

    # Check for the presence of a 'CustomHeader' header and deny or allow based on
    that header.
    is_token_present = "SuperSecretToken" in user_request_headers

    if is_token_present:
        # If the user presented our custom 'SuperSecretToken' header, we send the
        requested object back to the user.
        response = requests.get(s3_url)
        s3.write_get_object_response(RequestRoute=route, RequestToken=token,
        Body=response.content)
    else:
        # If the token is not present, we send an error back to the user.
        s3.write_get_object_response(RequestRoute=route, RequestToken=token,
        StatusCode=403,
```

```
        ErrorCode="NoSuperSecretTokenFound", ErrorMessage="The request was not  
secret enough.")
```

```
# Gracefully exit the Lambda function.  
return { 'status_code': 200 }
```

Node.js

```
const { S3 } = require('aws-sdk');  
const axios = require('axios').default;  
  
exports.handler = async (event) => {  
    const s3 = new S3();  
  
    // Retrieve the operation context object from the event. This object indicates  
    // where the WriteGetObjectResponse request  
    // should be delivered and contains a presigned URL in 'inputS3Url' where we can  
    // download the requested object from.  
    // The 'userRequest' object has information related to the user who made this  
    // 'GetObject' request to S3 Object Lambda.  
    const { userRequest, getObjectContext } = event;  
    const { outputRoute, outputToken, inputS3Url } = getObjectContext;  
  
    // Check for the presence of a 'CustomHeader' header and deny or allow based on  
    // that header.  
    const isTokenPresent = Object  
        .keys(userRequest.headers)  
        .includes("SuperSecretToken");  
  
    if (!isTokenPresent) {  
        // If the token is not present, we send an error back to the user. The  
        // 'await' in front of the request  
        // indicates that we want to wait for this request to finish sending before  
        // moving on.  
        await s3.writeGetObjectResponse({  
            RequestRoute: outputRoute,  
            RequestToken: outputToken,  
            StatusCode: 403,  
            ErrorCode: "NoSuperSecretTokenFound",  
            ErrorMessage: "The request was not secret enough.",  
        }).promise();  
    } else {
```

```
    // If the user presented our custom 'SuperSecretToken' header, we send the
    requested object back to the user.
    // Again, note the presence of 'await'.
    const presignedResponse = await axios.get(inputS3Url);
    await s3.writeGetObjectResponse({
      RequestRoute: outputRoute,
      RequestToken: outputToken,
      Body: presignedResponse.data,
    }).promise();
  }

  // Gracefully exit the Lambda function.
  return { statusCode: 200 };
}
```

Exemplo 2:responda com uma imagem transformada

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.amazon.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import javax.imageio.ImageIO;
import java.awt.image.BufferedImage;
import java.awt.Image;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example2 {
```

```
private static final int HEIGHT = 250;
private static final int WIDTH = 250;

public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
Exception {
    AmazonS3 s3Client = AmazonS3Client.builder().build();
    HttpClient httpClient = HttpClient.newBuilder().build();

    // Prepare the presigned URL for use and make the request to S3.
    var presignedResponse = httpClient.send(
        HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
        HttpResponse.BodyHandlers.ofInputStream());

    // The entire image is loaded into memory here so that we can resize it.
    // Once the resizing is completed, we write the bytes into the body
    // of the WriteGetObjectResponse request.
    var originalImage = ImageIO.read(presignedResponse.body());
    var resizingImage = originalImage.getScaledInstance(WIDTH, HEIGHT,
Image.SCALE_DEFAULT);
    var resizedImage = new BufferedImage(WIDTH, HEIGHT,
BufferedImage.TYPE_INT_RGB);
    resizedImage.createGraphics().drawImage(resizingImage, 0, 0, WIDTH, HEIGHT,
null);

    var baos = new ByteArrayOutputStream();
    ImageIO.write(resizedImage, "png", baos);

    // Stream the bytes back to the caller.
    s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
        .withRequestRoute(event.outputRoute())
        .withRequestToken(event.outputToken())
        .withInputStream(new ByteArrayInputStream(baos.toByteArray())));
}
}
```

Python

```
import boto3
import requests
import io
from PIL import Image
```

```

def handler(event, context):
    """
    Retrieve the operation context object from the event. This object indicates
    where the WriteGetObjectResponse request
    should be delivered and has a presigned URL in 'inputS3Url' where we can
    download the requested object from.
    The 'userRequest' object has information related to the user who made this
    'GetObject' request to
    S3 Object Lambda.
    """
    get_context = event["getObjectContext"]
    route = get_context["outputRoute"]
    token = get_context["outputToken"]
    s3_url = get_context["inputS3Url"]

    """
    In this case, we're resizing .png images that are stored in S3 and are
    accessible through the presigned URL
    'inputS3Url'.
    """
    image_request = requests.get(s3_url)
    image = Image.open(io.BytesIO(image_request.content))
    image.thumbnail((256,256), Image.ANTIALIAS)

    transformed = io.BytesIO()
    image.save(transformed, "png")

    # Send the resized image back to the client.
    s3 = boto3.client('s3')
    s3.write_get_object_response(Body=transformed.getvalue(), RequestRoute=route,
    RequestToken=token)

    # Gracefully exit the Lambda function.
    return { 'status_code': 200 }

```

Node.js

```

const { S3 } = require('aws-sdk');
const axios = require('axios').default;
const sharp = require('sharp');

exports.handler = async (event) => {
    const s3 = new S3();

```

```
// Retrieve the operation context object from the event. This object indicates
where the WriteGetObjectResponse request
// should be delivered and has a presigned URL in 'inputS3Url' where we can
download the requested object from.
const { getObjectContext } = event;
const { outputRoute, outputToken, inputS3Url } = getObjectContext;

// In this case, we're resizing .png images that are stored in S3 and are
accessible through the presigned URL
// 'inputS3Url'.
const { data } = await axios.get(inputS3Url, { responseType: 'arraybuffer' });

// Resize the image.
const resized = await sharp(data)
  .resize({ width: 256, height: 256 })
  .toBuffer();

// Send the resized image back to the client.
await s3.writeGetObjectResponse({
  RequestRoute: outputRoute,
  RequestToken: outputToken,
  Body: resized,
}).promise();

// Gracefully exit the Lambda function.
return { statusCode: 200 };
}
```

Exemplo 3: Transmitir conteúdo compactado

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.amazon.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
```

```
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example3 {

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
    Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();
        HttpClient httpClient = HttpClient.newBuilder().build();

        // Request the original object from S3.
        var presignedResponse = httpClient.send(
            HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
            HttpResponse.BodyHandlers.ofInputStream());

        // Consume the incoming response body from the presigned request,
        // apply our transformation on that data, and emit the transformed bytes
        // into the body of the WriteGetObjectResponse request as soon as they're
        ready.
        // This example compresses the data from S3, but any processing pertinent
        // to your application can be performed here.
        var bodyStream = new GZIPCompressingInputStream(presignedResponse.body());

        // Stream the bytes back to the caller.
        s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
            .withRequestRoute(event.outputRoute())
            .withRequestToken(event.outputToken())
            .withInputStream(bodyStream));
    }
}
```

Python

```
import boto3
import requests
```

```
import zlib
from botocore.config import Config

"""
A helper class to work with content iterators. Takes an iterator and compresses the
bytes that come from it. It
implements 'read' and '__iter__' so that the SDK can stream the response.
"""
class Compress:
    def __init__(self, content_iter):
        self.content = content_iter
        self.compressed_obj = zlib.compressobj()

    def read(self, _size):
        for data in self.__iter__():
            return data

    def __iter__(self):
        while True:
            data = next(self.content)
            chunk = self.compressed_obj.compress(data)
            if not chunk:
                break

            yield chunk

        yield self.compressed_obj.flush()

def handler(event, context):
    """
    Setting the 'payload_signing_enabled' property to False allows us to send a
    streamed response back to the client.
    in this scenario, a streamed response means that the bytes are not buffered into
    memory as we're compressing them,
    but instead are sent straight to the user.
    """
    my_config = Config(
        region_name='eu-west-1',
        signature_version='s3v4',
        s3={
            "payload_signing_enabled": False
        }
    )
```



```

)
s3 = boto3.client('s3', config=my_config)

"""
Retrieve the operation context object from the event. This object indicates
where the WriteGetObjectResponse request
should be delivered and has a presigned URL in 'inputS3Url' where we can
download the requested object from.
The 'userRequest' object has information related to the user who made this
'GetObject' request to S3 Object Lambda.
"""
get_context = event["getObjectContext"]
route = get_context["outputRoute"]
token = get_context["outputToken"]
s3_url = get_context["inputS3Url"]

# Compress the 'get' request stream.
with requests.get(s3_url, stream=True) as r:
    compressed = Compress(r.iter_content())

# Send the stream back to the client.
s3.write_get_object_response(Body=compressed, RequestRoute=route,
RequestToken=token, ContentType="text/plain",
                             ContentEncoding="gzip")

# Gracefully exit the Lambda function.
return {'status_code': 200}

```

Node.js

```

const { S3 } = require('aws-sdk');
const axios = require('axios').default;
const zlib = require('zlib');

exports.handler = async (event) => {
    const s3 = new S3();

    // Retrieve the operation context object from the event. This object indicates
    where the WriteGetObjectResponse request
    // should be delivered and has a presigned URL in 'inputS3Url' where we can
    download the requested object from.
    const { getObjectContext } = event;
    const { outputRoute, outputToken, inputS3Url } = getObjectContext;

```

```
// Download the object from S3 and process it as a stream, because it might be a
// huge object and we don't want to
// buffer it in memory. Note the use of 'await' because we want to wait for
// 'writeGetObjectResponse' to finish
// before we can exit the Lambda function.
await axios({
  method: 'GET',
  url: inputS3Url,
  responseType: 'stream',
}).then(
  // Gzip the stream.
  response => response.data.pipe(zlib.createGzip())
).then(
  // Finally send the gzip-ed stream back to the client.
  stream => s3.writeGetObjectResponse({
    RequestRoute: outputRoute,
    RequestToken: outputToken,
    Body: stream,
    ContentType: "text/plain",
    ContentEncoding: "gzip",
  }).promise()
);

// Gracefully exit the Lambda function.
return { statusCode: 200 };
}
```

Note

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 código de status HTTP 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 (Erro interno do servidor). 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 com código de status HTTP 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 a operação de API `WriteGetObjectResponse`, o Amazon S3 exige a rota e o token de solicitação do contexto do evento. Para ter mais informações, consulte [Formato e uso de contexto de evento](#).

Os parâmetros de token de roteamento e solicitação são necessários para conectar a resposta `WriteGetObjectResult` com o autor da chamada original. Ainda que sempre seja apropriado repetir respostas 500 (Erro interno do servidor), como o token de solicitação é um token de uso único, as tentativas subsequentes de usá-lo podem resultar em respostas com código de status HTTP 400 (Solicitação inválida). Embora a chamada para `WriteGetObjectResponse` com tokens de roteamento 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.

Trabalhar com solicitações **HeadObject** no Lambda

Esta seção pressupõe que seu ponto de acesso do Object Lambda está configurado para chamar a função do Lambda para `HeadObject`. O Lambda receberá uma carga útil JSON que contém uma chave chamada `headObjectContext`. Dentro do contexto, há uma única propriedade chamada `inputS3Url`, que é um URL pré-assinado para o ponto de acesso de suporte para `HeadObject`.

O URL pré-assinado incluirá as seguintes propriedades se elas forem especificadas:

- `versionId` (nos parâmetros de consulta)
- `requestPayer` (no cabeçalho de `x-amz-request-payer`)
- `expectedBucketOwner` (no cabeçalho de `x-amz-expected-bucket-owner`)

Outras propriedades não serão pré-assinadas e, portanto, não serão incluídas. As opções não assinadas enviadas como cabeçalhos podem ser adicionadas manualmente à solicitação ao chamar o URL pré-assinado encontrado nos cabeçalhos de `userRequest`. As opções de criptografia do lado do servidor não são compatíveis com `HeadObject`.

Para ver os parâmetros de URI da sintaxe da solicitação, consulte [HeadObject](#) na Amazon Simple Storage Service API Reference (Referência da API do Amazon Simple Storage Service).

O exemplo a seguir mostra uma carga útil JSON de entrada do Lambda para HeadObject.

```
{
  "xAmzRequestId": "requestId",
  "**headObjectContext**": {
    "**inputS3Url**": "https://my-s3-ap-111122223333.s3-accesspoint.us-east-1.amazonaws.com/example?X-Amz-Security-Token=<snip>"
  },
  "configuration": {
    "accessPointArn": "arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/example-object-lambda-ap",
    "supportingAccessPointArn": "arn:aws:s3:us-east-1:111122223333:accesspoint/example-ap",
    "payload": "{}"
  },
  "userRequest": {
    "url": "https://object-lambda-111122223333.s3-object-lambda.us-east-1.amazonaws.com/example",
    "headers": {
      "Host": "object-lambda-111122223333.s3-object-lambda.us-east-1.amazonaws.com",
      "Accept-Encoding": "identity",
      "X-Amz-Content-SHA256": "e3b0c44298fc1example"
    }
  },
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "principalId",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
    "accountId": "111122223333",
    "accessKeyId": "accessKeyId",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
      }
    },
    "sessionIssuer": {
      "type": "Role",
      "principalId": "principalId",
      "arn": "arn:aws:iam::111122223333:role/Admin",
      "accountId": "111122223333",

```

```
        "userName": "Admin"
      }
    },
    "protocolVersion": "1.00"
  }
```

Sua função do Lambda deve retornar um objeto JSON que contém os cabeçalhos e os valores que serão retornados para a chamada de `HeadObject`.

O exemplo a seguir mostra a estrutura do JSON de resposta do Lambda para `HeadObject`.

```
{
  "statusCode": <number>; // Required
  "errorCode": <string>;
  "errorMessage": <string>;
  "headers": {
    "Accept-Ranges": <string>,
    "x-amz-archive-status": <string>,
    "x-amz-server-side-encryption-bucket-key-enabled": <boolean>,
    "Cache-Control": <string>,
    "Content-Disposition": <string>,
    "Content-Encoding": <string>,
    "Content-Language": <string>,
    "Content-Length": <number>, // Required
    "Content-Type": <string>,
    "x-amz-delete-marker": <boolean>,
    "ETag": <string>,
    "Expires": <string>,
    "x-amz-expiration": <string>,
    "Last-Modified": <string>,
    "x-amz-missing-meta": <number>,
    "x-amz-object-lock-mode": <string>,
    "x-amz-object-lock-legal-hold": <string>,
    "x-amz-object-lock-retain-until-date": <string>,
    "x-amz-mp-parts-count": <number>,
    "x-amz-replication-status": <string>,
    "x-amz-request-charged": <string>,
    "x-amz-restore": <string>,
    "x-amz-server-side-encryption": <string>,
    "x-amz-server-side-encryption-customer-algorithm": <string>,
    "x-amz-server-side-encryption-aws-kms-key-id": <string>,
    "x-amz-server-side-encryption-customer-key-MD5": <string>,
  }
}
```

```
    "x-amz-storage-class": <string>,
    "x-amz-tagging-count": <number>,
    "x-amz-version-id": <string>,
    <x-amz-meta-headers>: <string>, // user-defined metadata
    "x-amz-meta-meta1": <string>, // example of the user-defined metadata header,
it will need the x-amz-meta prefix
    "x-amz-meta-meta2": <string>
    ...
};
}
```

O exemplo a seguir mostra como usar o URL pré-assinado para preencher sua resposta modificando os valores do cabeçalho conforme necessário antes de retornar o JSON.

Python

```
import requests

def lambda_handler(event, context):
    print(event)

    # Extract the presigned URL from the input.
    s3_url = event["headObjectContext"]["inputS3Url"]

    # Get the head of the object from S3.
    response = requests.head(s3_url)

    # Return the error to S3 Object Lambda (if applicable).
    if (response.status_code >= 400):
        return {
            "statusCode": response.status_code,
            "errorCode": "RequestFailure",
            "errorMessage": "Request to S3 failed"
        }

    # Store the headers in a dictionary.
    response_headers = dict(response.headers)

    # This obscures Content-Type in a transformation, it is optional to add
    response_headers["Content-Type"] = ""

    # Return the headers to S3 Object Lambda.
    return {
```

```
"statusCode": response.status_code,  
"headers": response_headers  
}
```

Trabalhar com solicitações **ListObjects** no Lambda

Esta seção pressupõe que seu ponto de acesso do Object Lambda está configurado para chamar a função do Lambda para `ListObjects`. O Lambda receberá a carga JSON com um novo objeto chamado `listObjectContext`. `listObjectContext` contém uma única propriedade, `inputS3Url`, que é um URL pré-assinado para o ponto de acesso de suporte para `ListObjects`.

Ao contrário de `GetObject` e `HeadObject`, o URL pré-assinado incluirá as seguintes propriedades se elas forem especificadas:

- Todos os parâmetros de consulta
- `requestPayer` (no cabeçalho de `x-amz-request-payer`)
- `expectedBucketOwner` (no cabeçalho de `x-amz-expected-bucket-owner`)

Para ver os parâmetros de URI da sintaxe da solicitação, consulte [ListObjects](#) na Referência da API do Amazon Simple Storage Service.

Important

Recomendamos que você use a versão mais recente, [ListObjectsV2](#), ao desenvolver aplicações. Para compatibilidade com versões anteriores, o Amazon S3 continua a oferecer suporte a `ListObjects`.

O exemplo a seguir mostra a carga útil JSON de entrada do Lambda para `ListObjects`.

```
{  
  "xAmzRequestId": "requestId",  
  "**listObjectContext**": {  
    "**inputS3Url**": "https://my-s3-ap-111122223333.s3-accesspoint.us-east-1.amazonaws.com/?X-Amz-Security-Token=<snip>",  
  },  
  "configuration": {  
    "accessPointArn": "arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/example-object-lambda-ap",
```

```

    "supportingAccessPointArn": "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
    "payload": "{}"
  },
  "userRequest": {
    "url": "https://object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com/example",
    "headers": {
      "Host": "object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com",
      "Accept-Encoding": "identity",
      "X-Amz-Content-SHA256": "e3b0c44298fc1example"
    }
  },
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "principalId",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
    "accountId": "111122223333",
    "accessKeyId": "accessKeyId",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "principalId",
        "arn": "arn:aws:iam::111122223333:role/Admin",
        "accountId": "111122223333",
        "userName": "Admin"
      }
    }
  },
  "protocolVersion": "1.00"
}

```

Sua função do Lambda deve retornar um objeto JSON que contém o código de status, o resultado XML da lista ou as informações de erro que serão retornadas do S3 Object Lambda.

O S3 Object Lambda não processa nem valida `listResultXml`. Em vez disso, o encaminha para o chamador de `ListObjects`. Para `listBucketResult`, o S3 Object Lambda espera que

certas propriedades sejam de um tipo específico e lançará exceções se não puder analisá-las. `listResultXml` e `listBucketResult` não podem ser fornecidas ao mesmo tempo.

O exemplo a seguir demonstra como usar o URL pré-assinado para chamar o Amazon S3 e usar o resultado para preencher uma resposta, incluindo a verificação de erros.

Python

```
import requests
import xmltodict

def lambda_handler(event, context):
    # Extract the presigned URL from the input.
    s3_url = event["listObjectsContext"]["inputS3Url"]

    # Get the head of the object from Amazon S3.
    response = requests.get(s3_url)

    # Return the error to S3 Object Lambda (if applicable).
    if (response.status_code >= 400):
        error = xmltodict.parse(response.content)
        return {
            "statusCode": response.status_code,
            "errorCode": error["Error"]["Code"],
            "errorMessage": error["Error"]["Message"]
        }

    # Store the XML result in a dict.
    response_dict = xmltodict.parse(response.content)

    # This obscures StorageClass in a transformation, it is optional to add
    for item in response_dict['ListBucketResult']['Contents']:
        item['StorageClass'] = ""

    # Convert back to XML.
    listResultXml = xmltodict.unparse(response_dict)

    # Create response with listResultXml.
    response_with_list_result_xml = {
        'statusCode': 200,
        'listResultXml': listResultXml
    }
```

```

# Create response with listBucketResult.
response_dict['ListBucketResult'] =
sanitize_response_dict(response_dict['ListBucketResult'])
response_with_list_bucket_result = {
    'statusCode': 200,
    'listBucketResult': response_dict['ListBucketResult']
}

# Return the list to S3 Object Lambda.
# Can return response_with_list_result_xml or response_with_list_bucket_result
return response_with_list_result_xml

# Converting the response_dict's key to correct casing
def sanitize_response_dict(response_dict: dict):
    new_response_dict = dict()
    for key, value in response_dict.items():
        new_key = key[0].lower() + key[1:] if key != "ID" else 'id'
        if type(value) == list:
            newlist = []
            for element in value:
                if type(element) == type(dict()):
                    element = sanitize_response_dict(element)
                newlist.append(element)
            value = newlist
        elif type(value) == dict:
            value = sanitize_response_dict(value)
        new_response_dict[new_key] = value
    return new_response_dict

```

O exemplo a seguir mostra a estrutura do JSON de resposta do Lambda para ListObjects.

```

{
  "statusCode": <number>; // Required
  "errorCode": <string>;
  "errorMessage": <string>;
  "listResultXml": <string>; // This can also be Error XML string in case S3 returned
  error response when calling the pre-signed URL

  "listBucketResult": { // listBucketResult can be provided instead of listResultXml,
  however they can not both be provided in the JSON response
    "name": <string>, // Required for 'listBucketResult'

```

```

    "prefix": <string>,
    "marker": <string>,
    "nextMarker": <string>,
    "maxKeys": <int>, // Required for 'listBucketResult'
    "delimiter": <string>,
    "encodingType": <string>
    "isTruncated": <boolean>, // Required for 'listBucketResult'
    "contents": [ {
      "key": <string>, // Required for 'content'
      "lastModified": <string>,
      "eTag": <string>,
      "checksumAlgorithm": <string>, // CRC32, CRC32C, SHA1, SHA256
      "size": <int>, // Required for 'content'
      "owner": {
        "displayName": <string>, // Required for 'owner'
        "id": <string>, // Required for 'owner'
      },
      "storageClass": <string>
    },
    ...
  ],
  "commonPrefixes": [ {
    "prefix": <string> // Required for 'commonPrefix'
  },
  ...
  ],
}
}

```

Trabalhar com solicitações **ListObjectsV2** no Lambda

Esta seção pressupõe que seu ponto de acesso do Object Lambda está configurado para chamar a função do Lambda para ListObjectsV2. O Lambda receberá a carga JSON com um novo objeto chamado listObjectsV2Context. listObjectsV2Context contém uma única propriedade, inputS3Url, que é um URL pré-assinado para o ponto de acesso de suporte para ListObjectsV2.

Ao contrário de GetObject e HeadObject, o URL pré-assinado incluirá as seguintes propriedades se elas forem especificadas:

- Todos os parâmetros de consulta
- requestPayer (no cabeçalho de x-amz-request-payer)

- `expectedBucketOwner` (no cabeçalho de `x-amz-expected-bucket-owner`)

Para ver os parâmetros de URI da sintaxe da solicitação, consulte [ListObjectsV2](#) na Referência da API do Amazon Simple Storage Service.

O exemplo a seguir mostra a carga útil JSON de entrada do Lambda para `ListObjectsV2`.

```
{
  "xAmzRequestId": "requestId",
  "**listObjectsV2Context**": {
    "**inputS3Url**": "https://my-s3-ap-111122223333.s3-accesspoint.us-east-1.amazonaws.com/?list-type=2&X-Amz-Security-Token=<snip>",
  },
  "configuration": {
    "accessPointArn": "arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/example-object-lambda-ap",
    "supportingAccessPointArn": "arn:aws:s3:us-east-1:111122223333:accesspoint/example-ap",
    "payload": "{}"
  },
  "userRequest": {
    "url": "https://object-lambda-111122223333.s3-object-lambda.us-east-1.amazonaws.com/example",
    "headers": {
      "Host": "object-lambda-111122223333.s3-object-lambda.us-east-1.amazonaws.com",
      "Accept-Encoding": "identity",
      "X-Amz-Content-SHA256": "e3b0c44298fc1example"
    }
  },
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "principalId",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
    "accountId": "111122223333",
    "accessKeyId": "accessKeyId",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
      }
    },
    "sessionIssuer": {
      "type": "Role",
```

```
        "principalId": "principalId",
        "arn": "arn:aws:iam::111122223333:role/Admin",
        "accountId": "111122223333",
        "userName": "Admin"
    }
},
"protocolVersion": "1.00"
}
```

Sua função do Lambda deve retornar um objeto JSON que contém o código de status, o resultado XML da lista ou as informações de erro que serão retornadas do S3 Object Lambda.

O S3 Object Lambda não processa nem valida `listResultXml`. Em vez disso, o encaminha para o chamador de `ListObjectsV2`. Para `listBucketResult`, o S3 Object Lambda espera que certas propriedades sejam de um tipo específico e lançará exceções se não puder analisá-las. `listResultXml` e `listBucketResult` não podem ser fornecidas ao mesmo tempo.

O exemplo a seguir demonstra como usar o URL pré-assinado para chamar o Amazon S3 e usar o resultado para preencher uma resposta, incluindo a verificação de erros.

Python

```
import requests
import xmldict

def lambda_handler(event, context):
    # Extract the presigned URL from the input.
    s3_url = event["listObjectsV2Context"]["inputS3Url"]

    # Get the head of the object from Amazon S3.
    response = requests.get(s3_url)

    # Return the error to S3 Object Lambda (if applicable).
    if (response.status_code >= 400):
        error = xmldict.parse(response.content)
        return {
            "statusCode": response.status_code,
            "errorCode": error["Error"]["Code"],
            "errorMessage": error["Error"]["Message"]
        }
```

```
# Store the XML result in a dict.
response_dict = xmltodict.parse(response.content)

# This obscures StorageClass in a transformation, it is optional to add
for item in response_dict['ListBucketResult']['Contents']:
    item['StorageClass'] = ""

# Convert back to XML.
listResultXml = xmltodict.unparse(response_dict)

# Create response with listResultXml.
response_with_list_result_xml = {
    'statusCode': 200,
    'listResultXml': listResultXml
}

# Create response with listBucketResult.
response_dict['ListBucketResult'] =
sanitize_response_dict(response_dict['ListBucketResult'])
response_with_list_bucket_result = {
    'statusCode': 200,
    'listBucketResult': response_dict['ListBucketResult']
}

# Return the list to S3 Object Lambda.
# Can return response_with_list_result_xml or response_with_list_bucket_result
return response_with_list_result_xml

# Converting the response_dict's key to correct casing
def sanitize_response_dict(response_dict: dict):
    new_response_dict = dict()
    for key, value in response_dict.items():
        new_key = key[0].lower() + key[1:] if key != "ID" else 'id'
        if type(value) == list:
            newlist = []
            for element in value:
                if type(element) == type(dict()):
                    element = sanitize_response_dict(element)
                newlist.append(element)
            value = newlist
        elif type(value) == dict:
            value = sanitize_response_dict(value)
        new_response_dict[new_key] = value
```

```
return new_response_dict
```

O exemplo a seguir mostra a estrutura do JSON de resposta do Lambda para ListObjectsV2.

```
{
  "statusCode": <number>; // Required
  "errorCode": <string>;
  "errorMessage": <string>;
  "listResultXml": <string>; // This can also be Error XML string in case S3 returned
error response when calling the pre-signed URL

  "listBucketResult": { // listBucketResult can be provided instead of
listResultXml, however they can not both be provided in the JSON response
    "name": <string>, // Required for 'listBucketResult'
    "prefix": <string>,
    "startAfter": <string>,
    "continuationToken": <string>,
    "nextContinuationToken": <string>,
    "keyCount": <int>, // Required for 'listBucketResult'
    "maxKeys": <int>, // Required for 'listBucketResult'
    "delimiter": <string>,
    "encodingType": <string>
    "isTruncated": <boolean>, // Required for 'listBucketResult'
    "contents": [ {
      "key": <string>, // Required for 'content'
      "lastModified": <string>,
      "eTag": <string>,
      "checksumAlgorithm": <string>, // CRC32, CRC32C, SHA1, SHA256
      "size": <int>, // Required for 'content'
      "owner": {
        "displayName": <string>, // Required for 'owner'
        "id": <string>, // Required for 'owner'
      },
      "storageClass": <string>
    },
    ...
  ],
  "commonPrefixes": [ {
    "prefix": <string> // Required for 'commonPrefix'
  },
  ...
],
```

```
}
}
```

Formato e uso de contexto de evento

O Amazon S3 Object Lambda fornece contexto sobre a solicitação que está sendo feita no evento passado para a função AWS Lambda. Veja a seguir um exemplo de solicitação. As descrições dos campos são incluídas após o exemplo.

```
{
  "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"
}
```

Os seguintes arquivos estão incluídos na solicitação:

- `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 as permissões desse usuário 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 a 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 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.

Trabalhar com cabeçalhos Range e partNumber

Ao trabalhar com objetos grandes no Amazon S3 Object Lambda, 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 executa uma solicitação em um intervalo para a parte especificada do objeto.

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 AWS Lambda deve implementar essa funcionalidade conforme necessário para a aplicação.

Para usar os parâmetros Range e `partNumber` com o S3 Object Lambda, faça o seguinte:

- Ative esses parâmetros na configuração do ponto de acesso do Object Lambda.
- Escreva uma função do Lambda capaz de lidar com solicitações que incluam esses parâmetros.

As etapas a seguir descrevem como fazer isso.

Etapa 1: configurar o ponto de acesso do Object Lambda


Por padrão, os pontos de acesso do Object Lambda respondem com um erro de código de status HTTP 501 (não implementado) a qualquer solicitação `GetObject` ou `HeadObject` 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 essas solicitações, é necessário incluir `GetObject-Range`, `GetObject-PartNumber`, `HeadObject-Range` ou `HeadObject-PartNumber` na seção `AllowedFeatures` da configuração do ponto de acesso do Object Lambda. Para obter mais informações sobre como atualizar a configuração do ponto de acesso do Object Lambda, consulte [Criar pontos de acesso do Object Lambda](#).

Etapa 2: Implementar o tratamento de **Range** ou **partNumber** na função do Lambda

Quando o ponto de acesso do Object Lambda invoca a função do Lambda com uma solicitação `GetObject` ou `HeadObject` 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
Range (cabeçalho)	<code>userRequest.headers.Range</code>
Range (parâmetro de consulta)	<code>userRequest.url</code> (parâmetro de consulta Range)
<code>partNumber</code>	<code>userRequest.url</code> (parâmetro de consulta <code>partNumber</code>)

 Important

O URL pré-assinado fornecido para o ponto de acesso do Object Lambda não contém o parâmetro Range ou `partNumber` da solicitação original. Veja as opções a seguir sobre como lidar com esses parâmetros na função do AWS Lambda.

Depois de extrair o valor Range ou `partNumber`, você poderá adotar uma das seguintes abordagens com base nas necessidades da aplicação:

A. Mapear o **Range** ou **partNumber** solicitado para o objeto transformado (recomendado).

O jeito mais confiável de lidar com solicitações Range e `partNumber` é fazer o seguinte:

- Recupere o objeto completo do Amazon S3.
- Transforme o objeto.
- Aplique 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. Associe o **Range** solicitado ao URL pré-assinado.

Em alguns casos, a função do Lambda pode associar o Range solicitado diretamente ao URL pré-assinado para recuperar apenas parte do objeto do Amazon S3. Essa abordagem será apropriada somente se sua transformação atender a ambos os seguintes critérios:

1. Sua função de transformação pode ser aplicada a intervalos de objetos parciais.

2. Aplicar o parâmetro Range antes ou depois da função de transformação ocasiona a transformação do mesmo objeto.

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 antes da transformação produz o 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.

Pense em 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 reverterá 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 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 na abordagem B, você poderá simplificar a função do AWS Lambda buscando apenas o intervalo de objetos solicitado e executando a transformação nesse intervalo.

O exemplo de código Java a seguir demonstra como fazer o seguinte:

- Recuperar o cabeçalho Range da solicitação `GetObject`.

- Adicionar o cabeçalho Range 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;
}
```

Usar funções do Lambda criadas pela AWS

A AWS fornece algumas funções pré-criadas do AWS Lambda que você pode usar com o Amazon 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. Essas funções do Lambda estão disponíveis no AWS Management Console quando você cria um ponto de acesso do Object Lambda.

Para obter mais informações sobre como implantar aplicações de tecnologia sem servidor pelo AWS Serverless Application Repository, consulte [Implantação de aplicações](#) no Guia do desenvolvedor do AWS Serverless Application Repository.

Note

Os exemplos a seguir só podem ser usados com solicitações `GetObject`.

Exemplo 1: Controle de acesso de PII

Essa função do Lambda usa o Amazon Comprehend, um serviço de processamento de linguagem natural (PLN) que usa machine learning para encontrar insights e relações no texto. Essa função 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

de controle de acesso de PII para detectar esses tipos de entidade de PII e restringir o acesso a usuários não autorizados.

Para começar, implante a função do Lambda a seguir em sua conta e adicione o nome do recurso da Amazon (ARN) da função na configuração do ponto de acesso do Object Lambda.

Veja o seguinte exemplo de ARN para essa função:

```
arn:aws:serverlessrepo:us-east-1:111122223333:applications/  
ComprehendPiiAccessControlS3ObjectLambda
```

Você pode adicionar ou exibir essa função no AWS Management Console usando o seguinte link do AWS Serverless Application Repository: [ComprehendPiiAccessControlS3ObjectLambda](#).

Para visualizar essa função no GitHub, consulte [Amazon Comprehend S3 Object Lambda](#).

Exemplo 2: Edição de PII

Essa função do Lambda usa o Amazon Comprehend, um serviço de processamento de linguagem natural (PLN) que usa machine learning para encontrar insights e relações no texto. Essa função 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, implante a função do Lambda a seguir em sua conta e adicione o ARN da função na configuração do ponto de acesso do Object Lambda.

Veja a seguir um exemplo de ARN para essa função:

```
arn:aws:serverlessrepo:us-east-1:111122223333::applications/  
ComprehendPiiRedactionS3ObjectLambda
```

Você pode adicionar ou exibir essa função no AWS Management Console usando o seguinte link do AWS Serverless Application Repository: [ComprehendPiiRedactionS3ObjectLambda](#).

Para visualizar essa função no GitHub, consulte [Amazon Comprehend S3 Object Lambda](#).

Para saber mais sobre procedimentos completos para algumas tarefas do S3 Object Lambda em edição de PII, consulte [Tutorial: Detectar e editar dados PII com o S3 Object Lambda e o Amazon Comprehend](#).

Exemplo 3: descompressão

A função `S3ObjectLambdaDecompression` do Lambda pode descompactar objetos armazenados no Amazon S3 em um dos seis formatos de arquivo compactados: `bzip2`, `gzip`, `snappy`, `zlib`, `zstandard` e `ZIP`.

Para começar, implante a função do Lambda a seguir em sua conta e adicione o ARN da função na configuração do ponto de acesso do Object Lambda.

Veja a seguir um exemplo de ARN para essa função:

```
arn:aws:serverlessrepo:us-east-1:111122223333::applications/S3ObjectLambdaDecompression
```

Você pode adicionar ou exibir essa função no AWS Management Console usando o seguinte link do AWS Serverless Application Repository: [S3ObjectLambdaDecompression](#).

Para visualizar essa função no GitHub, consulte [S3 Object Lambda Decompression](#).

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](#)
- [Serviços da AWS usados em conexão com o S3 Object Lambda](#)
- [Cabeçalhos Range e partNumber](#)
- [Transformar expiry-date](#)
- [Como trabalhar com a AWS CLI e AWS SDKs](#)

Trabalhar com o S3 Object Lambda

O S3 Object Lambda é compatível apenas com o processamento de solicitações GET, LIST e HEAD. Qualquer outra solicitação não invocará AWS Lambda; em vez disso, 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 AWS Lambda. Para obter mais informações, consulte [Cotas do Lambda](#) no Guia do desenvolvedor do AWS Lambda.

Quando o S3 Object Lambda invoca a função do Lambda especificada, você é responsável por garantir que todos os dados substituídos ou excluídos do Amazon S3 pela função ou aplicação do Lambda especificada sejam intencionais e corretos.

Você só pode usar o S3 Object Lambda para realizar 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 de ponto de acesso com operações do S3](#).

Além dessa lista, os pontos de acesso do Object Lambda não são compatíveis com as operações de API [POST Object](#), [CopyObject](#) (como origem) e [SelectObjectContent](#).

Serviços da AWS usados em conexão com o S3 Object Lambda

O S3 Object Lambda conecta Amazon S3, o AWS Lambda e, opcionalmente, outros Serviços da AWS de sua escolha para fornecer objetos relevantes para as aplicações solicitadas. Todos os Serviços da AWS usados com o S3 Object Lambda são regidos por seus respectivos Acordos de Serviço (SLAs). Por exemplo, no caso de qualquer AWS service (Serviço da AWS) não cumprir seu compromisso de serviço, você será elegível para receber um crédito de serviço conforme documentado no SLA do serviço.

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. Quando você usa o cabeçalho Range, sua solicitação busca somente a parte especificada do objeto. Você também pode usar o cabeçalho `partNumber` para realizar uma solicitação em um intervalo para a parte especificada do objeto.

Consulte mais informações em [Trabalhar com cabeçalhos Range e partNumber](#).

Transformar **expiry-date**

Você pode abrir ou baixar objetos transformados do ponto de acesso do Object Lambda no AWS Management Console. Esses objetos devem ser válidos (não expirados). Se sua função do Lambda

transformar `expiry-date` dos objetos, você poderá ver objetos expirados que não poderão ser abertos nem baixados. Esse comportamento se aplica somente a objetos restaurados do S3 Glacier Flexible Retrieval e do S3 Glacier Deep Archive.

Como trabalhar com a AWS CLI e AWS SDKs

Os subcomandos do S3 (`cp`, `mv` e `sync`) da AWS Command Line Interface (AWS CLI) e o uso da classe AWS SDK for Java `TransferManager` não são compatíveis com o S3 Object Lambda.

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](#)
- [Tutorial: Detectar e editar dados PII com o S3 Object Lambda e o Amazon Comprehend](#)
- [Tutorial: Using S3 Object Lambda to dynamically watermark images as they are retrieved](#) (Tutorial: Como usar o S3 Object Lambda para colocar marca d'água em imagens de maneira dinâmica à medida que são recuperadas)

Depuração do S3 Object Lambda

As solicitações para pontos de acesso do Amazon 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 os pontos de acesso do Object Lambda. Essas métricas ajudam você a monitorar a performance operacional do seu ponto de acesso. Você pode habilitar métricas de solicitação durante ou após a criação de um ponto de

acesso do Object Lambda. Para ter mais informações, consulte [Métricas de solicitação do S3 Object Lambda no CloudWatch](#).

Para obter registros em log mais granulares sobre solicitações feitas aos seus pontos de acesso do Object Lambda, é possível ativar 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.

Para tutoriais do S3 Object Lambda, consulte o seguinte:

- [Tutorial: Como transformar dados para sua aplicação com o S3 Object Lambda](#)
- [Tutorial: Detectar e editar dados PII com o S3 Object Lambda e o Amazon Comprehend](#)
- [Tutorial: Using S3 Object Lambda to dynamically watermark images as they are retrieved](#) (Tutorial: Como usar o S3 Object Lambda para colocar marca d'água em imagens de maneira dinâmica à medida que são recuperadas)

Para obter mais informações sobre pontos de acesso padrão, consulte [Gerenciamento de acesso a dados com pontos de acesso do Amazon S3](#).

Para obter informações sobre como trabalhar com buckets, consulte [Visão geral dos buckets](#). Para obter mais informações sobre como trabalhar com objetos, consulte [Visão geral de objetos Amazon S3](#).

O que é a classe S3 Express One Zone?

A classe Amazon S3 Express One Zone é uma classe de armazenamento de zona única e alto desempenho do Amazon S3 desenvolvida com o propósito específico de fornecer acesso consistente aos dados e com latência inferior a dez milissegundos para as aplicações mais sensíveis à latência. A classe S3 Express One Zone é a classe de armazenamento de objetos em nuvem de menor latência disponível atualmente, com velocidades de acesso aos dados até dez vezes mais rápidas e custos de solicitação 50% mais baixos do que a classe S3 Standard. As aplicações podem se beneficiar imediatamente da conclusão de solicitações até uma ordem de magnitude mais rápida. A classe S3 Express One Zone oferece elasticidade de performance semelhante a outras classes de armazenamento do S3.

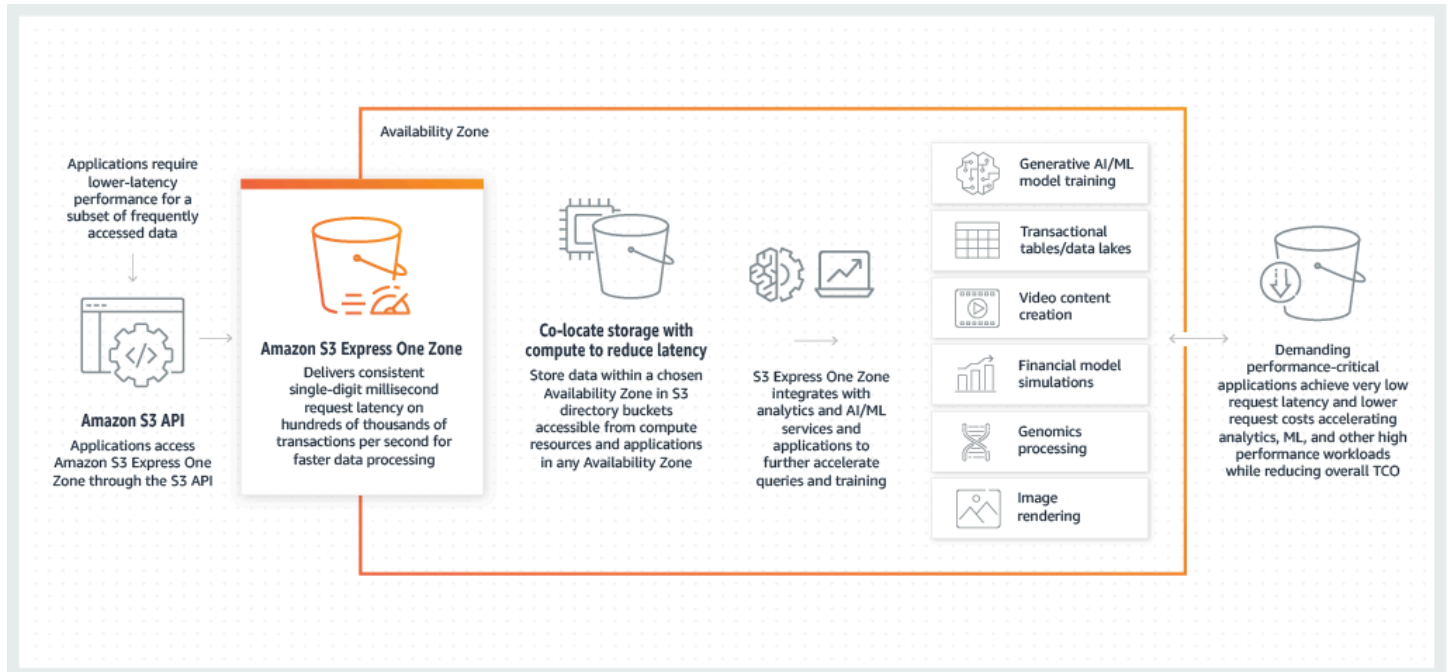
Assim como outras classes de armazenamento do Amazon S3, não é necessário planejar ou provisionar requisitos de capacidade ou throughput com antecedência. Você pode aumentar ou reduzir a escala do armazenamento verticalmente, de acordo com a necessidade, e acessar os dados por meio da API do Amazon S3.

A classe S3 Express One Zone é a primeira classe de armazenamento do S3 em que é possível selecionar uma única zona de disponibilidade com a opção de manter o armazenamento de objetos e os recursos de computação na mesma localização, o que fornece a maior velocidade de acesso possível. Além disso, para aumentar ainda mais a velocidade de acesso e compatibilidade a centenas de milhares de solicitações por segundo, os dados da classe S3 Express One Zone são armazenados em um novo tipo de bucket: um bucket de diretório do Amazon S3. Cada bucket de diretório é capaz de oferecer suporte a centenas de milhares de transações por segundo (TPS), independentemente dos nomes de chave ou do padrão de acesso.

A classe de armazenamento Amazon S3 Express One Zone foi projetada para oferecer disponibilidade de 99,95% em uma única zona de disponibilidade e tem o respaldo do [Acordo de Serviço do Amazon S3](#). Com a classe S3 Express One Zone, os dados são armazenados de forma redundante em vários dispositivos dentro de uma única zona de disponibilidade. A classe S3 Express One Zone foi projetada para lidar com falhas simultâneas de dispositivos, detectando e reparando rapidamente qualquer redundância perdida. Se o dispositivo existente encontrar uma falha, a classe S3 Express One Zone transferirá automaticamente as solicitações para novos dispositivos dentro da mesma zona de disponibilidade. Essa redundância ajuda a garantir o acesso ininterrupto aos dados dentro de uma zona de disponibilidade.

A classe S3 Express One Zone é ideal para qualquer aplicação em que seja importante minimizar a latência necessária para acessar um objeto. Tais aplicações podem ser fluxos de trabalho com

interatividade humana, como edição de vídeo, em que profissionais criativos precisam de acesso responsivo ao conteúdo por meio de suas interfaces de usuário. A classe S3 Express One Zone também beneficia workloads de análise e machine learning que têm requisitos de capacidade de resposta semelhantes dos dados, especialmente workloads com muitos acessos menores ou grandes quantidades de acessos aleatórios. A classe S3 Express One Zone pode ser usada com outros Serviços da AWS para oferecer suporte a workloads de análise e de inteligência artificial/machine learning (IA/ML), como Amazon EMR, Amazon SageMaker e Amazon Athena.



Ao usar a classe S3 Express One Zone, você pode interagir com o bucket de diretório em uma nuvem privada virtual (VPC) usando um endpoint da VPC do gateway. Com um endpoint de gateway, é possível acessar buckets de diretório da classe S3 Express One Zone pela VPC sem precisar de um gateway da Internet ou um dispositivo NAT para a VPC, e tudo isso sem custos adicionais.

Você pode usar muitas das mesmas operações de API e recursos do Amazon S3 com buckets de diretório que já usa com buckets de uso geral e outras classes de armazenamento. Isso inclui o Mountpoint para Amazon S3, a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) e as funcionalidades Operações em Lote do S3 e Bloqueio de Acesso Público do S3. Você pode acessar a classe de armazenamento S3 Express One Zone pelo console do Amazon S3, pela AWS Command Line Interface (AWS CLI), pelos AWS SDKs e pela API REST do Amazon S3.

Para obter mais informações sobre a classe S3 Express One Zone, consulte os tópicos a seguir.

- [Visão geral](#)
- [Recursos da classe S3 Express One Zone](#)
- [Serviços relacionados](#)
- [Próximas etapas](#)

Visão geral

Para otimizar a performance e reduzir a latência, a classe S3 Express One Zone apresenta os novos conceitos a seguir.

Zona de disponibilidade única

A classe de armazenamento Amazon S3 Express One Zone foi projetada para oferecer disponibilidade de 99,95% em uma única zona de disponibilidade e tem o respaldo do [Acordo de Serviço do Amazon S3](#). Com a classe S3 Express One Zone, os dados são armazenados de forma redundante em vários dispositivos dentro de uma única zona de disponibilidade. A classe S3 Express One Zone foi projetada para lidar com falhas simultâneas de dispositivos, detectando e reparando rapidamente qualquer redundância perdida. Se o dispositivo existente encontrar uma falha, a classe S3 Express One Zone transferirá automaticamente as solicitações para novos dispositivos dentro da mesma zona de disponibilidade. Essa redundância ajuda a garantir o acesso ininterrupto aos dados dentro de uma zona de disponibilidade.

Uma zona de disponibilidade é um ou mais datacenters discretos com energia, redes e conectividade redundantes em uma Região da AWS. Ao criar um bucket de diretório, escolha a zona de disponibilidade e a Região da AWS em que o bucket ficará localizado.

Buckets de diretório

Há dois tipos de buckets do Amazon S3: buckets de uso geral do S3 e buckets de diretório do S3. Os buckets de uso geral são o tipo padrão de bucket do Amazon S3 usado na grande maioria dos casos de uso do S3. Os buckets de diretório usam apenas a classe de armazenamento S3 Express One Zone, projetada para workloads ou aplicações de performance crítica que exigem latência consistente inferior a dez milissegundos. Escolha o tipo de bucket que melhor se adapte aos requisitos de performance e da aplicação.

Os buckets de diretório organizam os dados de forma hierárquica em diretórios, ao contrário da estrutura de armazenamento plana dos buckets de uso geral. Não há limites de prefixo para os buckets de diretório, e diretórios individuais podem ser escalados horizontalmente.

Os buckets de diretório usam a classe de armazenamento S3 Express One Zone, desenvolvida para ser usada por aplicações sensíveis à performance. Com a classe S3 Express One Zone, é possível selecionar uma única zona de disponibilidade com a opção de manter o armazenamento de objetos e os recursos de computação na mesma localização, o que fornece a maior velocidade de acesso possível. Isso é diferente dos buckets de uso geral, que armazenam objetos de forma redundante em várias zonas de disponibilidade em Regiões da AWS.

Para obter mais informações sobre os buckets de diretório, consulte [Buckets de diretório](#). Para obter mais informações sobre os buckets de uso geral, consulte [Visão geral dos buckets](#).

Endpoints e endpoints de VPC do gateway

As operações de API de gerenciamento para buckets de diretório estão disponíveis por meio de um endpoint regional e são chamadas de operações de API de endpoint regional. Exemplos de operações de API de endpoint regional são `CreateBucket` e `DeleteBucket`. Depois de criar um bucket de diretório, você pode usar operações de API de endpoint zonal para fazer upload e gerenciar os objetos no bucket de diretório. As operações de API de endpoint zonal estão disponíveis por meio de um endpoint zonal. Exemplos de operações de API de endpoint zonal são `PutObject` e `CopyObject`.

É possível acessar a classe S3 Express One Zone pela VPC usando endpoints de VPC do gateway. Depois de criar um endpoint de gateway, é possível adicioná-lo como um destino na tabela de rotas para o tráfego destinado da VPC à classe S3 Express One Zone. Assim como no Amazon S3, não há cobrança adicional pelo uso de endpoints de gateway. Para obter mais informações sobre como configurar endpoints de VPC do gateway, consulte [Redes para a classe S3 Express One Zone](#).

Autorização baseada em sessão

Com a classe S3 Express One Zone, é possível autenticar e autorizar solicitações por meio de um novo mecanismo baseado em sessão que é otimizado para fornecer a menor latência. Você pode usar `CreateSession` para solicitar credenciais temporárias que forneçam acesso de baixa latência ao bucket. Essas credenciais temporárias têm como escopo um bucket de diretório específico do S3. Os tokens de sessão são usados somente com operações zonais (nível de objeto) (com exceção de [CopyObject](#)). Para ter mais informações, consulte [CreateSession authorization](#).

Os [AWS SDKs compatíveis com a classe S3 Express One Zone](#) lidam com o estabelecimento e a atualização de sessões em seu nome. Para proteger as sessões, as credenciais de segurança temporárias expiram após 5 minutos. Depois de baixar e instalar os AWS SDKs e configurar as

permissões necessárias do AWS Identity and Access Management (IAM), você pode começar imediatamente a usar as operações de API.

Recursos da classe S3 Express One Zone

Os recursos do S3 a seguir estão disponíveis para a classe S3 Express One Zone. Para obter uma lista completa de operações de APIs compatíveis e recursos não compatíveis, consulte [O que diferencia a classe S3 Express One Zone?](#).

Gerenciamento de acesso e segurança

Com buckets de diretório, você pode usar os recursos a seguir para auditar e gerenciar o acesso. Por padrão, os buckets de diretório são privados e só podem ser acessados por usuários que tenham recebido acesso explícito. Ao contrário dos buckets de uso geral que podem definir o limite de controle de acesso no nível do bucket, do prefixo ou da etiqueta de objeto, o limite do controle de acesso para buckets de diretório é definido somente no nível do bucket. Para ter mais informações, consulte [AWS Identity and Access Management \(IAM\) para a classe S3 Express One Zone](#).

- [Bloqueio de Acesso Público do S3](#): todas as configurações da funcionalidade Bloqueio de Acesso Público do S3 são habilitadas por padrão no nível do bucket. Essa configuração padrão não pode ser modificada.
- [Propriedade de Objetos do S3](#) (padrão: Imposto pelo proprietário do bucket): as listas de controle de acesso (ACLs) não são compatíveis com buckets de diretório. Os buckets de diretório usam automaticamente a configuração Imposto pelo proprietário do bucket para a funcionalidade Propriedade de Objetos do S3. A opção Imposto pelo proprietário do bucket significa que as ACLs são desabilitadas e que o proprietário do bucket tem automaticamente a propriedade e o controle total sobre todos os objetos do bucket. Essa configuração padrão não pode ser modificada.
- [AWS Identity and Access Management \(IAM\)](#): o IAM ajuda você a controlar com segurança o acesso aos buckets de diretório. Você pode usar o IAM para conceder acesso às operações de API gerenciamento de buckets (regionais) e às operações de API de gerenciamento de objetos (zonais) por meio da ação `s3express:CreateSession`. Para ter mais informações, consulte [AWS Identity and Access Management \(IAM\) para a classe S3 Express One Zone](#). Ao contrário das ações de gerenciamento de objetos, as ações de gerenciamento de buckets não podem ser executadas entre contas. Somente o proprietário do bucket pode realizar essas ações.
- [Políticas de bucket](#): use a linguagem de política baseada no IAM para configurar permissões baseadas em recurso para os buckets de diretório. Você também pode usar o IAM para controlar

o acesso à operação de API `CreateSession`, que permite usar as operações de API zonais ou de gerenciamento de objetos. Você pode conceder acesso entre contas ou na mesma conta para operações de API zonais. Para obter mais informações sobre as permissões e políticas da classe S3 Express One Zone, consulte [AWS Identity and Access Management \(IAM\) para a classe S3 Express One Zone](#).

- [IAM Access Analyzer para S3](#): avalie e monitore as políticas de acesso a fim de garantir que elas forneçam apenas o acesso pretendido aos recursos do S3.

Registrar em log e monitoramento

A classe S3 Express One Zone usa as seguintes ferramentas de registro em log e monitoramento do S3, que você pode usar para monitorar e controlar como os recursos do estão sendo usados:

- [Métricas do Amazon CloudWatch](#): monitore os recursos e as aplicações da AWS usando o CloudWatch para coletar e monitorar métricas. A classe S3 Express One Zone usa o mesmo namespace do CloudWatch que outras classes de armazenamento do Amazon S3 (AWS/S3) e oferece suporte a métricas de armazenamento diárias para buckets de diretório: `BucketSizeBytes` e `NumberOfObjects`. Para ter mais informações, consulte [Monitoramento de métricas com o Amazon CloudWatch](#).
- [Logs do AWS CloudTrail](#): o AWS CloudTrail é um AWS service (Serviço da AWS) que ajuda você a implementar a auditoria, a governança e a conformidade operacionais e de risco da sua Conta da AWS ao registrar as ações realizadas por um usuário, um perfil ou um AWS service (Serviço da AWS). Para a classe S3 Express One Zone, o CloudTrail captura operações de API de endpoint regional (por exemplo, `CreateBucket` e `PutBucketPolicy`) como eventos de gerenciamento. Esses eventos incluem ações realizadas no AWS Management Console, na AWS Command Line Interface (AWS CLI), em AWS SDKs e em operações de API da AWS. O `eventsource` para os eventos de gerenciamento do CloudTrail para a classe S3 Express One Zone é `s3express.amazonaws.com`. Para ter mais informações, consulte [Eventos do CloudTrail no Amazon S3](#).

Note

Os logs de acesso ao servidor do Amazon S3 não são compatíveis com a classe S3 Express One Zone.

Gerenciamento de objetos

Depois de criar um bucket de diretório, você pode gerenciar o armazenamento de objetos usando o console do Amazon S3, os AWS SDKs e a AWS CLI. Os seguintes recursos estão disponíveis para o gerenciamento de objetos com a classe S3 Express One Zone:

- [Operações em Lote do S3](#): use a funcionalidade Operações em Lote para realizar operações em massa nos objetos em buckets de diretório, como Copiar e Invocar função do AWS Lambda. Por exemplo, você pode usar a funcionalidade Operações em Lote para copiar objetos entre buckets de diretório e buckets de uso geral. Com a funcionalidade Operações em Lote, você pode gerenciar bilhões de objetos em grande escala com uma única solicitação do S3 usando os AWS SDKs ou a AWS CLI ou com alguns cliques no console do Amazon S3.
- [Importação](#): depois de criar um bucket de diretório, você poderá preenchê-lo com objetos usando o recurso de importação no console do Amazon S3. Importar é um método simplificado para criar trabalhos da funcionalidade Operações em Lote a fim de copiar objetos de buckets de uso geral para buckets de diretório.

AWS SDKs e bibliotecas de clientes

Assim que criar um bucket de diretório e fazer upload de um objeto no bucket, você poderá gerenciar o armazenamento de objetos usando o seguinte:

- [Mountpoint para Amazon S3](#): o Mountpoint para Amazon S3 é um cliente de arquivos de código aberto que oferece acesso de alto throughput, reduzindo os custos de computação para data lakes no Amazon S3. O Mountpoint para Amazon S3 converte chamadas de API do sistema de arquivos local em chamadas de API de objetos do S3, como GET e LIST. É ideal para workloads de data lake com muitas leituras que processam petabytes de dados e precisam do alto throughput elástico fornecido pelo Amazon S3 para aumentar e reduzir a escala verticalmente em milhares de instâncias.
- [S3A](#): o S3A é uma interface compatível com Hadoop recomendada para acessar datastores no Amazon S3. O S3A substitui o cliente do sistema de arquivos S3N do Hadoop.
- [PyTorch na AWS](#): o PyTorch na AWS é um framework de aprendizado profundo de código aberto que facilita o desenvolvimento de modelos de machine learning e a implantação deles na produção.
- [AWS SDKs](#): você pode usar os AWS SDKs ao desenvolver aplicações com o Amazon S3. Os AWS SDKs simplificam suas tarefas de programação agrupando a API REST subjacente do

Amazon S3. Para obter mais informações sobre como usar os AWS SDKs com a classe S3 Express One Zone, consulte [the section called “SDKs da AWS”](#).

Criptografia e proteção de dados

Os objetos armazenados em buckets de diretório são criptografados automaticamente usando criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3). Os buckets de diretório não oferecem suporte à criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS), criptografia do lado do servidor com chaves de criptografia fornecidas pelo cliente (SSE-C) ou criptografia de camada dupla do lado do servidor com AWS KMS keys (DSSE-KMS). Para obter mais informações, consulte [Proteção e criptografia de dados](#) e [Usar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#).

A classe S3 Express One Zone oferece a opção de escolher o algoritmo de soma de verificação usado para validar os 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 cíclica (CRC): CRC32, CRC32C, SHA-1 e SHA-256. As somas de verificação baseadas em MD5 não são compatíveis com a classe de armazenamento S3 Express One Zone.

Para ter mais informações, consulte [Práticas recomendadas adicionais de soma de verificação do S3](#).

Signature Version 4 (SigV4) da AWS

A classe S3 Express One Zone usa o Signature Version 4 (SigV4) da AWS. SigV4 é um protocolo de assinatura usado para autenticar solicitações ao Amazon S3 via HTTPS. A classe S3 Express One Zone assina solicitações usando o Sigv4 da AWS. Para obter mais informações, consulte [Autenticação de solicitações \(AWS Signature Version 4\)](#) na Referência de APIs do Amazon Simple Storage Service.

Consistência forte

A classe S3 Express One Zone oferece uma sólida consistência de leitura após gravação para solicitações PUT e DELETE de objetos em buckets de diretório em todas as Regiões da AWS. Para ter mais informações, consulte [Modelo de consistência de dados do Amazon S3](#).

Serviços relacionados

Você pode usar os seguintes Serviços da AWS com a classe de armazenamento S3 Express One Zone para oferecer suporte ao seu caso de uso específico de baixa latência.

- [Amazon Elastic Compute Cloud \(Amazon EC2\)](#): o Amazon EC2 oferece capacidade de computação segura e 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.
- [AWS Lambda](#): o Lambda é um serviço de computação que permite que você execute o código sem provisionar ou gerenciar servidores. Você define as configurações de notificação em um bucket e concede permissão ao Amazon S3 para invocar uma função na política de permissões baseada em recursos da função.
- [Amazon Elastic Kubernetes Service \(Amazon EKS\)](#): o Amazon EKS é um serviço gerenciado que elimina a necessidade de instalar, operar e manter o seu próprio ambiente de gerenciamento do Kubernetes na AWS. O [Kubernetes](#) é um sistema de código aberto que automatiza o gerenciamento, a escalação e a implantação de aplicações containerizadas.
- [Amazon Elastic Container Service \(Amazon ECS\)](#): o Amazon ECS é um serviço totalmente gerenciado de orquestração de contêineres que ajuda a implantar, gerenciar e dimensionar facilmente aplicações em contêineres.
- [Amazon Athena](#): o Athena é um serviço de consultas interativas que facilita a análise de dados no Amazon S3 usando [SQL](#) padrão. Você também pode usar o Athena para executar análises de dados de maneira interativa usando o Apache Spark sem a necessidade de planejamento, configuração ou gerenciamento de recursos. Ao executar aplicações do Apache Spark no Athena, você envia o código Spark para processamento e recebe os resultados de forma direta.
- [Treinamento de modelos do Amazon SageMaker Runtime](#): o Amazon SageMaker Runtime é um serviço de machine learning totalmente gerenciado. Com o SageMaker Runtime, cientistas de dados e desenvolvedores podem criar e treinar modelos de machine learning com rapidez e facilidade, depois implantá-los diretamente em um ambiente hospedado pronto para produção.
- [AWS Glue](#): o AWS Glue é um serviço de integração de dados com tecnologia sem servidor que facilita aos usuários de análise a descoberta, preparação, transferência e integração de dados de várias fontes. Você pode usar o AWS Glue para análise, machine learning e desenvolvimento de aplicações. O AWS Glue também inclui outras ferramentas de produtividade e operações de dados para criação, execução de trabalhos e implementação de fluxos de trabalho de negócios.

- [Amazon EMR](#): o Amazon EMR é uma plataforma de cluster gerenciada que simplifica a execução de frameworks de big data, como Apache Hadoop e Apache Spark, na AWS a fim de processar e analisar grandes volumes de dados.

Próximas etapas

Para obter mais informações sobre como usar a classe de armazenamento S3 Express One Zone e os buckets de diretório, consulte os seguintes tópicos:

- [O que diferencia a classe S3 Express One Zone?](#)
- [Conceitos básicos da classe S3 Express One Zone](#)
- [Redes para a classe S3 Express One Zone](#)
- [Buckets de diretório](#)
- [Trabalhar com objetos em um bucket de diretório](#)
- [Segurança para a classe S3 Express One Zone](#)
- [Otimizar a performance da classe Amazon S3 Express One Zone](#)
- [Desenvolver com a classe S3 Express One Zone](#)

O que diferencia a classe S3 Express One Zone?

A classe Amazon S3 Express One Zone é uma classe de armazenamento de zona única e alto desempenho do Amazon S3 desenvolvida com o propósito específico de fornecer acesso consistente aos dados e com latência inferior a dez milissegundos para as aplicações mais sensíveis à latência. A classe S3 Express One Zone é a primeira classe de armazenamento do S3 em que é possível selecionar uma única zona de disponibilidade com a opção de manter o armazenamento de objetos e os recursos de computação na mesma localização, o que fornece a maior velocidade de acesso possível. Além disso, para aumentar ainda mais a velocidade de acesso e compatibilidade a centenas de milhares de solicitações por segundo, os dados da classe S3 Express One Zone são armazenados em um novo tipo de bucket: um bucket de diretório do Amazon S3.

Para obter mais informações, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Você pode criar buckets de diretório e acessar os dados na classe S3 Express One Zone usando a API do Amazon S3. A API do Amazon S3 é compatível com a classe S3 Express One Zone e os

buckets de diretório, com exceção de algumas diferenças notáveis. Para obter mais informações sobre o que diferencia a classe S3 Express One Zone, consulte os tópicos a seguir.

Tópicos

- [Diferenças da classe S3 Express One Zone](#)
- [Operações de API compatíveis com a classe S3 Express One Zone](#)
- [Recursos do Amazon S3 não compatíveis com a classe S3 Express One Zone](#)

Diferenças da classe S3 Express One Zone

- Tipo de bucket compatível: os objetos na classe de armazenamento S3 Express One Zone só podem ser armazenados em buckets de diretório. Para ter mais informações, consulte [Buckets de diretório](#).
- Durabilidade: com a classe S3 Express One Zone, os dados são armazenados de forma redundante em vários dispositivos dentro de uma única zona de disponibilidade. A classe S3 Express One Zone foi projetada para oferecer disponibilidade de 99,95% em uma única zona de disponibilidade e tem o respaldo do [Acordo de Serviço do Amazon S3](#). Para ter mais informações, consulte [Zona de disponibilidade única](#).
- Comportamento de **ListObjectsV2**
 - Para buckets de diretório, ListObjectsV2 não retorna objetos na ordem lexicográfica (alfabética). Além disso, os prefixos devem terminar com um delimitador e somente "/" pode ser especificado como delimitador.
 - Para buckets de diretório, a resposta de ListObjectsV2 inclui os prefixos relacionados somente aos carregamentos fracionados em andamento.
- Comportamento de exclusão: quando você exclui um objeto de um bucket de diretório, o Amazon S3 exclui recursivamente todos os diretórios vazios no caminho do objeto. Por exemplo, se você excluir a chave de objeto dir1/dir2/file1.txt, o Amazon S3 excluirá file1.txt. Se os diretórios dir1/ e dir2/ estiverem vazios e não contiverem outros objetos, o Amazon S3 também excluirá esses diretórios.
- ETags e somas de verificação: as etiquetas de entidade (ETags) da classe S3 Express One Zone são strings alfanuméricas aleatórias e não somas de verificação MD5. Para obter mais informações sobre como usar somas de verificação adicionais com a classe S3 Express One Zone, consulte [Práticas recomendadas adicionais de soma de verificação do S3](#).
- Chaves de objeto em solicitações **DeleteObjects**

- As chaves de objeto em solicitações `DeleteObjects` devem conter pelo menos um caractere diferente de um espaço em branco. Strings compostas somente por caracteres de espaço em branco não são compatíveis com solicitações `DeleteObjects`.
- As chaves de objeto em solicitações `DeleteObjects` não podem conter caracteres de controle Unicode, exceto nova linha (`\t`), tabulação (`\n`) e avanço de linha (`\r`).
- Endpoints regionais e zonais: ao usar a classe `S3 Express One Zone`, você deve especificar a região em todas as solicitações do cliente. Para endpoints regionais, especifique a região, como `s3express-control.us-west-2.amazonaws.com`. Para endpoints zonais, especifique a região e a zona de disponibilidade, como `s3express-usw2-az1.us-west-2.amazonaws.com`. Para ter mais informações, consulte [Endpoints regionais e zonais](#).
- Carregamentos multiparte: assim como acontece com outros objetos armazenados no Amazon S3, você pode fazer upload e copiar objetos grandes armazenados na classe de armazenamento `S3 Express One Zone` usando o processo de carregamento multiparte. No entanto, veja a seguir algumas diferenças ao usar o processo de carregamento multiparte com objetos armazenados na classe `S3 Express One Zone`. Para ter mais informações, consulte [the section called “Usar uploads multiparte com buckets de diretório”](#).
 - A data de criação do objeto é a data de conclusão do multipart upload.
 - Os números das partes de um carregamento multiparte devem ser consecutivos. Se você tentar concluir uma solicitação de multipart upload com números de parte não consecutivos, o Amazon S3 gerará um erro HTTP 400 (Bad Request).
 - O iniciador de um carregamento multiparte só poderá anular a solicitação de carregamento multiparte se tiver recebido acesso explícito a `AbortMultipartUpload` por meio da permissão `s3express:CreateSession`. Para ter mais informações, consulte [AWS Identity and Access Management \(IAM\) para a classe S3 Express One Zone](#).
- Esvaziar um bucket de diretório: o comando `s3 rm` por meio da AWS Command Line Interface (CLI), a operação `delete` por meio do Mountpoint e o botão de opção Esvaziar bucket por meio do AWS Management Console não conseguem excluir carregamentos fracionados em andamento em um bucket de diretório. Para excluir esses carregamentos fracionados em andamento, use a operação `ListMultipartUploads` para listá-los no bucket e a operação `AbortMultipartUpload` para anular todos eles.

Operações de API compatíveis com a classe S3 Express One Zone

A classe de armazenamento Amazon S3 Express One Zone é compatível com operações de API de endpoint regional (nível de bucket ou ambiente de gerenciamento) e zonal (nível de objeto ou

plano de dados). Para obter mais informações, consulte [Redes para a classe S3 Express One Zone](#) e [Endpoints e endpoints de VPC do gateway](#).

Operações de API de endpoint regional

As seguintes operações de API de endpoint regional são compatíveis com a classe S3 Express One Zone:

- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteBucketPolicy](#)
- [GetBucketPolicy](#)
- [ListDirectoryBuckets](#)
- [PutBucketPolicy](#)

Operações de API de endpoint zonal

As seguintes operações de API de endpoint zonal são compatíveis com a classe S3 Express One Zone:

- [CreateSession](#)
- [CopyObject](#)
- [DeleteObject](#)
- [DeleteObjects](#)
- [GetObject](#)
- [GetObjectAttributes](#)
- [HeadBucket](#)
- [HeadObject](#)
- [ListObjectsV2](#)
- [PutObject](#)
- [AbortMultipartUpload](#)
- [CompleteMultiPartUpload](#)
- [CreateMultipartUpload](#)

- [ListMultipartUploads](#)
- [ListParts](#)
- [UploadPart](#)
- [UploadPartCopy](#)

Recursos do Amazon S3 não compatíveis com a classe S3 Express One Zone

Os seguintes recursos do Amazon S3 não são compatíveis com a classe S3 Express One Zone:

- Eventos de plano de dados do AWS CloudTrail
- Políticas gerenciadas pela AWS
- AWS PrivateLink para S3
- Somas de verificação MD5
- Exclusão de autenticação multifator (MFA)
- Bloqueio de objetos do S3
- Requester Pays (Pagamento pelo solicitante)
- Concessões de Acesso do S3
- Pontos de acesso do S3
- Etiquetas de bucket
- Métricas de solicitação do Amazon CloudWatch
- Notificações de eventos do S3
- Ciclo de vida do S3
- Pontos de Acesso Multirregionais do S3
- Pontos de Acesso do S3 Object Lambda
- Versionamento do S3
- Inventário S3
- Replicação do S3
- Etiquetas de objeto
- S3 Select
- Logs de acesso ao servidor

- Hospedagem de site estático
- Lente de Armazenamento do S3
- Grupos da Lente de Armazenamento do S3
- Aceleração de Transferências do S3
- Criptografia de camada dupla do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (DSSE-KMS)
- 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)
- A opção de copiar as configurações de um bucket existente ao criar um bucket no AWS Management Console

Conceitos básicos da classe S3 Express One Zone

A seção a seguir descreve como começar a usar a classe de armazenamento Amazon S3 Express One Zone e os buckets de diretório. Para ter mais informações, consulte [O que é a classe S3 Express One Zone?](#)

Tópicos

- [Configurar o AWS Identity and Access Management \(IAM\) com a classe S3 Express One Zone](#)
- [Configurar endpoints de VPC do gateway](#)
- [Trabalhar com a classe S3 Express One Zone usando o console do S3, AWS CLI e os AWS SDKs](#)

Configurar o AWS Identity and Access Management (IAM) com a classe S3 Express One Zone

O AWS Identity and Access Management (IAM) é um AWS service (Serviço da AWS) que ajuda os administradores a controlar o acesso aos recursos da AWS de forma segura. Os administradores do IAM controlam quem pode se autenticar (se conectar) e ter autorização (permissões) para utilizar os recursos do Amazon S3 na classe S3 Express One Zone. Você pode usar o IAM sem custo adicional.

Por padrão, os usuários não têm permissões para os buckets de diretório e as operações da classe S3 Express One Zone. Para conceder permissões de acesso a buckets de diretório e operações

da classe S3 Express One Zone, você pode usar o IAM para criar usuários ou perfis e anexar permissões a essas identidades.

Para começar a usar o IAM, consulte [AWS Identity and Access Management \(IAM\) para a classe S3 Express One Zone](#) e [Políticas baseadas em identidade do IAM para a classe S3 Express One Zone](#).

Configurar endpoints de VPC do gateway

Para acessar a classe S3 Express One Zone, use endpoints regionais e zonais que são diferentes dos endpoints padrão do Amazon S3. Dependendo da operação de API do Amazon S3 usada, será necessário um endpoint regional ou zonal. Para obter uma lista completa das operações de API compatíveis por tipo de endpoint, consulte [Operações de API compatíveis com a classe S3 Express One Zone](#). Você deve acessar endpoints zonais e regionais por meio de um endpoint de nuvem privada virtual (VPC) do gateway. Para configurar endpoints de gateway, consulte [Redes para a classe S3 Express One Zone](#).

Trabalhar com a classe S3 Express One Zone usando o console do S3, AWS CLI e os AWS SDKs

Você pode trabalhar com a classe de armazenamento S3 Express One Zone e os buckets de diretório usando os AWS SDKs, o console do Amazon S3, a AWS CLI (AWS Command Line Interface) e a API REST do Amazon S3.

Console do S3

Para começar a usar o console do S3, siga estas etapas:

- [Criar um bucket de diretório](#)
- [Esvaziar um bucket de diretório](#)
- [Excluir um bucket de diretório](#)

SDKs da AWS

A classe S3 Express One Zone é compatível com os seguintes AWS SDKs:

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java 2.x

- [AWS SDK for JavaScript v3](#)
- [AWS SDK for .NET](#)
- [AWS SDK for PHP](#)
- [AWS SDK for Python \(Boto3\)](#)
- [AWS SDK for Ruby](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK para Rust](#)

Ao trabalhar com a classe S3 Express One Zone, recomendamos usar a versão mais recente dos AWS SDKs. Os AWS SDKs compatíveis com a classe S3 Express One Zone lidam com o estabelecimento, a atualização e o encerramento de sessões em seu nome. Isso significa que você pode começar imediatamente a usar as operações de API depois de fazer download e instalar os AWS SDKs e configurar as permissões necessárias do IAM. Para ter mais informações, consulte [AWS Identity and Access Management \(IAM\) para a classe S3 Express One Zone](#).

Para obter informações sobre os AWS SDKs, inclusive sobre como baixá-los e instalá-los, consulte [Aprenda a criar na AWS](#).

Para obter exemplos de AWS SDK, consulte os seguintes tópicos:

- [Criar um bucket de diretório](#)
- [Esvaziar um bucket de diretório](#)
- [Excluir um bucket de diretório](#)

AWS Command Line Interface (AWS CLI)

Você pode usar a AWS Command Line Interface (AWS CLI) para criar buckets de diretório e usar as operações de API de endpoint regional e zonal compatíveis com a classe S3 Express One Zone.

Para começar a usar a AWS CLI, consulte [Conceitos básicos da AWS CLI](#) na Referência de comandos da AWS CLI.

Note

Para usar buckets de diretório com os [comandos aws s3 de alto nível](#), atualize a AWS CLI para a versão mais recente. Para obter mais informações sobre como instalar e configurar

a AWS CLI, consulte [Instalar ou atualize para a versão mais recente da AWS CLI](#) na Referência de comandos da AWS CLI.

Para obter exemplos da AWS CLI, consulte os seguintes tópicos:

- [Criar um bucket de diretório](#)
- [Esvaziar um bucket de diretório](#)
- [Excluir um bucket de diretório](#)

Redes para a classe S3 Express One Zone

Para acessar objetos da classe de armazenamento Amazon S3 Express One Zone e buckets de diretório, use endpoints de API regionais e zonais que são diferentes dos endpoints padrão do Amazon S3. Dependendo da operação de API do S3 usada, será necessário um endpoint regional ou zonal. Para obter uma lista completa das operações de API por tipo de endpoint, consulte [Operações de API compatíveis com a classe S3 Express One Zone](#).

Você pode acessar operações de API zonais e regionais por meio de endpoints de nuvem privada virtual (VPC) de gateway. Para configurar endpoints de VPC do gateway, consulte [the section called “Configurar endpoints de gateway da VPC”](#).

Os tópicos a seguir descrevem os requisitos de rede para acessar a classe S3 Express One Zone usando um endpoint da VPC do gateway.

Tópicos

- [Endpoints](#)
- [Configurar endpoints de gateway da VPC](#)

Endpoints

É possível acessar objetos da classe de armazenamento Amazon S3 Express One Zone e buckets de diretório pela VPC usando endpoints de VPC do gateway. A classe S3 Express One Zone usa endpoints de API regionais e zonais. Dependendo da operação de API do Amazon S3 usada, será necessário um endpoint regional ou zonal. Não há cobrança adicional pelo uso de endpoints do gateway.

As operações de API em nível de bucket (ou ambiente de gerenciamento) estão disponíveis por meio de endpoints regionais e são chamadas de operações de API do endpoint regional. Exemplos de operações de API de endpoint regional são `CreateBucket` e `DeleteBucket`. Ao criar um bucket de diretório, escolha uma única zona de disponibilidade em que o bucket de diretório será criado. Depois de criar um bucket de diretório, você pode usar operações de API de endpoint zonal para fazer upload e gerenciar os objetos no bucket de diretório.

As operações de API em nível de objeto (ou plano de dados) estão disponíveis por meio de endpoints zonais e são chamadas de operações de API do endpoint zonal. Exemplos de operações de API de endpoint zonal são `CreateSession` e `PutObject`.

A tabela a seguir mostra os endpoints de API regionais e zonais que estão disponíveis para cada região e zona de disponibilidade.

Configurar endpoints de gateway da VPC

Use o procedimento a seguir para criar um endpoint de gateway que se conecte aos objetos da classe de armazenamento Amazon S3 Express One Zone e aos buckets de diretório.

Como configurar um endpoint da VPC do gateway

1. Abra o console do Amazon VPC em <https://console.aws.amazon.com/vpc/>.
2. No painel de navegação, escolha Endpoints.
3. Escolha Criar endpoint.
4. Crie um nome para o endpoint.
5. Em Service category (Categoria de serviço), escolha Serviços da AWS.
6. Em Serviços, adicione o filtro Tipo=Gateway e escolha o botão de opções ao lado de `com.amazonaws.região.s3express`.
7. Em VPC, escolha a VPC na qual deseja criar o endpoint.
8. Em Route tables (Tabelas de rotas), selecione as tabelas de rotas a serem usadas pelo endpoint. O Amazon VPC adiciona automaticamente uma rota que aponta o tráfego destinado ao serviço para a interface de rede do endpoint.
9. Em Política, escolha Acesso total para permitir todas as operações de todas as entidades principais em todos os recursos no endpoint da VPC. Como alternativa, escolha Personalizado para anexar uma política de endpoint da VPC que controle as permissões das entidades principais para realizar ações em recursos pelo endpoint da VPC.

10. (Opcional) Para adicionar uma etiqueta, escolha Adicionar nova tag e insira a chave e o valor da etiqueta.
11. Escolha Criar endpoint.

Depois de criar um endpoint de gateway, você pode usar endpoints de API regionais e endpoints de API zonais para acessar objetos da classe de armazenamento Amazon S3 Express One Zone e buckets de diretório.

Buckets de diretório

Há dois tipos de buckets do Amazon S3: buckets de uso geral e buckets de diretório. Escolha o tipo de bucket que melhor se adapte aos requisitos de performance e da aplicação:

- Os buckets de uso geral são o tipo original de bucket do S3 e são recomendados para a maioria dos casos de uso e padrões de acesso. Os buckets de uso geral também permitem que objetos sejam armazenados em todas as classes de armazenamento, exceto na classe S3 Express One Zone.
- Os buckets de diretório usam a classe de armazenamento S3 Express One Zone, que é recomendada se sua aplicação é sensível à performance e se beneficia de latências inferiores a dez milissegundos para PUT e GET.

Os buckets de diretório são usados para workloads ou aplicações críticas à performance que requerem latência consistente inferior a dez milissegundos. Os buckets de diretório organizam os dados de forma hierárquica em diretórios, ao contrário da estrutura de armazenamento plana dos buckets de uso geral. Não há limites de prefixo para os buckets de diretório, e diretórios individuais podem ser escalados horizontalmente.

Os buckets de diretório usam a classe de armazenamento S3 Express One Zone, que armazena os dados em vários dispositivos dentro de uma única zona de disponibilidade, mas não os armazena de forma redundante entre zonas de disponibilidade. Ao criar um bucket de diretório, recomendamos que você especifique uma Região da AWS e uma zona de disponibilidade que seja local para suas instâncias computacionais do Amazon EC2, do Amazon Elastic Kubernetes Service ou do Amazon Elastic Container Service (Amazon ECS) a fim de otimizar a performance.

Você pode criar até dez buckets de diretório em cada Contas da AWS, sem limite no número de objetos que podem ser armazenados em um bucket. A cota de buckets é aplicada a cada região

na Conta da AWS. Se sua aplicação precisar aumentar esse limite, entre em contato com o AWS Support. Para obter mais informações, acesse o [console do Service Quotas](#).

Important

Os buckets de diretório que não tiverem nenhuma atividade de solicitação por um período de pelo 90 dias passarão para um estado inativo. Enquanto estiver inativo, um bucket de diretório ficará temporariamente inacessível para leituras e gravações. Os buckets inativos retêm todo o armazenamento, os metadados de objetos e os metadados de bucket. As cobranças de armazenamento existentes se aplicam aos buckets inativos. Se você fizer uma solicitação de acesso a um bucket inativo, o bucket passará para um estado ativo, normalmente em alguns minutos. Durante esse período de transição, as leituras e gravações retornarão um código de erro HTTP 503 (Service Unavailable).

Os tópicos a seguir fornecem informações sobre buckets de diretório. Para obter mais informações sobre os buckets de uso geral, consulte [Visão geral dos buckets](#).

Tópicos

- [Zonas de disponibilidade](#)
- [Nomes de bucket de diretório](#)
- [Diretórios](#)
- [Nomes de chave](#)
- [Gerenciamento de acesso](#)
- [Trabalhar com buckets de diretório](#)
- [Regras de nomenclatura de buckets de diretório](#)
- [Criar um bucket de diretório](#)
- [Visualizar as propriedades de buckets de diretório](#)
- [Gerenciar políticas de bucket para buckets de diretório](#)
- [Esvaziar um bucket de diretório](#)
- [Excluir um bucket de diretório](#)
- [Listar buckets de diretório](#)
- [Usar HeadBucket com buckets de diretório](#)

Zonas de disponibilidade

Ao criar um bucket de diretório, escolha a zona de disponibilidade e a Região da AWS.

Os buckets de diretório usam a classe de armazenamento S3 Express One Zone, desenvolvida para ser usada por aplicações sensíveis à performance. A classe S3 Express One Zone é a primeira classe de armazenamento do S3 em que é possível selecionar uma única zona de disponibilidade com a opção de manter o armazenamento de objetos e os recursos de computação na mesma localização, o que fornece a maior velocidade de acesso possível.

Com a classe S3 Express One Zone, os dados são armazenados de forma redundante em vários dispositivos dentro de uma única zona de disponibilidade. A classe S3 Express One Zone foi projetada para oferecer disponibilidade de 99,95% em uma única zona de disponibilidade e tem o respaldo do [Acordo de Serviço do Amazon S3](#). Para ter mais informações, consulte [Zona de disponibilidade única](#).

Nomes de bucket de diretório

Um nome de bucket de diretório consiste em um nome base que você fornece e um sufixo que contém o ID da zona de disponibilidade em que o bucket está localizado. Os nomes de bucket de diretório devem usar o seguinte formato e seguir as regras de nomenclatura para buckets de diretório:

```
bucket-base-name--azid--x-s3
```

Por exemplo, o seguinte nome de bucket de diretório contém o ID da zona de disponibilidade usw2-az1:

```
bucket-base-name--usw2-az1--x-s3
```

Para ter mais informações, consulte [Regras de nomenclatura de buckets de diretório](#).

Diretórios

Os buckets de diretório organizam os dados de forma hierárquica em diretórios, ao contrário da estrutura de organização plana dos buckets de uso geral. Cada bucket de diretório do S3 oferece suporte a centenas de milhares de transações por segundo (TPS), independentemente do número de diretórios dentro dele.

Com um namespace hierárquico, o delimitador na chave de objeto é importante. O único delimitador compatível é a barra (/). Os diretórios são determinados pelos limites de delimitador. Por exemplo, a chave de objeto `dir1/dir2/file1.txt` resulta na criação automática dos diretórios `dir1/` e `dir2/`, além da adição do objeto `file1.txt` ao diretório `/dir2` no caminho `dir1/dir2/file1.txt`.

O modelo de indexação de buckets de diretório retorna resultados não organizados para a operação de API `ListObjectsV2`. Se precisar limitar os resultados a uma subseção do bucket, você poderá especificar um caminho de subdiretório no parâmetro `prefix`, como `prefix=dir1/`.

Nomes de chave

Para buckets de diretório, os subdiretórios que são comuns a várias chaves de objeto são criados com a primeira chave de objeto. As chaves de objeto adicionais para o mesmo subdiretório usam o subdiretório criado anteriormente. Esse modelo oferece flexibilidade na escolha de chaves de objeto mais adequadas à aplicação, com o mesmo suporte a diretórios esparsos e densos.

Gerenciamento de acesso

Os buckets de diretório têm todas as configurações da funcionalidade Bloqueio de Acesso Público do S3 habilitadas por padrão no nível do bucket. A Propriedade de Objetos do S3 está definida como Imposto pelo proprietário do bucket e as listas de controle de acesso (ACLs) estão desabilitadas. Essas configurações não podem ser modificadas.

Por padrão, os usuários não têm permissões para os buckets de diretório e as operações da classe S3 Express One Zone. Para conceder permissões de acesso a buckets de diretório, você pode usar o IAM para criar usuários ou perfis e anexar permissões a essas identidades. Para obter mais informações, consulte [AWS Identity and Access Management \(IAM\) para a classe S3 Express One Zone](#).

Trabalhar com buckets de diretório

Para obter mais informações sobre como trabalhar com buckets de diretório, consulte os tópicos a seguir.

Tópicos

- [Regras de nomenclatura de buckets de diretório](#)
- [Criar um bucket de diretório](#)

- [Visualizar as propriedades de buckets de diretório](#)
- [Gerenciar políticas de bucket para buckets de diretório](#)
- [Esvaziar um bucket de diretório](#)
- [Excluir um bucket de diretório](#)
- [Listar buckets de diretório](#)
- [Usar HeadBucket com buckets de diretório](#)

Regras de nomenclatura de buckets de diretório

Ao criar um bucket de diretório no Amazon S3, as regras de nomenclatura de buckets a seguir se aplicam. Para conferir as regras de nomenclatura para buckets de uso geral, consulte [Regras de nomeação de bucket](#).

Um nome de bucket de diretório consiste em um nome base que você fornece e um sufixo que contém o ID da zona de disponibilidade AWS em que o bucket está localizado e `--x-s3`.

```
base-name--azid--x-s3
```

Por exemplo, o seguinte nome de bucket de diretório contém o ID da zona de disponibilidade `usw2-az1`:

```
bucket-base-name--usw2-az1--x-s3
```

Note

Quando você cria um bucket de diretório usando o console, um sufixo é adicionado automaticamente ao nome base que você fornece. Esse sufixo inclui o ID da zona de disponibilidade que você escolheu.

Ao criar um bucket de diretório usando uma API, você deve fornecer o sufixo completo, incluindo o ID da zona de disponibilidade, na solicitação. Consulte uma lista de IDs de zona de disponibilidade em [Zonas de disponibilidade e regiões da classe S3 Express One Zone](#).

Os nomes dos buckets de diretório devem:

- Ser únicos dentro da Região da AWS e da zona de disponibilidade escolhidas.

- Ter entre 3 e 63 caracteres, incluindo o sufixo.
- Consistir em letras minúsculas, números e hifens (-).
- Inicie e termine com uma letra ou um número.
- Deve incluir o seguinte sufixo: `--azid--x-s3`.

Criar um bucket de diretório

Para começar a usar a classe de armazenamento Amazon S3 Express One Zone, crie um bucket de diretório. A classe de armazenamento S3 Express One Zone só pode ser usada com buckets de diretório. A classe de armazenamento S3 Express One Zone oferece suporte a casos de uso de baixa latência e fornece processamento de dados mais rápido em uma única zona de disponibilidade. Se sua aplicação é sensível à performance e se beneficia de latências inferiores a dez milissegundos para PUT e GET, recomendamos criar um bucket de diretório para que possa usar a classe de armazenamento S3 Express One Zone.

Há dois tipos de buckets do Amazon S3: buckets de uso geral e buckets de diretório. Você deve escolher o tipo de bucket que melhor se adapte aos requisitos de desempenho e da aplicação. Os buckets de uso geral são o tipo original de bucket do S3. Os buckets de uso geral são recomendados para a maioria dos casos de uso e padrões de acesso, permitindo que objetos sejam armazenados em todas as classes de armazenamento, exceto na classe S3 Express One Zone. Para obter mais informações sobre os buckets de uso geral, consulte [Visão geral dos buckets](#).

Os buckets de diretório usam a classe de armazenamento S3 Express One Zone, projetada para workloads ou aplicações de performance crítica que exigem latência consistente inferior a dez milissegundos. A classe S3 Express One Zone é a primeira classe de armazenamento do S3 em que é possível selecionar uma única zona de disponibilidade com a opção de manter o armazenamento de objetos e os recursos de computação na mesma localização, o que fornece a maior velocidade de acesso possível. Ao criar um bucket de diretório, você tem a opção de especificar uma Região da AWS e uma zona de disponibilidade que seja local para suas instâncias computacionais do Amazon EC2, do Amazon Elastic Kubernetes Service ou do Amazon Elastic Container Service (Amazon ECS) a fim de otimizar a performance.

Com a classe S3 Express One Zone, os dados são armazenados de forma redundante em vários dispositivos dentro de uma única zona de disponibilidade. A classe S3 Express One Zone foi projetada para oferecer disponibilidade de 99,95% em uma única zona de disponibilidade e tem o respaldo do [Acordo de Serviço do Amazon S3](#). Para ter mais informações, consulte [Zona de disponibilidade única](#).

Os buckets de diretório organizam os dados de forma hierárquica em diretórios, ao contrário da estrutura de armazenamento plana dos buckets de uso geral. Não há limites de prefixo para os buckets de diretório, e diretórios individuais podem ser escalados horizontalmente.

Para obter mais informações sobre os buckets de diretório, consulte [Buckets de diretório](#).

Nomes de bucket de diretório

Os nomes de bucket de diretório devem seguir o seguinte formato e estar em conformidade com as regras de nomenclatura para buckets de diretório:

```
bucket-base-name--azid--x-s3
```

Por exemplo, o seguinte nome de bucket de diretório contém o ID da zona de disponibilidade usw2-az1:

```
bucket-base-name--usw2-az1--x-s3
```

Para obter mais informações sobre as regras de nomenclatura para buckets de diretório, consulte [Regras de nomenclatura de buckets de diretório](#).

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 barra de navegação na parte superior da página, escolha o nome da Região da AWS exibida no momento. Em seguida, escolha a região na qual você deseja criar um bucket.


Note

Para reduzir a latência e os custos e atender aos requisitos regulatórios, selecione uma região perto de você. Os objetos armazenados em uma região nunca saem dessa região, a menos que você os transfira para outra região. Para obter uma lista de Regiões da AWS do Amazon S3, consulte [Endpoints de AWS service \(Serviço da AWS\)](#) no Referência geral da Amazon Web Services.

3. No painel de navegação à esquerda, escolha Buckets.
4. Selecione Create bucket (Criar bucket).


A página Create bucket (Criar bucket) é aberta.

5. Em Configuração geral, visualize a Região da AWS onde o bucket será criado.
6. Em Tipo de bucket, escolha Diretório.

 Note


- Se você escolher uma região que não oferece suporte a buckets de diretório, a opção Tipo de bucket desaparecerá e o tipo de bucket assumirá um bucket de uso geral como padrão. Para criar um bucket de diretório, escolha uma região compatível. Para obter uma lista de regiões que oferecem suporte a buckets de diretório e à classe de armazenamento Amazon S3 Express One Zone, consulte [the section called “Zonas de disponibilidade e regiões da classe S3 Express One Zone”](#).
- Depois de criar o bucket, não será possível alterar o tipo.

Em Zona de disponibilidade, escolha uma zona de disponibilidade que seja local para os seus serviços de computação. Para obter uma lista de zonas de disponibilidade que oferecem suporte a buckets de diretório e à classe de armazenamento S3 Express One Zone, consulte [the section called “Zonas de disponibilidade e regiões da classe S3 Express One Zone”](#).

 Note

A zona de disponibilidade não poderá ser alterada após a criação do bucket.

7. Em Zona de disponibilidade, marque a caixa de seleção para indicar que, em caso de interrupção na zona de disponibilidade, seus dados poderão ficar indisponíveis ou ser perdidos.

 Important

Embora os buckets de diretório sejam armazenados em vários dispositivos dentro de uma única zona de disponibilidade, eles não armazenam dados de forma redundante entre zonas de disponibilidade.


8. Em Nome do bucket, insira um nome para o bucket de diretório.

Os nomes dos buckets de diretório devem:

- Ser únicos dentro da Região da AWS e da zona de disponibilidade escolhidas.
- Ter entre 3 e 63 caracteres, incluindo o sufixo.
- Consistir em letras minúsculas, números e hifens (-).
- Inicie e termine com uma letra ou um número.
- Deve incluir o seguinte sufixo: `--azid--x-s3`.

Um sufixo é adicionado automaticamente ao nome de base que você fornece ao criar um bucket de diretório usando o console. Esse sufixo inclui o ID da zona de disponibilidade que você escolheu.

Depois de criado o bucket, você não pode mudar seu nome. Para obter mais informações sobre como nomear buckets, consulte [Regras de nomeação de bucket](#).

 Important

Não inclua 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.

9. Em Propriedade do objeto, a configuração Imposto pelo proprietário do bucket está habilitada automaticamente e todas as listas de controle de acesso (ACLs) estão desabilitadas. Para os buckets de diretório, não é possível habilitar ACLs.

ACLs desabilitadas

- Imposto pelo proprietário do bucket (padrão): 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 de acesso aos dados no bucket do S3. O bucket usa políticas exclusivamente para definir o controle de acesso.

A maioria dos casos de uso modernos no Amazon S3 não exige mais o uso de ACLs. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

10. Em Configurações de bloqueio do acesso público deste bucket, todas as configurações da funcionalidade Bloqueio de Acesso Público para o bucket de diretório estão automaticamente habilitadas. Não é possível modificar essas configurações para buckets de diretório. Para obter

mais informações sobre como bloquear o acesso público, consulte [Bloquear o acesso público ao armazenamento do Amazon S3](#).

11. Em Configurações de criptografia no lado do servidor, o Amazon S3 aplica a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) como o nível básico de criptografia para todos os buckets do S3. Todos os uploads de objetos para buckets de diretório são criptografados com SSE-S3. Para buckets de diretório, não é possível modificar o tipo de criptografia. Para ter mais informações sobre SSE-S3, consulte [the section called “Chaves de criptografia gerenciadas pelo Amazon S3 \(SSE-S3\)”](#).

12. Selecione Criar bucket.

Depois de criar o bucket, você pode adicionar arquivos e pastas a ele. Para ter mais informações, consulte [the section called “Trabalhar com objetos em um bucket de diretório”](#).

Uso da SDKs AWS

SDK for Go

Este exemplo mostra como criar um bucket de diretório usando o AWS SDK for Go.

Example

```
var bucket = "..."  
  
func runCreateBucket(c *s3.Client) {  
    resp, err := c.CreateBucket(context.Background(), &s3.CreateBucketInput{  
        Bucket: &bucket,  
        CreateBucketConfiguration: &types.CreateBucketConfiguration{  
            Location: &types.LocationInfo{  
                Name: aws.String("usw2-az1"),  
                Type: types.LocationTypeAvailabilityZone,  
            },  
            Bucket: &types.BucketInfo{  
                DataRedundancy: types.DataRedundancySingleAvailabilityZone,  
                Type: types.BucketTypeDirectory,  
            },  
        },  
    })  
    var terr *types.BucketAlreadyOwnedByYou  
    if errors.As(err, &terr) {  
        fmt.Printf("BucketAlreadyOwnedByYou: %s\n", aws.ToString(terr.Message))  
        fmt.Printf("noop...\n")  
    }  
}
```



```

        return
    }
    if err != nil {
        log.Fatal(err)
    }

    fmt.Printf("bucket created at %s\n", aws.ToString(resp.Location))
}

```

SDK for Java 2.x

Este exemplo mostra como criar um bucket de diretório usando o AWS SDK for Java 2.x.

Example

```

public static void createBucket(S3Client s3Client, String bucketName) {

    //Bucket name format is {base-bucket-name}--{az-id}--x-s3
    //example: doc-example-bucket--usw2-az1--x-s3 is a valid name for a directory
    bucket created in
    //Region us-west-2, Availability Zone 2

    CreateBucketConfiguration bucketConfiguration =
    CreateBucketConfiguration.builder()
        .location(LocationInfo.builder()
            .type(LocationType.AVAILABILITY_ZONE)
            .name("usw2-az1").build()) //this must match the Region and
    Availability Zone in your bucket name
        .bucket(BucketInfo.builder()
            .type(BucketType.DIRECTORY)
            .dataRedundancy(DataRedundancy.SINGLE_AVAILABILITY_ZONE)
            .build()).build();

    try {

        CreateBucketRequest bucketRequest =
    CreateBucketRequest.builder().bucket(bucketName).createBucketConfiguration(bucketConfigurat
        CreateBucketResponse response = s3Client.createBucket(bucketRequest);
        System.out.println(response);
    }

    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

```

```
    }  
  }
```

AWS SDK for JavaScript

Este exemplo mostra como criar um bucket de diretório usando o AWS SDK for JavaScript.

Example

```
// file.mjs, run with Node.js v16 or higher  
// To use with the preview build, place this in a folder  
// inside the preview build directory, such as /aws-sdk-js-v3/workspace/  
  
import { S3 } from "@aws-sdk/client-s3";  
  
const region = "us-east-1";  
const zone = "use1-az4";  
const suffix = `${zone}--x-s3`;  
  
const s3 = new S3({ region });  
  
const bucketName = `...--${suffix}`;  
  
const createResponse = await s3.createBucket(  
  { Bucket: bucketName,  
    CreateBucketConfiguration: {Location: {Type: "AvailabilityZone", Name: zone},  
      Bucket: { Type: "Directory", DataRedundancy: "SingleAvailabilityZone" }}  
  )  
);
```

AWS SDK for .NET

Este exemplo mostra como criar um bucket de diretório usando o AWS SDK for .NET.

Example

```
using (var amazonS3Client = new AmazonS3Client())  
{  
    var putBucketResponse = await amazonS3Client.PutBucketAsync(new PutBucketRequest  
    {
```

```

    BucketName = "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
    PutBucketConfiguration = new PutBucketConfiguration
    {
        BucketInfo = new BucketInfo { DataRedundancy =
DataRedundancy.SingleAvailabilityZone, Type = BucketType.Directory },
        Location = new LocationInfo { Name = "usw2-az1", Type =
LocationType.AvailabilityZone }
    }
}).ConfigureAwait(false);
}

```

SDK for PHP

Este exemplo mostra como criar um bucket de diretório usando o AWS SDK for PHP.

Example

```

require 'vendor/autoload.php';

$s3Client = new S3Client([

    'region'      => 'us-east-1',
]);

$result = $s3Client->createBucket([
    'Bucket' => 'doc-example-bucket--use1-az4--x-s3',
    'CreateBucketConfiguration' => [
        'Location' => ['Name'=> 'use1-az4', 'Type'=> 'AvailabilityZone'],
        'Bucket' => ["DataRedundancy" => "SingleAvailabilityZone" ,"Type" =>
"Directory"]    ],
]);

```

SDK for Python

Este exemplo mostra como criar um bucket de diretório usando o AWS SDK for Python (Boto3).

Example

```

import logging
import boto3
from botocore.exceptions import ClientError

```

```
def create_bucket(s3_client, bucket_name, availability_zone):
    '''
    Create a directory bucket in a specified Availability Zone

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket to create; for example, 'doc-example-bucket--usw2-az1--x-s3'
    :param availability_zone: String; Availability Zone ID to create the bucket in,
    for example, 'usw2-az1'
    :return: True if bucket is created, else False
    '''

    try:
        bucket_config = {
            'Location': {
                'Type': 'AvailabilityZone',
                'Name': availability_zone
            },
            'Bucket': {
                'Type': 'Directory',
                'DataRedundancy': 'SingleAvailabilityZone'
            }
        }
        s3_client.create_bucket(
            Bucket = bucket_name,
            CreateBucketConfiguration = bucket_config
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'us-west-2'
    availability_zone = 'usw2-az1'
    s3_client = boto3.client('s3', region_name = region)
    create_bucket(s3_client, bucket_name, availability_zone)
```

SDK for Ruby

Este exemplo mostra como criar um bucket de diretório usando o AWS SDK for Ruby.

Example

```
s3 = Aws::S3::Client.new(region:'us-west-2')
s3.create_bucket(
  bucket: "bucket_base_name--az_id--x-s3",
  create_bucket_configuration: {
    location: { name: 'usw2-az1', type: 'AvailabilityZone' },
    bucket: { data_redundancy: 'SingleAvailabilityZone', type: 'Directory' }
  }
)
```

Usando a AWS CLI

Este exemplo mostra como criar um bucket de diretório usando o AWS CLI. Para usar o comando, substitua os *espaços reservados para entrada do usuário* por suas próprias informações.

Ao criar um bucket de diretório, você deve fornecer detalhes de configuração e usar a seguinte convenção de nomenclatura: *bucket-base-name--azid--x-s3*.

```
aws s3api create-bucket
--bucket bucket-base-name--azid--x-s3
--create-bucket-configuration 'Location={Type=AvailabilityZone,Name=usw2-az1},Bucket={DataRedundancy=SingleAvailabilityZone,Type=Directory}'
--region us-west-2
```

Para obter mais informações, consulte [create-bucket](#) na AWS Command Line Interface.

Visualizar as propriedades de buckets de diretório

Você pode visualizar e configurar as propriedades de um bucket de diretório do Amazon S3 usando o console do Amazon S3. Para obter mais informações, consulte [Buckets de diretório](#) e [O que é a classe S3 Express One Zone?](#)

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 Buckets.
3. Escolha a guia Buckets de diretórios.

4. Na lista Buckets de diretórios, escolha o nome do bucket cujas propriedades deseja visualizar.
5. Escolha a guia Properties (Propriedades).
6. Na guia Propriedades, você pode visualizar as seguintes propriedades do bucket:
 - Visão geral do bucket de diretório: você pode ver a Região da AWS, a zona de disponibilidade, o nome do recurso da Amazon (ARN) e a data de criação do bucket.
 - Criptografia padrão: o Amazon S3 aplica a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) como o nível básico de criptografia para todos os buckets do S3. Para buckets de diretório, essa configuração não pode ser modificada. O Amazon S3 criptografa um objeto antes de salvá-lo em um disco e descriptografa o objeto ao baixá-lo. Para ter mais informações, consulte [Definir o comportamento padrão da criptografia para os buckets do Amazon S3](#).

Para obter mais informações sobre os recursos compatíveis com buckets de diretório, consulte [Recursos da classe S3 Express One Zone](#).

Gerenciar políticas de bucket para buckets de diretório

Você pode adicionar, excluir, atualizar e visualizar políticas de bucket para buckets de diretório do Amazon S3 usando o console do Amazon S3 e os AWS SDKs. Para obter mais informações, consulte os tópicos a seguir. Para obter mais informações sobre as chaves de condição e ações do AWS Identity and Access Management (IAM) compatíveis com a classe S3 Express One Zone, consulte [AWS Identity and Access Management \(IAM\) para a classe S3 Express One Zone](#). Para conferir exemplos de políticas para buckets de diretório, consulte [Exemplos de políticas de bucket de diretório para a classe S3 Express One Zone](#).

Tópicos

- [Adicionar uma política de bucket](#)
- [Visualizar uma política de bucket](#)
- [Excluir uma política de bucket](#)

Adicionar uma política de bucket

Para adicionar uma política de bucket a um bucket de diretório, você pode usar o console do Amazon S3, os SDKs da AWS ou a AWS CLI.

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 Buckets.
3. Escolha a guia Buckets de diretórios.
4. Na lista Buckets de diretórios, escolha o nome do bucket no qual você deseja fazer upload de pastas ou arquivos.
5. Escolha a aba Permissões.
6. Em Bucket policy (Política de bucket), escolha Edit (Editar). A página Edit bucket policy (Editar política de bucket) é exibida.
7. Para gerar uma política automaticamente, escolha Gerador de políticas.

Se você escolher Policy generator (Gerador de políticas), o Gerador de Políticas da AWS será aberto em uma nova janela.

Se você não quiser usar o gerador de políticas da AWS, poderá adicionar ou editar instruções JSON na seção Política.


- a. Na página Gerador de Políticas da AWS, em Selecionar tipo de política, escolha Política de bucket do S3.
- b. Adicione uma instrução inserindo as informações nos campos fornecidos e escolha Add Statement (Adicionar instrução). Repita esta etapa para todas as instruções que quiser adicionar. Para ter 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 Editar política de bucket exibe o ARN do bucket (nome do recurso da Amazon) do bucket atual acima do campo de texto 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.
8. Na caixa Policy (Política), edite a política existente ou cole a política de bucket do Gerador de Políticas da AWS. Solucione avisos de segurança, erros, avisos gerais e sugestões antes de salvar a política.

 Note

As políticas de bucket são limitadas a 20 KB.

9. Escolha Save changes (Salvar alterações), o que levará você de volta à guia Permissions (Permissões).

Uso dos AWS SDKs

SDK for Java 2.x

Example

PutBucketPolicy AWS SDK for Java 2.x

```
public static void setBucketPolicy(S3Client s3Client, String bucketName, String
policyText) {

    //sample policy text
    /**
     * policy_statement = {
     *     'Version': '2012-10-17',
     *     'Statement': [
     *         {
     *             'Sid': 'AdminPolicy',
     *             'Effect': 'Allow',
     *             'Principal': {
     *                 "AWS": "111122223333"
     *             },
     *             'Action': 's3express:*',
     *             'Resource':
     * 'arn:aws:s3express:region:111122223333:bucket/bucket-base-name--azid--x-s3'
     *         }
     *     ]
     * }
    */
}
```



```
*/
System.out.println("Setting policy:");
System.out.println("----");
System.out.println(policyText);
System.out.println("----");
System.out.format("On Amazon S3 bucket: \"%s\"\n", bucketName);

try {
    PutBucketPolicyRequest policyReq = PutBucketPolicyRequest.builder()
        .bucket(bucketName)
        .policy(policyText)
        .build();
    s3Client.putBucketPolicy(policyReq);
    System.out.println("Done!");
}

catch (S3Exception e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
}
```

Usando a AWS CLI

Este exemplo mostra como adicionar uma política de bucket a um bucket de diretório usando a AWS CLI. Para usar o comando, substitua os *espaços reservados para entrada do usuário* por suas próprias informações.

```
aws s3api put-bucket-policy --bucket bucket-base-name--azid--x-s3 --policy file://
bucket_policy.json
```

bucket_policy.json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AdminPolicy",
      "Effect": "Allow",
      "Principal": {
        "AWS": "111122223333"
      }
    }
  ]
}
```

```
    },
    "Action": "s3express*",
    "Resource": "arn:aws:s3express:us-west-2:111122223333:bucket/bucket-name--usw2-az1--x-s3"
  }
]
```

Para obter mais informações, consulte [put-bucket-policy](#) na AWS Command Line Interface.

Visualizar uma política de bucket

Para visualizar uma política de um bucket de diretório, use os exemplos a seguir.

Usando a AWS CLI

Este exemplo mostra como visualizar a política de bucket anexada a um bucket de diretório usando a AWS CLI. Para usar o comando, substitua os *espaços reservados para entrada do usuário* por suas próprias informações.

```
aws s3api get-bucket-policy --bucket bucket-base-name--azid--x-s3
```

Para obter mais informações, consulte [get-bucket-policy](#) na AWS Command Line Interface.

Excluir uma política de bucket

Para excluir uma política de um bucket de diretório, use os exemplos a seguir.

Uso dos AWS SDKs

SDK for Java 2.x

Example

DeleteBucketPolicy AWS SDK for Java 2.x

```
public static void deleteBucketPolicy(S3Client s3Client, String bucketName) {
    try {
        DeleteBucketPolicyRequest deleteBucketPolicyRequest =
        DeleteBucketPolicyRequest
            .builder()
            .bucket(bucketName)
```

```
        .build()
        s3Client.deleteBucketPolicy(deleteBucketPolicyRequest);
        System.out.println("Successfully deleted bucket policy");
    }

    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

Usando a AWS CLI

Este exemplo mostra como excluir uma política de bucket de um bucket de diretório usando a AWS CLI. Para usar o comando, substitua os *espaços reservados para entrada do usuário* por suas próprias informações.

```
aws s3api delete-bucket-policy --bucket bucket-base-name--azid--x-s3
```

Para obter mais informações, consulte [delete-bucket-policy](#) na AWS Command Line Interface.

Esvaziar um bucket de diretório

Você pode esvaziar um bucket de diretório do Amazon S3 usando o console do Amazon S3. Para obter mais informações sobre os buckets de diretório, consulte [Buckets de diretório](#).

Antes de esvaziar um bucket de diretório, observe o seguinte:

- Ao esvaziar um bucket de diretório, você exclui todos os objetos, mas mantém o bucket.
- Não é possível desfazer a ação de esvaziar um bucket de diretório.
- Os objetos que forem adicionados ao bucket de diretório enquanto a ação de esvaziar estiver em andamento também poderão ser excluídos.

Se você também quiser excluir o bucket, observe o seguinte:

- Todos os objetos no bucket de diretório devem ser excluídos antes que o bucket em si possa ser excluído.
- Os multipart uploads em andamento no bucket de diretório devem ser anulados antes que o bucket em si possa ser excluído.

Note

O comando `s3 rm` por meio da AWS Command Line Interface (CLI), a operação `delete` por meio do Mountpoint e o botão de opção Esvaziar bucket por meio do AWS Management Console não conseguem excluir carregamentos fracionados em andamento em um bucket de diretório. Para excluir esses carregamentos fracionados em andamento, use a operação `ListMultipartUploads` para listá-los no bucket e a operação `AbortMultipartUpload` para anular todos eles.

Para excluir um bucket de diretório, consulte [Excluir um bucket de diretório](#). Para anular um upload multiparte em andamento, consulte [the section called “Abortar um multipart upload”](#).

Para esvaziar um bucket de uso geral, consulte [Esvaziar um bucket](#).

Uso do console do S3

Como esvaziar um bucket de diretório

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. Escolha a guia Buckets de diretórios.
4. Escolha o botão de opções ao lado do nome do bucket que você deseja esvaziar, depois selecione Esvaziar.
5. Na página Empty bucket (Esvaziar bucket), confirme se deseja esvaziar o bucket inserindo **permanently delete** no campo de texto e depois escolha Empty (Esvaziar).
6. Monitore o andamento do processo de esvaziamento do bucket na página Bucket vazio: status.

Excluir um bucket de diretório

Só é possível excluir buckets de diretório vazios do Amazon S3. Antes de excluir um bucket de diretório, é necessário excluir todos os objetos do bucket e anular todos os uploads multiparte em andamento.

Para esvaziar um bucket de diretório, consulte [Esvaziar um bucket de diretório](#). Para anular um upload multiparte em andamento, consulte [the section called “Abortar um multipart upload”](#).

Para excluir um bucket de uso geral, consulte [Excluir um bucket](#).

Uso do console do S3

Depois de esvaziar o bucket de diretório e anular todos os uploads multiparte em andamento, você poderá excluir 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. Escolha a guia Buckets de diretórios.
4. Na lista Buckets de diretório, escolha o botão de opções ao lado do bucket que você deseja excluir.
5. Escolha Excluir.
6. Na página Excluir bucket, insira o nome do bucket no campo de texto para confirmar a exclusão do bucket.

Important

Não é possível desfazer a exclusão de um bucket.

7. Para excluir um bucket de diretório, escolha Excluir bucket.

Uso dos AWS SDKs

Os exemplos a seguir excluem um bucket de diretório usando o AWS SDK for Java 2.x e o AWS SDK for Python (Boto3).

SDK for Java 2.x

Example

```
public static void deleteBucket(S3Client s3Client, String bucketName) {  
  
    try {  
        DeleteBucketRequest del = DeleteBucketRequest.builder()  
            .bucket(bucketName)  
            .build();  
        s3Client.deleteBucket(del);  
    }  
}
```

```
        System.out.println("Bucket " + bucketName + " has been deleted");
    }
    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

SDK for Python

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def delete_bucket(s3_client, bucket_name):
    """
    Delete a directory bucket in a specified Region

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket to delete; for example, 'doc-example-bucket--usw2-az1--x-s3'
    :return: True if bucket is deleted, else False
    """

    try:
        s3_client.delete_bucket(Bucket = bucket_name)
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'us-west-2'
    s3_client = boto3.client('s3', region_name = region)
```

Usando a AWS CLI

Este exemplo mostra como excluir um bucket de diretório usando a AWS CLI. Para usar o comando, substitua os *espaços reservados para entrada do usuário* por suas próprias informações.

```
aws s3api delete-bucket --bucket bucket-base-name--azid--x-s3 --region us-west-2
```

Para obter mais informações, consulte [delete-bucket](#) na AWS Command Line Interface.

Listar buckets de diretório

Os exemplos a seguir mostram como listar buckets de diretório usando SDKs da AWS e a AWS CLI.

Uso dos AWS SDKs

SDK for Java 2.x

Example

O exemplo a seguir lista os buckets de diretório usando o AWS SDK for Java 2.x.

```
public static void listBuckets(S3Client s3Client) {
    try {
        ListDirectoryBucketsRequest listDirectoryBucketsRequest =
ListDirectoryBucketsRequest.builder().build();
        ListDirectoryBucketsResponse response =
s3Client.listDirectoryBuckets(listDirectoryBucketsRequest);
        if (response.hasBuckets()) {
            for (Bucket bucket: response.buckets()) {
                System.out.println(bucket.name());
                System.out.println(bucket.creationDate());
            }
        }
    }
    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

SDK for Python

Example

O exemplo a seguir lista os buckets de diretório usando o AWS SDK for Python (Boto3).

```
import logging
import boto3
from botocore.exceptions import ClientError

def list_directory_buckets(s3_client):
    """
    Prints a list of all directory buckets in a Region

    :param s3_client: boto3 S3 client
    :return: True if there are buckets in the Region, else False
    """
    try:
        response = s3_client.list_directory_buckets()
        for bucket in response['Buckets']:
            print (bucket['Name'])
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    region = 'us-east-1'
    s3_client = boto3.client('s3', region_name = region)
    list_directory_buckets(s3_client)
```

AWS SDK for .NET

Example

O exemplo a seguir lista os buckets de diretório usando o AWS SDK for .NET.

```
var listDirectoryBuckets = await amazonS3Client.ListDirectoryBucketsAsync(new
    ListDirectoryBucketsRequest
{
    MaxDirectoryBuckets = 10
}).ConfigureAwait(false);
```


SDK for PHP

Example

O exemplo a seguir lista os buckets de diretório usando o AWS SDK for PHP.

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

$s3Client = new S3Client([
    'region' => 'us-east-1',
]);
$result = $s3Client->listDirectoryBuckets();
```

SDK for Ruby

Example

O exemplo a seguir lista os buckets de diretório usando o AWS SDK for Ruby.

```
s3 = Aws::S3::Client.new(region:'us-west-1')
s3.list_directory_buckets
```

Usando a AWS CLI

O exemplo de comando `list-directory-buckets` a seguir mostra como você pode usar a AWS CLI para listar seus buckets de diretório na região `us-east-1`. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3api list-directory-buckets --region us-east-1
```

Para obter mais informações, consulte [list-directory-buckets](#) na Referência de comandos da AWS CLI.

Usar **HeadBucket** com buckets de diretório

Os exemplos de AWS SDK a seguir mostram como usar a operação de API `HeadBucket` para determinar se existe um bucket de diretório do Amazon S3 e se você tem permissão para acessá-lo.

Uso dos AWS SDKs

O exemplo a seguir para AWS SDK for Java 2.x mostra como determinar se um bucket existe e se você tem permissão para acessá-lo.

SDK for Java 2.x

Example

AWS SDK for Java 2.x

```
public static void headBucket(S3Client s3Client, String bucketName) {
    try {
        HeadBucketRequest headBucketRequest = HeadBucketRequest
            .builder()
            .bucket(bucketName)
            .build();
        s3Client.headBucket(headBucketRequest);
        System.out.format("Amazon S3 bucket: \"%s\" found.", bucketName);
    }

    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

Usando a AWS CLI

O exemplo de comando `head-bucket` a seguir mostra como é possível usar a AWS CLI para determinar se um bucket de diretório existe e se você tem permissão para acessá-lo. Para executar esse comando, substitua os espaços reservados de entrada por suas próprias informações.

```
aws s3api head-bucket --bucket bucket-base-name--azid--x-s3
```

Para obter mais informações, consulte [head-bucket](#) na Referência de comandos da AWS CLI.

Trabalhar com objetos em um bucket de diretório

Depois de criar um bucket de diretório do Amazon S3, você pode trabalhar com objetos usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI) e os AWS SDKs.

Para obter mais informações sobre as operações em massa com objetos armazenados na classe S3 Express One Zone, consulte [Gerenciamento de objetos](#). Para obter mais informações sobre como importar, fazer upload, copiar, excluir e baixar objetos e ler metadados dos objetos em buckets de diretório, consulte os tópicos a seguir.

Tópicos

- [Importar objetos em um bucket de diretório](#)
- [Usar a funcionalidade Operações em Lote com a classe S3 Express One Zone](#)
- [Fazer upload de um objeto em um bucket de diretório](#)
- [Usar uploads multiparte com buckets de diretório](#)
- [Copiar um objeto para um bucket de diretório](#)
- [Excluir um objeto de um bucket de diretório](#)
- [Baixar um objeto de um bucket de diretório](#)
- [Usar HeadObject com buckets de diretório](#)

Importar objetos em um bucket de diretório

Depois de criar um bucket de diretório no Amazon S3, você pode preencher o novo bucket com dados usando a ação de importação. A importação é um método simplificado para criar trabalhos da funcionalidade Operações em Lote do S3 a fim de copiar objetos de buckets de uso geral para buckets de diretório.

Note

As seguintes limitações se aplicam aos trabalhos de importação:

- O bucket de origem e o bucket de destino devem estar nas mesmas conta e Região da AWS.
- O bucket de origem não pode ser um bucket de diretório.
- Objetos maiores que 5 GB não são compatíveis e serão omitidos da operação de cópia.

- Os objetos nas classes de armazenamento Glacier Flexible Retrieval, Glacier Deep Archive, Intelligent-Tiering Archive Access e Intelligent-Tiering Deep Archive devem ser restaurados antes que possam ser importados.
- Os objetos importados com algoritmos de soma de verificação MD5 são convertidos para usar somas de verificação CRC32.
- Os objetos importados usam criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3).
- Os objetos importados usam a classe de armazenamento Express One Zone, que tem uma estrutura de preços diferente das classes de armazenamento usadas por buckets de uso geral. Considere essa diferença de custo ao importar um grande número de objetos.

Ao configurar um trabalho de importação, especifique o bucket ou prefixo de origem de onde os objetos existentes serão copiados. Forneça também um perfil do AWS Identity and Access Management (IAM) com permissões para acessar os objetos de origem. Depois, o Amazon S3 inicia um trabalho da funcionalidade Operações em Lote que copia os objetos e aplica automaticamente as configurações apropriadas de classe de armazenamento e soma de verificação.


Para configurar trabalhos de importação, use o console do Amazon S3.

Usar o console do Amazon S3

Como importar objetos em um bucket de diretório

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 Buckets e selecione a guia Buckets de diretórios. Selecione o botão de opções ao lado do bucket de diretório para o qual você deseja importar objetos.
3. Escolha Importar.
4. Em Origem, insira o bucket de uso geral (ou o caminho do bucket com o prefixo) que contém os objetos que você deseja importar. Para escolher um bucket de uso geral existente em uma lista, selecione Procurar no S3.
5. Em Permissão para acessar e copiar objetos de origem, execute uma das seguintes ações para especificar um perfil do IAM com as permissões necessárias para importar os objetos da origem:

- Para permitir que o Amazon S3 crie um perfil do IAM em seu nome, escolha Criar perfil do IAM.

 Note

Se os objetos da origem estiverem criptografados com criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS), não escolha a opção Criar perfil do IAM. Em vez disso, especifique um perfil do IAM existente que tenha a permissão `kms:Decrypt`.

O Amazon S3 usará essa permissão para descriptografar os objetos. Durante o processo de importação, o Amazon S3 recriptografará esses objetos usando criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3).

- Para escolher um perfil do IAM existente em uma lista, escolha Escolher entre perfis do IAM existentes.
 - Para especificar um perfil do IAM existente inserindo o nome do recurso da Amazon (ARN), escolha Inserir ARN do perfil do IAM e insira o ARN no campo correspondente.
6. Revise as informações exibidas nas seções Destino e Configurações do objeto copiado. Se as informações na seção Destino estiverem corretas, escolha Importar para iniciar o trabalho de cópia.

O console do Amazon S3 exibirá o status do novo trabalho na página Operações em lotes. Para obter mais informações sobre o trabalho, escolha o botão de opções ao lado do nome do trabalho e selecione Visualizar detalhes no menu Ações. Para abrir o bucket de diretório para o qual os objetos serão importados, escolha Exibir destino de importação.

Usar a funcionalidade Operações em Lote com a classe S3 Express One Zone

Você pode usar a funcionalidade Operações em Lote do Amazon S3 para executar operações em lote de grande escala em objetos armazenados em buckets do S3. Para saber mais sobre a funcionalidade Operações em Lote do S3, consulte [Executar operações em lote de grande escala em objetos do Amazon S3](#).

Os tópicos a seguir discutem a execução de operações em lote em objetos armazenados em buckets de diretório na classe de armazenamento S3 Express One Zone.

Tópicos

- [Usar a funcionalidade Operações em Lote com buckets de diretório](#)
- [Principais diferenças](#)

Usar a funcionalidade Operações em Lote com buckets de diretório

Você pode realizar a operação Copiar e as operações Invocar função do AWS Lambda em objetos armazenados em buckets de diretório. Com Copiar, é possível copiar objetos entre buckets do mesmo tipo (por exemplo, de um bucket de diretório para outro bucket de diretório). Você também pode copiar entre buckets de uso geral e buckets de diretório. Com Invocar função do AWS Lambda, você pode usar uma função do Lambda para realizar ações em objetos no seu bucket de diretório com código definido por você.

Cópia de objetos

Você pode copiar entre o mesmo tipo de bucket ou entre buckets de diretório e buckets de uso geral. Ao copiar para um bucket de diretório, você deve usar o formato de nome do recurso da Amazon (ARN) correto para esse tipo de bucket. O formato de ARN para um bucket de diretório é `arn:aws:s3express:region:account-id:bucket/bucket-base-name--x-s3`.

Você também pode preencher o bucket de diretório com dados usando a ação Importar no console do S3. Importar é um método simplificado para criar trabalhos da funcionalidade Operações em Lote a fim de copiar objetos de buckets de uso geral para buckets de diretório. Para trabalhos de cópia com Importar de buckets de uso geral para buckets de diretório, o S3 gera um manifesto automaticamente. Para obter mais informações, consulte [Importar objetos em um bucket de diretório](#) e [Specifying a manifest](#).

Invocar funções do Lambda (**LambdaInvoke**)

Há requisitos especiais para usar a funcionalidade Operações em Lote para invocar funções do Lambda que atuam em buckets de diretório. Por exemplo, você deve estruturar a solicitação do Lambda usando um esquema de invocação JSON v2 e especificar `InvocationSchemaVersion 2.0` ao criar o trabalho. Para obter mais informações, consulte [Invocar função do AWS Lambda](#).

Principais diferenças

Veja a seguir uma lista das principais diferenças ao usar a funcionalidade Operações em Lote para realizar operações em massa para objetos armazenados em buckets de diretório com a classe de armazenamento S3 Express One Zone:

- O Amazon S3 criptografa automaticamente todos os novos objetos que são carregados em um bucket do S3. A configuração de criptografia padrão de um bucket do S3 está sempre habilitada e, no mínimo, definida como criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3). Buckets de diretório só são compatíveis com SSE-S3. Se você fizer uma solicitação CopyObject que defina a criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C) ou a criptografia do lado do servidor com chaves do AWS KMS (AWS Key Management Service) (SSE-KMS) em um bucket de diretório (origem ou destino), a resposta retornará um erro HTTP 400 (Bad Request).
- Não é possível marcar objetos em buckets de diretório. Você só pode especificar um conjunto vazio de etiquetas. Por padrão, a funcionalidade Operações em Lote copia as etiquetas. Se você copiar um objeto que tenha etiquetas entre buckets de uso geral e buckets de diretório, receberá uma resposta 501 (Not Implemented).
- A classe S3 Express One Zone oferece a opção de escolher o algoritmo de soma de verificação usado para validar os dados durante uploads ou downloads. Você pode selecionar um dos seguintes algoritmos de verificação de integridade de dados de hash seguro (SHA) ou de verificação de redundância cíclica (CRC): CRC32, CRC32, SHA-1 e SHA-256. As somas de verificação baseadas em MD5 não são compatíveis com a classe de armazenamento S3 Express One Zone.
- Por padrão, todos os buckets do Amazon S3 definem a configuração Propriedade do objeto do S3 como Imposto pelo proprietário do bucket e as listas de controle de acesso (ACLs) estão desabilitadas. Para buckets de diretório, essa configuração não pode ser modificada. Você pode copiar um objeto de buckets de uso geral para buckets de diretório. No entanto, não é possível sobrescrever a ACL padrão ao copiar de ou para um bucket de diretório.
- Independentemente de como você especifica o manifesto, a lista em si deve ser armazenada em um bucket de uso geral. A funcionalidade Operações em Lote não pode importar manifestos existentes de buckets de diretório (nem salvar manifestos gerados neles). No entanto, os objetos descritos no manifesto podem ser armazenados em buckets de diretório.
- A funcionalidade Operações em Lote não consegue especificar um bucket de diretório como um local em um relatório da funcionalidade Inventário S3. Os relatórios da funcionalidade Inventário não são compatíveis com buckets de diretório. Você pode criar um arquivo de manifesto para objetos em um bucket de diretório usando a operação de API ListObjectsV2 para listar os objetos. Depois, você pode inserir a lista em um arquivo CSV.

Como conceder acesso ao

Para realizar trabalhos de cópia, você deve ter as seguintes permissões:

- Para copiar objetos de um bucket de diretório para outro bucket de diretório, você precisa ter a permissão `s3express:CreateSession`.
- Para copiar objetos de buckets de diretório para buckets de uso geral, você precisa ter a permissão `s3express:CreateSession` e a permissão `s3:PutObject` para gravar a cópia do objeto no bucket de destino.
- Para copiar objetos de buckets de uso geral para buckets de diretório, você precisa ter a permissão `s3express:CreateSession` e a permissão `s3:GetObject` para ler o objeto de origem que está sendo copiado.

Para obter mais informações, consulte [CopyObject](#) na Referência da API do Amazon Simple Storage Service.

- Para invocar uma função do Lambda, você deve conceder permissões ao recurso com base na função do Lambda. Para determinar quais permissões são necessárias, verifique as operações de API correspondentes.

Fazer upload de um objeto em um bucket de diretório

Depois de criar um bucket de diretório do Amazon S3, você pode fazer upload de objetos nele. Os exemplos a seguir mostram como fazer upload de um objeto em um bucket de diretório usando o console do S3 e os AWS SDKs. Para obter informações sobre as operações de upload de objetos em massa com a classe S3 Express One Zone, consulte [Gerenciamento 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. No painel de navegação à esquerda, escolha Buckets.
3. Escolha a guia Buckets de diretórios.
4. Escolha o nome do bucket no qual você deseja fazer upload de pastas ou arquivos.
5. Na guia Objetos, escolha Fazer upload.
6. Na página Fazer upload, execute uma das seguintes ações:
 - Arraste e solte arquivos e pastas para a área de upload tracejada.
 - Escolha Adicionar arquivos ou Adicionar pasta, selecione os arquivos ou as pastas que deseja carregar e escolha Abrir ou Fazer upload.

7. Em Checksums, escolha a Função de soma de verificação que deseja usar.

(Opcional) Se você estiver fazendo upload de um único objeto com menos de 16 MB, também poderá especificar um valor de soma de verificação pré-calculado. Quando você fornece um valor pré-calculado, o Amazon S3 compara esse valor com o valor calculado usando a função de soma de verificação selecionada. Se os valores não corresponderem, o upload não será iniciado.

8. As opções nas seções Permissões e Propriedades são definidas automaticamente com as configurações padrão e não podem ser modificadas. A funcionalidade Bloqueio de Acesso Público é habilitada automaticamente, e as funcionalidades Versionamento do S3 e Bloqueio de Objetos do S3 não podem ser habilitadas para buckets de diretório.

(Opcional) Se você quiser adicionar metadados em pares de chave e valor aos objetos, expanda a seção Propriedades e escolha Adicionar metadados na seção Metadados.

9. Para fazer upload das pastas e dos arquivos listados, escolha Fazer upload.

O Amazon S3 faz o upload de seus objetos e pastas. Quando o upload for concluído, você verá uma mensagem de sucesso na página Upload: status.

Uso dos AWS SDKs

SDK for Java 2.x

Example

```
public static void putObject(S3Client s3Client, String bucketName, String objectKey,
    Path filePath) {
    //Using File Path to avoid loading the whole file into memory
    try {
        PutObjectRequest putObj = PutObjectRequest.builder()
            .bucket(bucketName)
            .key(objectKey)
            //.metadata(metadata)
            .build();
        s3Client.putObject(putObj, filePath);
        System.out.println("Successfully placed " + objectKey + " into bucket
            "+bucketName);
    }
}
```

```
        catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

SDK for Python

Example

```
import boto3
import botocore
from botocore.exceptions import ClientError

def put_object(s3_client, bucket_name, key_name, object_bytes):
    """
    Upload data to a directory bucket.
    :param s3_client: The boto3 S3 client
    :param bucket_name: The bucket that will contain the object
    :param key_name: The key of the object to be uploaded
    :param object_bytes: The data to upload
    """
    try:
        response = s3_client.put_object(Bucket=bucket_name, Key=key_name,
                                         Body=object_bytes)
        print(f"Upload object '{key_name}' to bucket '{bucket_name}'.")
        return response
    except ClientError:
        print(f"Couldn't upload object '{key_name}' to bucket '{bucket_name}'.")
        raise

def main():
    # Share the client session with functions and objects to benefit from S3 Express
    # One Zone auth key
    s3_client = boto3.client('s3')
    # Directory bucket name must end with --azid--x-s3
    resp = put_object(s3_client, 'doc-bucket-example--use1-az5--x-s3', 'sample.txt',
                      b'Hello, World!')
    print(resp)

if __name__ == "__main__":
    main()
```

Usando a AWS CLI

O exemplo de comando `put-object` a seguir mostra como você pode usar a AWS CLI para carregar um objeto do Amazon S3. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3api put-object --bucket bucket-base-name--azid--x-s3 --key sampleinput/file001.bin
--body bucket-seed/file001.bin
```

Para obter mais informações, consulte [put-object](#) na Referência de comandos da AWS CLI.

Usar uploads multipart com buckets de diretório

Você pode usar o processo de upload multipart para fazer 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.
- Rápida recuperação de qualquer problema de rede: o tamanho menor das partes minimiza o impacto de reiniciar um upload 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. Um upload multipart iniciado não tem data de expiração. Você deve concluir ou anular explicitamente o upload multipart.
- 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 uploads multipart das seguintes maneiras:

- Se você estiver fazendo upload de objetos grandes em uma rede de banda larga estável, use os uploads multipart para maximizar o uso da banda larga disponível, fazendo upload das partes do objeto em paralelo para performance com vários threads.

- Se você estiver fazendo upload em uma rede lenta, use os uploads multiparte para aumentar a resiliência dos erros de rede, evitando reinícios de upload. Ao usar os uploads multiparte, tente fazer upload novamente apenas das partes que foram interrompidas durante o upload. Você não precisa reiniciar o upload do seu objeto do começo.

Quando você usa uploads multiparte para fazer upload de objetos para a classe de armazenamento Amazon S3 Express One Zone em buckets de diretório, o processo de upload multiparte é semelhante ao processo de uso de um upload multiparte para fazer upload de objetos em buckets de uso geral. No entanto, há algumas diferenças importantes.

Para obter mais informações sobre como usar uploads multiparte para fazer upload de objetos na classe S3 Express One Zone, consulte os tópicos a seguir.

Tópicos

- [O processo de upload multiparte](#)
- [Somadas de verificação com operações de carregamento fracionado](#)
- [Operações simultâneas de carregamento fracionado](#)
- [Uploads multiparte e preços](#)
- [Operações de API de upload multiparte e permissões](#)
- [Exemplos](#)

O processo de upload multiparte

Um upload multiparte é um processo de três etapas:

- Você inicia o upload.
- Você faz upload das partes do objeto.
- Depois de fazer upload de todas as partes, você conclui o upload multiparte.

Ao receber a solicitação de conclusão do upload multiparte, o Amazon S3 cria o objeto com base nas partes carregadas, e você poderá acessar o objeto como qualquer outro objeto no bucket.

Iniciação do multipart upload

Quando você envia uma solicitação para iniciar um multipart upload, o Amazon S3 retorna uma resposta com um ID de upload, que é um identificador exclusivo do seu multipart upload. Você deve

incluir esse ID de upload sempre que fizer upload de partes, listar as partes, concluir um upload ou anular um upload.

Carregamento de partes

Ao fazer upload de uma parte, além do ID de upload, você deve especificar um número de parte. Ao usar um upload multiparte com a classe S3 Express One Zone, os números do upload multiparte devem ser consecutivos. Se você tentar concluir uma solicitação de upload multiparte com números de parte não consecutivos, será gerado um erro HTTP 400 Bad Request (ordem das partes inválida).

Um número de parte identifica de maneira única uma parte e sua posição no objeto que está sendo carregado. 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. Os valores de ETag para todos os uploads de partes do objeto permanecerá o mesmo, mas cada parte receberá um número de parte diferente. Você tem que incluir esses valores na solicitação subsequente para concluir o multipart upload.

O Amazon S3 criptografa automaticamente todos os novos objetos que são carregados em um bucket do S3. Ao realizar um carregamento multiparte, se você não especificar informações de criptografia na solicitação, a configuração de criptografia das partes carregadas será definida como a configuração de criptografia padrão do bucket de destino. A configuração de criptografia padrão de um bucket do Amazon S3 está sempre habilitada e, no mínimo, definida como criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3). Buckets de diretório só são compatíveis com SSE-S3. Para ter mais informações, consulte [Criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#).

Conclusão de carregamento fracionado

Quando você concluir um upload multiparte, o Amazon S3 criará o objeto concatenando as partes em ordem crescente com base no número da parte. Depois de uma solicitação de conclusão bem-sucedida, as partes não existem mais.

Sua solicitação concluir upload multiparte deve incluir o ID de upload e uma lista dos números das partes 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 um hash MD5 dos dados do objeto.

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 carregamento multiparte, você deverá usar a paginação para recuperar todas as partes.

A lista de partes retornada não inclui aquelas cujo upload 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 ou anulou. 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.

Para obter mais informações sobre listagens de upload multiparte, consulte [ListParts](#) na Referência da API do Amazon Simple Storage Service.

Somas de verificação com operações de carregamento fracionado

Ao fazer upload de um objeto, você pode especificar um algoritmo de soma de verificação para verificar a integridade dele. O MD5 não é compatível com buckets de diretório. Você pode especificar um dos seguintes algoritmos de verificação de integridade de dados de hash seguro (SHA) ou de verificação de redundância cíclica (CRC):

- CRC32
- CRC32C
- SHA-1
- SHA-256

Você pode usar a API REST do Amazon S3 ou os AWS SDKs para recuperar o valor da soma de verificação para partes específicas usando `GetObject` ou `HeadObject`. Se você quiser recuperar

os valores da soma de verificação para partes individuais de carregamentos fracionados ainda em andamento, você pode usar `ListParts`.

Important

Ao usar os algoritmos de soma de verificação anteriores, os números de parte do upload multiparte devem ser consecutivos. Se você tentar concluir uma solicitação de upload multiparte com números de parte não consecutivos, o Amazon S3 gerará um erro HTTP 400 Bad Request (ordem das partes inválida).

Para obter mais informações sobre como as somas de verificação funcionam com objetos fracionados, consulte [Verificar a integridade do objeto](#).

Operações simultâneas de carregamento fracionado

Em um ambiente de desenvolvimento distribuído, a aplicação pode iniciar várias atualizações no mesmo objeto ao mesmo tempo. Por exemplo, a aplicação pode iniciar vários uploads multiparte 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. Para a classe S3 Express One Zone, a hora de criação do objeto é a data de conclusão do carregamento multiparte.

Important

O versionamento não é compatível com objetos armazenados em buckets de diretório.

Uploads multiparte e preços

Depois que você iniciar um carregamento multiparte, o Amazon S3 reterá todas as partes até você concluir ou anular o upload. Durante todo o ciclo de vida, você será cobrado por armazenamento, largura de banda e solicitações desse multipart upload e das partes associadas. Se você anular o upload multiparte, o Amazon S3 excluirá os artefatos de upload e todas as partes carregadas, o que significa que não serão mais cobrados de você. Não há cobrança de exclusão antecipada pela exclusão de uploads multiparte incompletos, independentemente da classe de armazenamento especificada. Para obter mais informações sobre a definição de preços, consulte [Definição de preços do Amazon S3](#).

⚠ Important

Se a solicitação de conclusão do upload multipart não for enviada com sucesso, as partes do objeto não serão montadas e o objeto não será criado. Você é cobrado por todo o armazenamento associado a partes enviadas por upload. É importante que você conclua o upload multipart para que o objeto seja criado ou anule o upload multipart para remover todas as partes carregadas.

Antes de excluir um bucket de diretório, é necessário concluir ou anular todos os uploads multipart em andamento. Os buckets de diretório não oferecem suporte às configurações da funcionalidade Ciclo de Vida do S3. Se necessário, você pode listar os uploads multipart ativos, depois anulá-los e excluir o bucket.

Operações de API de upload multipart e permissões

Para permitir o acesso às operações de API de gerenciamento de objetos em um bucket de diretório, conceda a permissão `s3express:CreateSession` em uma política de bucket ou em uma política baseada em identidade do AWS Identity and Access Management (IAM).

Você deve ter as permissões necessárias para usar as operações do multipart upload. Você pode usar políticas de bucket ou políticas baseadas em identidade do IAM para conceder permissões a entidades principais do IAM para realizar essas operações. A tabela a seguir lista as permissões necessárias para diversas operações de carregamento multipart.

Você pode identificar o iniciador de um upload multipart por meio do elemento `Initiator`. Se o iniciador for uma Conta da AWS, esse elemento fornecerá as mesmas informações que o elemento `Owner`. Se o iniciador for um usuário do IAM, esse elemento fornecerá o ARN e o nome da exibição do usuário.

Ação	Permissões obrigatórias
Criar um carregamento fracionado	Para criar um upload multipart, você deve ter permissão para realizar a ação <code>s3express:CreateSession</code> em um bucket de diretório.
Iniciar um upload multipart	Para iniciar o upload multipart, você deve ter permissão para realizar a ação <code>s3express:CreateSession</code> no bucket de diretório.

Ação	Permissões obrigatórias
Fazer upload de uma parte	<p>Para fazer upload de uma parte, você deve ter permissão para realizar a ação <code>s3express:CreateSession</code> em um bucket de diretório.</p> <p>Para que o iniciador possa fazer upload de uma parte, o proprietário do bucket deve permitir que o iniciador realize a ação <code>s3express:CreateSession</code> em um bucket de diretório.</p>
Fazer upload de uma parte (cópia)	<p>Para fazer upload de uma parte, você deve ter permissão para realizar a ação <code>s3express:CreateSession</code> em um bucket de diretório.</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>s3express:CreateSession</code> no objeto.</p>
Concluir um carregamento fracionado	<p>Para concluir um upload multiparte, você deve ter permissão para realizar a ação <code>s3express:CreateSession</code> no bucket de diretório.</p> <p>Para que o iniciador possa concluir um upload multiparte, o proprietário do bucket deve permitir que o iniciador realize a ação <code>s3express:CreateSession</code> no objeto.</p>
Anular um multipart upload	<p>Para anular um upload multiparte, você deve ter permissão para realizar a ação <code>s3express:CreateSession</code> .</p> <p>Para que o iniciador anule um upload multiparte, ele deve receber permissão explícita para realizar a ação <code>s3express:CreateSession</code> .</p>
Listar partes	<p>Para listar as partes de um upload multiparte, você deve ter permissão para realizar a ação <code>s3express:CreateSession</code> no bucket de diretório.</p>
Listar carregamentos fracionados em andamento	<p>Para listar os uploads multiparte em andamento em um bucket, você deve ter permissão para realizar a ação <code>s3:ListBucketMultipartUploads</code> nesse bucket.</p>

Suporte às operações de API para uploads multiparte

As seções a seguir na Referência da API do Amazon Simple Storage Service descrevem a API REST do Amazon S3 para uploads multiparte.

- [CreateMultipartUpload](#)
- [UploadPart](#)
- [UploadPartCopy](#)
- [CompleteMultipartUpload](#)
- [AbortMultipartUpload](#)
- [ListParts](#)
- [ListMultipartUploads](#)

Exemplos

Para usar um upload multiparte a fim de fazer upload de um objeto na classe S3 Express One Zone em um bucket de diretório, consulte os exemplos a seguir.

Tópicos

- [Criar um carregamento multiparte](#)
- [Fazer upload das partes de um upload multiparte](#)
- [Concluir um carregamento multiparte](#)
- [Abortar um multipart upload](#)
- [Criar uma operação de cópia de carregamento multiparte](#)
- [Listar os uploads multiparte em andamento](#)
- [Listar as partes de um upload multiparte](#)

Criar um carregamento multiparte

Os exemplos a seguir mostram como criar um carregamento fracionado.

Uso dos AWS SDKs

SDK for Java 2.x

Example

```
/**
 * This method creates a multipart upload request that generates a unique upload ID
 * that is used to track
 * all the upload parts
 *
 * @param s3
 * @param bucketName - for example, 'doc-example-bucket--use1-az4--x-s3'
 * @param key
 * @return
 */
private static String createMultipartUpload(S3Client s3, String bucketName, String
key) {

    CreateMultipartUploadRequest createMultipartUploadRequest =
CreateMultipartUploadRequest.builder()
        .bucket(bucketName)
        .key(key)
        .build();

    String uploadId = null;

    try {
        CreateMultipartUploadResponse response =
s3.createMultipartUpload(createMultipartUploadRequest);
        uploadId = response.uploadId();
    }
    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return uploadId;
}
```

SDK for Python

Example

```
def create_multipart_upload(s3_client, bucket_name, key_name):
```

```

'''
Create a multipart upload to a directory bucket

:param s3_client: boto3 S3 client
:param bucket_name: The destination bucket for the multipart upload
:param key_name: The key name for the object to be uploaded
:return: The UploadId for the multipart upload if created successfully, else None
'''

try:
    mpu = s3_client.create_multipart_upload(Bucket = bucket_name, Key =
key_name)
    return mpu['UploadId']
except ClientError as e:
    logging.error(e)
    return None

```

Usando a AWS CLI

Este exemplo mostra como criar um carregamento fracionado em um bucket de diretório usando a AWS CLI. Esse comando inicia um carregamento fracionado no bucket de diretório *bucket-base-name--azid--x-s3* para o objeto *KEY_NAME*. Para usar o comando, substitua os *espaços reservados para entrada do usuário* por suas próprias informações.

```
aws s3api create-multipart-upload --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
```

Para obter mais informações, consulte [create-multipart-upload](#) na AWS Command Line Interface.

Fazer upload das partes de um upload multiparte

Os exemplos a seguir mostram como carregar partes de um carregamento fracionado.

Uso dos AWS SDKs

SDK for Java 2.x

O exemplo a seguir mostra como dividir um único objeto em partes e carregá-las em um bucket de diretório usando o SDK para Java 2.x.

Example

```
/**
```

```
* This method creates part requests and uploads individual parts to S3 and then
returns all the completed parts
*
* @param s3
* @param bucketName
* @param key
* @param uploadId
* @throws IOException
*/
private static List<CompletedPart> multipartUpload(S3Client s3, String bucketName,
String key, String uploadId, String filePath) throws IOException {

    int partNumber = 1;
    List<CompletedPart> completedParts = new ArrayList<>();
    ByteBuffer bb = ByteBuffer.allocate(1024 * 1024 * 5); // 5 MB byte buffer

    // read the local file, breakdown into chunks and process
    try (RandomAccessFile file = new RandomAccessFile(filePath, "r")) {
        long fileSize = file.length();
        int position = 0;
        while (position < fileSize) {
            file.seek(position);
            int read = file.getChannel().read(bb);

            bb.flip(); // Swap position and limit before reading from the
buffer.

            UploadPartRequest uploadPartRequest = UploadPartRequest.builder()
                .bucket(bucketName)
                .key(key)
                .uploadId(uploadId)
                .partNumber(partNumber)
                .build();

            UploadPartResponse partResponse = s3.uploadPart(
                uploadPartRequest,
                RequestBody.fromByteBuffer(bb));

            CompletedPart part = CompletedPart.builder()
                .partNumber(partNumber)
                .eTag(partResponse.eTag())
                .build();
            completedParts.add(part);

            bb.clear();
```

```
        position += read;
        partNumber++;
    }
}

catch (IOException e) {
    throw e;
}
return completedParts;
}
```

SDK for Python

O exemplo a seguir mostra como dividir um único objeto em partes e carregá-las em um bucket de diretório usando o SDK para Python.

Example

```
def multipart_upload(s3_client, bucket_name, key_name, mpu_id, part_size):
    """
    Break up a file into multiple parts and upload those parts to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Destination bucket for the multipart upload
    :param key_name: Key name for object to be uploaded and for the local file
    that's being uploaded
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_size: The size parts that the object will be broken into, in bytes.
        Minimum 5 MiB, Maximum 5 GiB. There is no minimum size for the
    last part of your multipart upload.
    :return: part_list for the multipart upload if all parts are uploaded
    successfully, else None
    """

    part_list = []
    try:
        with open(key_name, 'rb') as file:
            part_counter = 1
            while True:
                file_part = file.read(part_size)
                if not len(file_part):
                    break
                upload_part = s3_client.upload_part(
                    Bucket = bucket_name,
```

```

        Key = key_name,
        UploadId = mpu_id,
        Body = file_part,
        PartNumber = part_counter
    )
    part_list.append({'PartNumber': part_counter, 'ETag':
upload_part['ETag']})
    part_counter += 1
except ClientError as e:
    logging.error(e)
    return None
return part_list

```

Usando a AWS CLI

O exemplo a seguir mostra como dividir um único objeto em partes e carregá-las em um bucket de diretório usando a AWS CLI. Para usar o comando, substitua os *espaços reservados para entrada do usuário* por suas próprias informações.

```

aws s3api upload-part --bucket bucket-base-name--azid--x-s3 --
key KEY_NAME --part-number 1 --body LOCAL_FILE_NAME --upload-id
"AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAAAA0AAAAAAAAAAAH2AfYAA"

```

Para obter mais informações, consulte [upload-part](#) na AWS Command Line Interface.

Concluir um carregamento multiparte

Os exemplos a seguir mostram como concluir um carregamento fracionado.

Uso dos AWS SDKs

SDK for Java 2.x

Os exemplos a seguir mostram como concluir um carregamento fracionado usando o SDK para Java 2.x.

Example

```

/**
 * This method completes the multipart upload request by collating all the upload
 parts
 * @param s3

```

```
* @param bucketName - for example, 'doc-example-bucket--usw2-az1--x-s3'
* @param key
* @param uploadId
* @param uploadParts
*/
private static void completeMultipartUpload(S3Client s3, String bucketName, String
key, String uploadId, List<CompletedPart> uploadParts) {
    CompletedMultipartUpload completedMultipartUpload =
CompletedMultipartUpload.builder()
        .parts(uploadParts)
        .build();

    CompleteMultipartUploadRequest completeMultipartUploadRequest =
        CompleteMultipartUploadRequest.builder()
            .bucket(bucketName)
            .key(key)
            .uploadId(uploadId)
            .multipartUpload(completedMultipartUpload)
            .build();

    s3.completeMultipartUpload(completeMultipartUploadRequest);
}

public static void multipartUploadTest(S3Client s3, String bucketName, String
key, String localFilePath) {
    System.out.println("Starting multipart upload for: " + key);
    try {
        String uploadId = createMultipartUpload(s3, bucketName, key);
        System.out.println(uploadId);
        List<CompletedPart> parts = multipartUpload(s3, bucketName, key, uploadId,
localFilePath);
        completeMultipartUpload(s3, bucketName, key, uploadId, parts);
        System.out.println("Multipart upload completed for: " + key);
    }

    catch (Exception e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
}
```


SDK for Python

Os exemplos a seguir mostram como concluir um carregamento fracionado usando o SDK para Python.

Example

```
def complete_multipart_upload(s3_client, bucket_name, key_name, mpu_id, part_list):
    """
    Completes a multipart upload to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: The destination bucket for the multipart upload
    :param key_name: The key name for the object to be uploaded
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_list: The list of uploaded part numbers with their associated ETags
    :return: True if the multipart upload was completed successfully, else False
    """

    try:
        s3_client.complete_multipart_upload(
            Bucket = bucket_name,
            Key = key_name,
            UploadId = mpu_id,
            MultipartUpload = {
                'Parts': part_list
            }
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    MB = 1024 ** 2
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    key_name = 'OBJECT_NAME'
    part_size = 10 * MB
    s3_client = boto3.client('s3', region_name = region)
    mpu_id = create_multipart_upload(s3_client, bucket_name, key_name)
    if mpu_id is not None:
        part_list = multipart_upload(s3_client, bucket_name, key_name, mpu_id,
            part_size)
```

```

    if part_list is not None:
        if complete_multipart_upload(s3_client, bucket_name, key_name, mpu_id,
part_list):
            print (f'{key_name} successfully uploaded through a multipart upload
to {bucket_name}')
        else:
            print (f'Could not upload {key_name} through a multipart upload to
{bucket_name}')

```

Usando a AWS CLI

Este exemplo mostra como concluir um carregamento fracionado em um bucket de diretório usando a AWS CLI. Para usar o comando, substitua os *espaços reservados para entrada do usuário* por suas próprias informações.

```

aws s3api complete-multipart-upload --bucket bucket-base-name--azid--x-s3 --
key KEY_NAME --upload-id
"AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAA0AAAAAAAAAAH2AfYAA
--multipart-upload file://parts.json

```

Este exemplo usa uma estrutura JSON que descreve quais partes do carregamento fracionado devem ser reagrupadas no arquivo completo. Neste exemplo, o prefixo `file://` é usado para carregar a estrutura JSON de um arquivo na pasta local chamada `parts`.

parts.json:

```

parts.json
{
  "Parts": [
    {
      "ETag": "6b78c4a64dd641a58dac8d9258b88147",
      "PartNumber": 1
    }
  ]
}

```

Para obter mais informações, consulte [complete-multipart-upload](#) na AWS Command Line Interface.

Abortar um multipart upload

Os exemplos a seguir mostram como interromper um carregamento fracionado.

Uso dos AWS SDKs

SDK for Java 2.x

O exemplo a seguir mostra como interromper um carregamento fracionado usando o SDK para Java 2.x.

Example

```
public static void abortMultiPartUploads( S3Client s3, String bucketName ) {

    try {
        ListMultipartUploadsRequest listMultipartUploadsRequest =
ListMultipartUploadsRequest.builder()
        .bucket(bucketName)
        .build();

        ListMultipartUploadsResponse response =
s3.listMultipartUploads(listMultipartUploadsRequest);
        ListMultipartUpload uploads = response.uploads();

        AbortMultipartUploadRequest abortMultipartUploadRequest;
        for (MultipartUpload upload: uploads) {
            abortMultipartUploadRequest = AbortMultipartUploadRequest.builder()
                .bucket(bucketName)
                .key(upload.key())
                .uploadId(upload.uploadId())
                .build();

            s3.abortMultipartUpload(abortMultipartUploadRequest);
        }
    }

    catch (S3Exception e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
}
```

SDK for Python

O exemplo a seguir mostra como interromper um carregamento fracionado usando o SDK para Python.

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def abort_multipart_upload(s3_client, bucket_name, key_name, upload_id):
    """
    Aborts a partial multipart upload in a directory bucket.

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket where the multipart upload was initiated - for
    example, 'doc-example-bucket--usw2-az1--x-s3'
    :param key_name: Name of the object for which the multipart upload needs to be
    aborted
    :param upload_id: Multipart upload ID for the multipart upload to be aborted
    :return: True if the multipart upload was successfully aborted, False if not
    """
    try:
        s3_client.abort_multipart_upload(
            Bucket = bucket_name,
            Key = key_name,
            UploadId = upload_id
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    key_name = 'KEY_NAME'
    upload_id = 'UPLOAD_ID'
    s3_client = boto3.client('s3', region_name = region)
    if abort_multipart_upload(s3_client, bucket_name, key_name, upload_id):
        print (f'Multipart upload for object {key_name} in {bucket_name} bucket has
        been aborted')
    else:
        print (f'Unable to abort multipart upload for object {key_name} in
        {bucket_name} bucket')
```

Usando a AWS CLI

O exemplo a seguir mostra como interromper um carregamento fracionado usando a AWS CLI. Para usar o comando, substitua os *espaços reservados para entrada do usuário* por suas próprias informações.

```
aws s3api abort-multipart-upload --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
--upload-id
"AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAA0AAAAAAAAAAH2AfYAA
MAQAAAAB00xUFeA7LTbWWFS8WYwhrxDxTIDN-pdEEq_agIHqsbg"
```

Para obter mais informações, consulte [abort-multipart-upload](#) na AWS Command Line Interface.

Criar uma operação de cópia de carregamento multiparte

Os exemplos a seguir mostram como copiar objetos de um bucket para outro usando um carregamento fracionado.

Uso dos AWS SDKs

SDK for Java 2.x

O exemplo a seguir mostra como usar um carregamento fracionado para copiar um objeto de um bucket para outro de forma programática usando o SDK para Java 2.x.

Example

```
/**
 * This method creates a multipart upload request that generates a unique upload ID
 * that is used to track
 * all the upload parts.
 *
 * @param s3
 * @param bucketName
 * @param key
 * @return
 */
private static String createMultipartUpload(S3Client s3, String bucketName, String
key) {
    CreateMultipartUploadRequest createMultipartUploadRequest =
    CreateMultipartUploadRequest.builder()
        .bucket(bucketName)
        .key(key)
```

```

        .build();
        String uploadId = null;
        try {
            CreateMultipartUploadResponse response =
s3.createMultipartUpload(createMultipartUploadRequest);
            uploadId = response.uploadId();
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return uploadId;
    }

/**
 * Creates copy parts based on source object size and copies over individual parts
 *
 * @param s3
 * @param sourceBucket
 * @param sourceKey
 * @param destnBucket
 * @param destnKey
 * @param uploadId
 * @return
 * @throws IOException
 */
    public static List multipartUploadCopy(S3Client s3, String
sourceBucket, String sourceKey, String destnBucket, String destnKey, String
uploadId) throws IOException {

        // Get the object size to track the end of the copy operation.
        HeadObjectRequest headObjectRequest = HeadObjectRequest
            .builder()
            .bucket(sourceBucket)
            .key(sourceKey)
            .build();
        HeadObjectResponse response = s3.headObject(headObjectRequest);
        Long objectSize = response.contentLength();

        System.out.println("Source Object size: " + objectSize);

        // Copy the object using 20 MB parts.
        long partSize = 20 * 1024 * 1024;
        long bytePosition = 0;
        int partNum = 1;

```

```
ListCompletedPart completedParts = new ArrayList<>();
while (bytePosition < objectSize) {
    // The last part might be smaller than partSize, so check to make sure
    // that lastByte isn't beyond the end of the object.
    long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);

    System.out.println("part no: " + partNum + ", bytePosition: " +
bytePosition + ", lastByte: " + lastByte);

    // Copy this part.
    UploadPartCopyRequest req = UploadPartCopyRequest.builder()
        .uploadId(uploadId)
        .sourceBucket(sourceBucket)
        .sourceKey(sourceKey)
        .destinationBucket(destnBucket)
        .destinationKey(destnKey)
        .copySourceRange("bytes="+bytePosition+"-"+lastByte)
        .partNumber(partNum)
        .build();
    UploadPartCopyResponse res = s3.uploadPartCopy(req);
    CompletedPart part = CompletedPart.builder()
        .partNumber(partNum)
        .eTag(res.copyPartResult().eTag())
        .build();
    completedParts.add(part);
    partNum++;
    bytePosition += partSize;
}
return completedParts;
}

public static void multipartCopyUploadTest(S3Client s3, String srcBucket, String
srcKey, String destnBucket, String destnKey) {
    System.out.println("Starting multipart copy for: " + srcKey);
    try {
        String uploadId = createMultipartUpload(s3, destnBucket, destnKey);
        System.out.println(uploadId);
        ListCompletedPart parts = multipartUploadCopy(s3, srcBucket,
srcKey, destnBucket, destnKey, uploadId);
        completeMultipartUpload(s3, destnBucket, destnKey, uploadId, parts);
        System.out.println("Multipart copy completed for: " + srcKey);
    } catch (Exception e) {
        System.err.println(e.getMessage());
    }
}
```

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

SDK for Python

O exemplo a seguir mostra como usar um carregamento fracionado para copiar um objeto de um bucket para outro de forma programática usando o SDK para Python.

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def head_object(s3_client, bucket_name, key_name):
    """
    Returns metadata for an object in a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket that contains the object to query for metadata
    :param key_name: Key name to query for metadata
    :return: Metadata for the specified object if successful, else None
    """

    try:
        response = s3_client.head_object(
            Bucket = bucket_name,
            Key = key_name
        )
        return response
    except ClientError as e:
        logging.error(e)
        return None

def create_multipart_upload(s3_client, bucket_name, key_name):
    """
    Create a multipart upload to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Destination bucket for the multipart upload
    :param key_name: Key name of the object to be uploaded
    :return: UploadId for the multipart upload if created successfully, else None
    """
```



```

try:
    mpu = s3_client.create_multipart_upload(Bucket = bucket_name, Key =
key_name)
    return mpu['UploadId']
except ClientError as e:
    logging.error(e)
    return None

def multipart_copy_upload(s3_client, source_bucket_name, key_name,
target_bucket_name, mpu_id, part_size):
    '''
    Copy an object in a directory bucket to another bucket in multiple parts of a
specified size

:param s3_client: boto3 S3 client
:param source_bucket_name: Bucket where the source object exists
:param key_name: Key name of the object to be copied
:param target_bucket_name: Destination bucket for copied object
:param mpu_id: The UploadId returned from the create_multipart_upload call
:param part_size: The size parts that the object will be broken into, in bytes.
                    Minimum 5 MiB, Maximum 5 GiB. There is no minimum size for the
last part of your multipart upload.
:return: part_list for the multipart copy if all parts are copied successfully,
else None
    '''

    part_list = []
    copy_source = {
        'Bucket': source_bucket_name,
        'Key': key_name
    }
    try:
        part_counter = 1
        object_size = head_object(s3_client, source_bucket_name, key_name)
        if object_size is not None:
            object_size = object_size['ContentLength']
            while (part_counter - 1) * part_size < object_size:
                bytes_start = (part_counter - 1) * part_size
                bytes_end = (part_counter * part_size) - 1
                upload_copy_part = s3_client.upload_part_copy (
                    Bucket = target_bucket_name,
                    CopySource = copy_source,
                    CopySourceRange = f'bytes={bytes_start}-{bytes_end}',

```

```

        Key = key_name,
        PartNumber = part_counter,
        UploadId = mpu_id
    )
    part_list.append({'PartNumber': part_counter, 'ETag':
upload_copy_part['CopyPartResult']['ETag']})
    part_counter += 1
except ClientError as e:
    logging.error(e)
    return None
return part_list

def complete_multipart_upload(s3_client, bucket_name, key_name, mpu_id, part_list):
    """
    Completes a multipart upload to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Destination bucket for the multipart upload
    :param key_name: Key name of the object to be uploaded
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_list: List of uploaded part numbers with associated ETags
    :return: True if the multipart upload was completed successfully, else False
    """

    try:
        s3_client.complete_multipart_upload(
            Bucket = bucket_name,
            Key = key_name,
            UploadId = mpu_id,
            MultipartUpload = {
                'Parts': part_list
            }
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    MB = 1024 ** 2
    region = 'us-west-2'
    source_bucket_name = 'SOURCE_BUCKET_NAME'
    target_bucket_name = 'TARGET_BUCKET_NAME'
    key_name = 'KEY_NAME'

```

```

part_size = 10 * MB
s3_client = boto3.client('s3', region_name = region)
mpu_id = create_multipart_upload(s3_client, target_bucket_name, key_name)
if mpu_id is not None:
    part_list = multipart_copy_upload(s3_client, source_bucket_name, key_name,
target_bucket_name, mpu_id, part_size)
    if part_list is not None:
        if complete_multipart_upload(s3_client, target_bucket_name, key_name,
mpu_id, part_list):
            print (f'{key_name} successfully copied through multipart copy from
{source_bucket_name} to {target_bucket_name}')
        else:
            print (f'Could not copy {key_name} through multipart copy from
{source_bucket_name} to {target_bucket_name}')

```

Usando a AWS CLI

O exemplo a seguir mostra como usar um carregamento fracionado para copiar um objeto de um bucket para outro de forma programática usando a AWS CLI. Para usar o comando, substitua os *espaços reservados para entrada do usuário* por suas próprias informações.

```

aws s3api upload-part-copy --bucket bucket-base-name--azid--x-s3 --key TARGET_KEY_NAME
--copy-source SOURCE_BUCKET_NAME/SOURCE_KEY_NAME --part-number 1 --upload-id
"AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAAAEMAAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAAAA0AAAAAAAAAAAAH2AfYAA"

```

Para obter mais informações, consulte [upload-part-copy](#) na AWS Command Line Interface.

Listar os uploads multipart em andamento

Para listar carregamentos fracionados em andamento em um bucket de diretório, é possível usar os SDKs da AWS ou a AWS CLI.

Uso dos AWS SDKs

SDK for Java 2.x

Os exemplos a seguir mostram como listar carregamentos fracionados em andamento (incompletos) usando o SDK para Java 2.x.

Example

```

public static void listMultiPartUploads( S3Client s3, String bucketName) {

```

```

    try {
        ListMultipartUploadsRequest listMultipartUploadsRequest =
ListMultipartUploadsRequest.builder()
        .bucket(bucketName)
        .build();

        ListMultipartUploadsResponse response =
s3.listMultipartUploads(listMultipartUploadsRequest);
        List MultipartUpload uploads = response.uploads();
        for (MultipartUpload upload: uploads) {
            System.out.println("Upload in progress: Key = \"" + upload.key() +
"\", id = " + upload.uploadId());
        }
    }
    catch (S3Exception e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
}

```

SDK for Python

Os exemplos a seguir mostram como listar carregamentos fracionados em andamento (incompletos) usando o SDK para Python.

Example

```

import logging
import boto3
from botocore.exceptions import ClientError

def list_multipart_uploads(s3_client, bucket_name):
    """
    List any incomplete multipart uploads in a directory bucket in e specified gion

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket to check for incomplete multipart uploads
    :return: List of incomplete multipart uploads if there are any, None if not
    """

    try:
        response = s3_client.list_multipart_uploads(Bucket = bucket_name)
        if 'Uploads' in response.keys():

```

```
        return response['Uploads']
    else:
        return None
except ClientError as e:
    logging.error(e)

if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'us-west-2'
    s3_client = boto3.client('s3', region_name = region)
    multipart_uploads = list_multipart_uploads(s3_client, bucket_name)
    if multipart_uploads is not None:
        print (f'There are {len(multipart_uploads)} ncomplete multipart uploads for
{bucket_name}')
    else:
        print (f'There are no incomplete multipart uploads for {bucket_name}')
```

Usando a AWS CLI

Os exemplos a seguir mostram como listar carregamentos fracionados em andamento (incompletos) usando a AWS CLI. Para usar o comando, substitua os *espaços reservados para entrada do usuário* por suas próprias informações.

```
aws s3api list-multipart-uploads --bucket bucket-base-name--azid--x-s3
```

Para obter mais informações, consulte [list-multipart-uploads](#) na AWS Command Line Interface.

Listar as partes de um upload multiparte

Os exemplos a seguir mostram como listar partes de um carregamento fracionado em um bucket de diretório.

Uso dos AWS SDKs

SDK for Java 2.x

Os exemplos a seguir mostram como listar partes de um carregamento fracionado em um bucket de diretório usando o SDK para Java 2.x.

```
public static void listMultiPartUploadsParts( S3Client s3, String bucketName, String
objKey, String uploadID) {
```

```

    try {
        ListPartsRequest listPartsRequest = ListPartsRequest.builder()
            .bucket(bucketName)
            .uploadId(uploadID)
            .key(objKey)
            .build();

        ListPartsResponse response = s3.listParts(listPartsRequest);
        ListPart parts = response.parts();
        for (Part part: parts) {
            System.out.println("Upload in progress: Part number = \" +
part.partNumber() + "\", etag = \" + part.eTag());
        }

    }

    catch (S3Exception e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }

}

```

SDK for Python

Os exemplos a seguir mostram como listar partes de um carregamento fracionado em um bucket de diretório usando o SDK para Python.

```

import logging
import boto3
from botocore.exceptions import ClientError

def list_parts(s3_client, bucket_name, key_name, upload_id):
    """
    Lists the parts that have been uploaded for a specific multipart upload to a
    directory bucket.

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket that multipart uploads parts have been uploaded to
    :param key_name: Name of the object that has parts uploaded
    :param upload_id: Multipart upload ID that the parts are associated with
    """

```

```
:return: List of parts associated with the specified multipart upload, None if
there are no parts
'''
parts_list = []
next_part_marker = ''
continuation_flag = True
try:
    while continuation_flag:
        if next_part_marker == '':
            response = s3_client.list_parts(
                Bucket = bucket_name,
                Key = key_name,
                UploadId = upload_id
            )
        else:
            response = s3_client.list_parts(
                Bucket = bucket_name,
                Key = key_name,
                UploadId = upload_id,
                NextPartMarker = next_part_marker
            )
        if 'Parts' in response:
            for part in response['Parts']:
                parts_list.append(part)
            if response['IsTruncated']:
                next_part_marker = response['NextPartNumberMarker']
            else:
                continuation_flag = False
        else:
            continuation_flag = False
    return parts_list
except ClientError as e:
    logging.error(e)
    return None

if __name__ == '__main__':
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    key_name = 'KEY_NAME'
    upload_id = 'UPLOAD_ID'
    s3_client = boto3.client('s3', region_name = region)
    parts_list = list_parts(s3_client, bucket_name, key_name, upload_id)
    if parts_list is not None:
        print (f'{key_name} has {len(parts_list)} parts uploaded to {bucket_name}')
```

```
else:
    print (f'There are no multipart uploads with that upload ID for
{bucket_name} bucket')
```

Usando a AWS CLI

Os exemplos a seguir mostram como listar partes de um carregamento fracionado em um bucket de diretório usando a AWS CLI. Para usar o comando, substitua os *espaços reservados para entrada do usuário* por suas próprias informações.

```
aws s3api list-parts --bucket bucket-base-name--azid--x-s3 --key KEY_NAME --upload-id
"AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
```

Para obter mais informações, consulte [list-parts](#) na AWS Command Line Interface.

Copiar um objeto para um bucket de diretório

A operação de cópia cria uma cópia de um objeto que já está armazenado no Amazon S3. Você pode copiar objetos entre buckets de diretório e buckets de uso geral. Você também pode copiar objetos dentro de um bucket e entre buckets do mesmo tipo, por exemplo, de um bucket de diretório para outro bucket de diretório.

Você pode criar uma cópia do objeto de até 5 GB em uma única operação atômica. Contudo, para copiar um objeto maior do que 5 GB, você deve usar as operações de API de upload multiparte. Para ter mais informações, consulte [Usar uploads multiparte com buckets de diretório](#).

Permissões

Para copiar objetos, você deve ter as seguintes permissões:

- Para copiar objetos de um bucket de diretório para outro bucket de diretório, você precisa ter a permissão `s3express:CreateSession`.
- Para copiar objetos de buckets de diretório para buckets de uso geral, você precisa ter a permissão `s3express:CreateSession` e a permissão `s3:PutObject` para gravar a cópia do objeto no bucket de destino.
- Para copiar objetos de buckets de uso geral para buckets de diretório, você precisa ter a permissão `s3express:CreateSession` e a permissão `s3:GetObject` para ler o objeto de origem que está sendo copiado.

Para obter mais informações, consulte [CopyObject](#) na Referência da API do Amazon Simple Storage Service.

Criptografia

O Amazon S3 criptografa automaticamente todos os novos objetos que são carregados em um bucket do S3. A configuração de criptografia padrão de um bucket do S3 está sempre habilitada e, no mínimo, definida como criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3).

Buckets de diretório só são compatíveis com SSE-S3. Para buckets de uso geral, você pode usar SSE-S3 (padrão), criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS), criptografia de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS) ou criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C).

Se você fizer uma solicitação de cópia que defina os parâmetros SSE-C, SSE-KMS ou DSSE-KMS em um bucket de diretório como origem ou destino, a resposta retornará um erro.

Tags

Os buckets de diretório não oferecem suporte a etiquetas. Se você copiar um objeto que tenha etiquetas de um bucket de uso geral para um bucket de diretório, receberá uma resposta HTTP 501 (Not Implemented). Para obter mais informações, consulte [CopyObject](#) na Referência da API do Amazon Simple Storage Service.

ETags

As etiquetas de entidade (ETags) para a classe S3 Express One Zone são strings alfanuméricas aleatórias e não somas de verificação MD5. Para ajudar a garantir a integridade do objeto, use somas de verificação adicionais.

Somas de verificação adicionais

A classe S3 Express One Zone oferece a opção de escolher o algoritmo de soma de verificação usado para validar os 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 cíclica (CRC): CRC32, CRC32C, SHA-1 e SHA-256. As somas de verificação baseadas em MD5 não são compatíveis com a classe de armazenamento S3 Express One Zone.

Para ter mais informações, consulte [Práticas recomendadas adicionais de soma de verificação do S3](#).

Atributos compatíveis

Para obter mais informações sobre quais recursos do Amazon S3 são compatíveis com a classe S3 Express One Zone, consulte [O que diferencia a classe S3 Express One Zone?](#).

Usar o console do S3 (copiar para um bucket de diretório)

Como copiar um objeto de um bucket de uso geral ou de diretório para um bucket de diretório

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. Escolha o bucket do qual deseja copiar objetos:
 - Para copiar de um bucket de uso geral, escolha a guia Buckets de uso geral.
 - Para copiar de um bucket de diretório, escolha a guia Buckets de diretórios.
4. Escolha o bucket de uso geral ou de diretório que contém os objetos que você deseja copiar.
5. Escolha a guia Objetos. Na página Objetos, marque a caixa de seleção à esquerda dos nomes dos objetos que você deseja copiar.
6. No menu Actions (Ações), escolha Copy (Copiar).

A página Copiar é exibida.

7. Em Destino, escolha Bucket de diretório como tipo de destino. Para especificar o caminho de destino, escolha Procurar no S3, navegue até o destino e selecione o botão de opções à esquerda do destino. Escolha Choose destination (Escolher destino) no canto inferior direito.

Outra alternativa é inserir o caminho de destino.

8. Em Checksums, escolha se deseja copiar os objetos com suas funções de soma de verificação existentes ou substituí-las por funções novas. 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 uma soma de verificação adicional, poderá usar a seção Checksums para adicionar uma.

Note

Mesmo se optar por usar a mesma função de soma de verificação, o valor dela poderá mudar se o objeto 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](#).

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 lista suspensa. Quando o objeto é copiado, a nova soma de verificação é calculada e armazenada usando o algoritmo especificado.

9. Escolha Copy (Copiar) no canto inferior direito. O Amazon S3 copia o objeto no destino.

Usar o console do S3 (copiar para um bucket de uso geral)

Como copiar um objeto de um bucket de diretório para um bucket de uso geral

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. Escolha a guia Buckets de diretórios.
4. Selecione o bucket de diretório que contém os objetos que você deseja copiar.
5. Escolha a guia Objetos. Na página Objetos, marque a caixa de seleção à esquerda dos nomes dos objetos que você deseja copiar.
6. No menu Actions (Ações), escolha Copy (Copiar).
7. Em Destino, escolha Bucket de uso geral como tipo de destino. Para especificar o caminho de destino, escolha Procurar no S3, navegue até o destino e selecione o botão de opções à esquerda do destino. Escolha Choose destination (Escolher destino) no canto inferior direito.

Outra alternativa é inserir o caminho de destino.

8. Em Checksums, escolha se deseja copiar os objetos com suas funções de soma de verificação existentes ou substituí-las por funções novas. 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 uma soma de verificação adicional, poderá usar a seção Checksums para adicionar uma.

 Note

Mesmo se optar por usar a mesma função de soma de verificação, o valor dela poderá mudar se o objeto 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](#).

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 lista suspensa. Quando o objeto é copiado, a nova soma de verificação é calculada e armazenada usando o algoritmo especificado.

9. Escolha Copy (Copiar) no canto inferior direito. O Amazon S3 copia o objeto no destino.

Uso dos AWS SDKs

SDK for Java 2.x

Example

```
public static void copyBucketObject (S3Client s3, String sourceBucket, String
objectKey, String targetBucket) {
    CopyObjectRequest copyReq = CopyObjectRequest.builder()
        .sourceBucket(sourceBucket)
        .sourceKey(objectKey)
        .destinationBucket(targetBucket)
        .destinationKey(objectKey)
        .build();
    String temp = "";

    try {
```

```
CopyObjectResponse copyRes = s3.copyObject(copyReq);
System.out.println("Successfully copied " + objectKey + " from bucket " +
sourceBucket + " into bucket "+targetBucket);
}

catch (S3Exception e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
}
```

Usando a AWS CLI

O exemplo de comando `copy-object` a seguir mostra como é possível usar a AWS CLI para copiar um objeto de um bucket para outro. Você pode copiar objetos entre tipos de bucket. Para executar esse comando, substitua os espaços reservados de entrada por suas próprias informações.

```
aws s3api copy-object --copy-source bucket SOURCE_BUCKET/SOURCE_KEY_NAME --
key TARGET_KEY_NAME --bucket TARGET_BUCKET_NAME
```

Para obter mais informações, consulte [copy-object](#) na Referência de comandos da AWS CLI.

Excluir um objeto de um bucket de diretório

Você pode excluir objetos de um bucket de diretório do Amazon S3 usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI) ou os AWS SDKs. Para obter mais informações, consulte [Buckets de diretório](#) e [O que é a classe S3 Express One Zone?](#).

Warning

- Não é possível desfazer a exclusão de um objeto.
- 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.

Note

Ao excluir vários objetos de um bucket de diretório de forma programática, observe o seguinte:

- As chaves de objeto em solicitações `DeleteObjects` devem conter pelo menos um caractere diferente de um espaço em branco. As strings que têm somente espaços em branco não são compatíveis.
- As chaves de objeto em solicitações `DeleteObjects` não podem conter caracteres de controle Unicode, exceto nova linha (`\t`), tabulação (`\n`) e avanço de linha (`\r`).

Uso do console do S3

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. No painel de navegação à esquerda, escolha Buckets.
3. Escolha a guia Buckets de diretórios.
4. Selecione o bucket de diretório que contém os objetos que você deseja excluir.
5. Escolha a guia Objetos. Na lista Objetos, marque as caixas de seleção à esquerda dos objetos que você deseja excluir.
6. Escolha Excluir.
7. Na página Excluir objetos, insira **permanently delete** na caixa de texto.
8. Escolha Delete objects (Excluir objetos).

Uso dos AWS SDKs

SDK for Java 2.x

Example

O exemplo a seguir exclui objetos de um bucket de diretório usando o AWS SDK for Java 2.x.

```
static void deleteObject(S3Client s3Client, String bucketName, String objectKey) {
```

```
try {  
  
    DeleteObjectRequest del = DeleteObjectRequest.builder()  
        .bucket(bucketName)  
        .key(objectKey)  
        .build();  
  
    s3Client.deleteObject(del);  
  
    System.out.println("Object " + objectKey + " has been deleted");  
  
} catch (S3Exception e) {  
    System.err.println(e.awsErrorDetails().errorMessage());  
    System.exit(1);  
}  
  
}
```

SDK for Python

Example

O exemplo a seguir exclui objetos de um bucket de diretório usando o AWS SDK for Python (Boto3).

```
import logging  
import boto3  
from botocore.exceptions import ClientError  
  
def delete_objects(s3_client, bucket_name, objects):  
    """  
    Delete a list of objects in a directory bucket  
  
    :param s3_client: boto3 S3 client  
    :param bucket_name: Bucket that contains objects to be deleted; for example,  
    'doc-example-bucket--usw2-az1--x-s3'  
    :param objects: List of dictionaries that specify the key names to delete  
    :return: Response output, else False  
    """
```

```
try:
    response = s3_client.delete_objects(
        Bucket = bucket_name,
        Delete = {
            'Objects': objects
        }
    )
    return response
except ClientError as e:
    logging.error(e)
    return False

if __name__ == '__main__':
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    objects = [
        {
            'Key': '0.txt'
        },
        {
            'Key': '1.txt'
        },
        {
            'Key': '2.txt'
        },
        {
            'Key': '3.txt'
        },
        {
            'Key': '4.txt'
        }
    ]

    s3_client = boto3.client('s3', region_name = region)
    results = delete_objects(s3_client, bucket_name, objects)
    if results is not None:
        if 'Deleted' in results:
            print (f'Deleted {len(results["Deleted"])} objects from {bucket_name}')
        if 'Errors' in results:
            print (f'Failed to delete {len(results["Errors"])} objects from
{bucket_name}')
```


Usando a AWS CLI

O exemplo de comando `delete-object` a seguir mostra como você pode usar a AWS CLI para excluir um objeto de um bucket. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3api delete-object --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
```

Para obter mais informações, consulte [delete-object](#) na Referência de comandos da AWS CLI.

Baixar um objeto de um bucket de diretório

Os exemplos de código a seguir mostram como ler dados de um objeto (download) em um bucket de diretório do Amazon S3 usando a operação de API `GetObject`.

Uso dos AWS SDKs

SDK for Java 2.x

Example

O exemplo de código a seguir mostra como ler dados de um objeto em um bucket de diretório usando o AWS SDK for Java 2.x.

```
public static void getObject(S3Client s3Client, String bucketName, String objectKey)
{
    try {
        GetObjectRequest objectRequest = GetObjectRequest
            .builder()
            .key(objectKey)
            .bucket(bucketName)
            .build();

        ResponseBytes GetObjectResponse objectBytes =
s3Client.getObjectAsBytes(objectRequest);
        byte[] data = objectBytes.asByteArray();

        //Print object contents to console
        String s = new String(data, StandardCharsets.UTF_8);
        System.out.println(s);
    }
}
```

```
catch (S3Exception e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
}
```

SDK for Python

Example

O exemplo de código a seguir mostra como ler dados de um objeto em um bucket de diretório usando o AWS SDK for Python (Boto3).

```
import boto3
from botocore.exceptions import ClientError
from botocore.response import StreamingBody

def get_object(s3_client: boto3.client, bucket_name: str, key_name: str) ->
StreamingBody:
    """
    Gets the object.
    :param s3_client:
    :param bucket_name: The bucket that contains the object.
    :param key_name: The key of the object to be downloaded.
    :return: The object data in bytes.
    """
    try:
        response = s3_client.get_object(Bucket=bucket_name, Key=key_name)
        body = response['Body'].read()
        print(f"Got object '{key_name}' from bucket '{bucket_name}'.")
    except ClientError:
        print(f"Couldn't get object '{key_name}' from bucket '{bucket_name}'.")
        raise
    else:
        return body

def main():
    s3_client = boto3.client('s3')
    resp = get_object(s3_client, 'doc-example-bucket--use1-az4--x-s3', 'sample.txt')
    print(resp)

if __name__ == "__main__":
```

```
main()
```

Usando a AWS CLI

O exemplo de `get-object` a seguir mostra como você pode usar a AWS CLI para baixar um objeto do Amazon S3. Esse comando obtém o objeto `KEY_NAME` do bucket de diretório `bucket-base-name--azid--x-s3`. O objeto será baixado em um arquivo chamado `LOCAL_FILE_NAME`. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3api get-object --bucket bucket-base-name--azid--x-s3 --  
key KEY_NAME LOCAL_FILE_NAME
```

Para obter mais informações, consulte [get-object](#) na Referência de comandos da AWS CLI.

Usar **HeadObject** com buckets de diretório

Os exemplos a seguir de SDK da AWS e da AWS CLI mostram como usar a operação de API `HeadObject` para recuperar metadados de um objeto em um bucket de diretório do Amazon S3 sem retornar o objeto em si.

Uso dos AWS SDKs

SDK for Java 2.x

Example

```
public static void headObject(S3Client s3Client, String bucketName, String  
objectKey) {  
    try {  
        HeadObjectRequest headObjectRequest = HeadObjectRequest  
            .builder()  
            .bucket(bucketName)  
            .key(objectKey)  
            .build();  
        HeadObjectResponse response = s3Client.headObject(headObjectRequest);  
        System.out.format("Amazon S3 object: \"%s\" found in bucket: \"%s\" with  
ETag: \"%s\"", objectKey, bucketName, response.eTag());  
    }  
    catch (S3Exception e) {  
        System.err.println(e.awsErrorDetails().errorMessage());  
    }  
}
```

Usando a AWS CLI

O exemplo de comando `head-object` a seguir mostra como você pode usar a AWS CLI para recuperar metadados de um objeto. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3api head-object --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
```

Para obter mais informações, consulte [head-object](#) na Referência de comandos da AWS CLI.

Segurança para a classe S3 Express One Zone

A segurança para com a 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 isto 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 [AWS Compliance Programs](#).

Para saber mais sobre os programas de conformidade que se aplicam à classe Amazon S3 Express One Zone, consulte [Serviços da AWS in Scope by Compliance Program](#).

- **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 dos dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajudará você a entender como aplicar o modelo de responsabilidade compartilhada ao usar a classe S3 Express One Zone. Os tópicos a seguir mostram como configurar a classe S3 Express One Zone para atender aos seus objetivos de segurança e conformidade. Você também aprenderá como usar outros Serviços da AWS que podem ajudar a monitorar e proteger os recursos ao trabalhar com a classe S3 Express One Zone.

Tópicos

- [Proteção e criptografia de dados](#)

- [AWS Identity and Access Management \(IAM\) para a classe S3 Express One Zone](#)
- [Políticas baseadas em identidade do IAM para a classe S3 Express One Zone](#)
- [Exemplos de políticas de bucket de diretório para a classe S3 Express One Zone](#)
- [CreateSession authorization](#)
- [Práticas recomendadas de segurança para a classe S3 Express One Zone](#)

Proteção e criptografia de dados

Para obter mais informações sobre como a classe S3 Express One Zone criptografa e protege os dados, consulte os tópicos a seguir.

Tópicos

- [Criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#)
- [Criptografia em trânsito](#)
- [Somadas de verificação adicionais](#)
- [Exclusão de dados](#)

Criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3)

Por padrão, todos os objetos armazenados em buckets de diretório são criptografados automaticamente usando criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3). Não são permitidos uploads não criptografados em buckets de diretório. Para obter mais informações, consulte [Usar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#) e [Proteger dados com criptografia](#).

Os buckets de diretório não oferecem suporte à criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS), criptografia de camada dupla do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (DSSE-KMS) ou criptografia do lado do servidor com chaves de criptografia fornecidas pelo cliente (SSE-C).

Criptografia em trânsito

A classe S3 Express One Zone só pode ser acessada por meio de HTTPS (TLS).

A classe S3 Express One Zone usa endpoints de API regionais e zonais. Dependendo da operação de API do Amazon S3 usada, será necessário um endpoint regional ou zonal. Você pode acessar endpoints zonais e regionais por meio de um endpoint de nuvem privada virtual (VPC) do gateway.

Não há cobrança adicional pelo uso de endpoints do gateway. Para saber mais sobre os endpoints de API regionais e zonais, consulte [Redes para a classe S3 Express One Zone](#).

Somas de verificação adicionais

A classe S3 Express One Zone oferece a opção de escolher o algoritmo de soma de verificação usado para validar os 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 cíclica (CRC): CRC32, CRC32C, SHA-1 e SHA-256. As somas de verificação baseadas em MD5 não são compatíveis com a classe de armazenamento S3 Express One Zone.

Para ter mais informações, consulte [Práticas recomendadas adicionais de soma de verificação do S3](#).

Exclusão de dados

Você pode excluir um ou mais objetos diretamente da classe S3 Express One Zone usando o console do Amazon S3, os AWS SDKs, a AWS Command Line Interface (AWS CLI) ou a API REST do Amazon S3. Como todos os objetos nos buckets de diretório incorrem em custos de armazenamento, recomendamos que você exclua os objetos de que não precisa mais.

A exclusão de um objeto armazenado em um bucket de diretório também exclui recursivamente todos os diretórios pai, desde que esses diretórios pai não contenham nenhum objeto além do objeto que está sendo excluído.

Note

A exclusão de autenticação multifator (MFA) e a funcionalidade Versionamento do S3 não são compatíveis com a classe S3 Express One Zone.

AWS Identity and Access Management (IAM) para a classe S3 Express One Zone

O AWS Identity and Access Management (IAM) é um AWS service (Serviço da AWS) que ajuda os administradores a controlar o acesso aos recursos da AWS de forma segura. Os administradores do IAM controlam quem pode se autenticar (se conectar) e ter autorização (permissões) para utilizar os recursos do Amazon S3 na classe S3 Express One Zone. Você pode usar o IAM sem custo adicional.

Por padrão, os usuários não têm permissões para os buckets de diretório e as operações da classe S3 Express One Zone. Para conceder permissões de acesso a buckets de diretório, você pode usar o IAM para criar usuários ou perfis e anexar permissões a essas identidades. Para obter mais informações sobre o IAM, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Para fornecer acesso, você pode adicionar permissões aos usuários, grupos ou perfis das seguintes formas:

- Usuários e grupos no AWS IAM Identity Center: crie um conjunto de permissões. Siga as instruções em [Criação de um conjunto de permissões](#) no Guia do usuário do AWS IAM Identity Center.
- Usuários gerenciados no IAM por meio de um provedor de identidades: crie um perfil para federação de identidades. Siga as instruções em [Criar um perfil para um provedor de identidades de terceiros \(federação\)](#) no Guia do usuário do IAM.
- Perfis e usuários do IAM: crie um perfil que o usuário possa assumir. Siga as instruções em [Criação de uma função para delegar permissões a um usuário do IAM](#) no Guia do usuário do IAM.

Por padrão, os buckets de diretório são privados e só podem ser acessados por usuários que tenham recebido acesso explícito. O limite de controle de acesso para buckets de diretório é definido somente no nível do bucket. Por outro lado, o limite de controle de acesso para buckets de uso geral pode ser definido no nível do bucket, do prefixo ou da etiqueta de objeto. Essa diferença significa que os buckets de diretório são o único recurso que você pode incluir nas políticas de bucket ou nas políticas de identidade do IAM para acesso à classe S3 Express One Zone.

Com a classe S3 Express One Zone, além da autorização do IAM, você pode autenticar e autorizar solicitações por meio de um mecanismo com base em nova sessão que é processado pela operação da API `CreateSession`. Você pode usar `CreateSession` para solicitar credenciais temporárias que forneçam acesso de baixa latência ao bucket. Essas credenciais temporárias têm como escopo um bucket de diretório específico.

Para trabalhar com `CreateSession`, recomendamos usar a versão mais recente dos AWS SDKs ou usar a AWS Command Line Interface (AWS CLI). Os AWS SDKs compatíveis e a AWS CLI lidam com o estabelecimento, a atualização e o encerramento de sessões em seu nome.

Use tokens de sessão somente com operações zonais (nível de objeto) (exceto para `CopyObject` e `HeadBucket`) para distribuir a latência associada à autorização em várias solicitações em uma sessão. Para as operações de API do endpoint regional (operações em nível de bucket), use a

autorização do IAM, que não envolve o gerenciamento de uma sessão. Para obter mais informações, consulte [AWS Identity and Access Management \(IAM\) para a classe S3 Express One Zone](#) e [CreateSession authorization](#).

Para obter mais informações sobre o IAM para a classe S3 Express One Zone, consulte os tópicos a seguir.

Tópicos

- [Entidades principais](#)
- [Recursos](#)
- [Ações para a classe S3 Express One Zone](#)
- [Chaves de condição da classe S3 Express One Zone](#)
- [Como as operações de API são autorizadas e autenticadas](#)

Entidades principais

Ao criar uma política baseada em recurso para conceder acesso aos buckets, você deve usar o elemento `Principal` para especificar a pessoa ou a aplicação que pode fazer uma solicitação de uma ação ou operação no respectivo recurso. Para políticas de bucket de diretório, você pode usar as seguintes entidades principais:

- Uma conta da AWS
- Um usuário do IAM
- Um perfil do IAM
- Um usuário federado

Para obter mais informações, consulte [Principal](#) no Guia do usuário do IAM.

Recursos

Os nomes de recurso da Amazon (ARNs) para buckets de diretório contêm o namespace `s3express`, a Região da AWS, o ID da conta da AWS e o nome do bucket de diretório, que inclui o ID da zona de disponibilidade. Para acessar e executar ações no bucket de diretório, você deve usar o seguinte formato de ARN:

```
arn:aws:s3express:region:account-id:bucket/base-bucket-name--azid--x-s3
```


Para obter mais informações sobre os ARNs, consulte [Amazon Resource Names \(ARNs\)](#) no Guia do usuário do IAM. Para obter mais informações sobre os recursos, consulte [Elementos de política JSON do IAM: Resource](#) no Guia do usuário do IAM.

Ações para a classe S3 Express One Zone

Em uma política baseada em identidade ou recurso do IAM, defina quais ações do S3 serão permitidas ou negadas. As ações da classe S3 Express One Zone correspondem a operações específicas da API. A classe S3 Express One Zone tem um namespace do IAM exclusivo que é distinto do namespace padrão do Amazon S3. Esse namespace é o `s3express`.

Quando você concede a permissão `s3express:CreateSession`, isso permite que a operação de API `CreateSession` recupere tokens de sessão ao acessar operações de API de endpoint zonal (ou de nível de objeto). Esses tokens de sessão retornam credenciais que são usadas para conceder acesso a todas as outras operações de API de endpoint zonal. Como resultado, você não precisa conceder permissões de acesso às operações de API zonais usando políticas do IAM. Em vez disso, o token de sessão permite o acesso.

Para obter mais informações sobre as operações de API de endpoints zonais e regionais, consulte [Redes para a classe S3 Express One Zone](#). Para saber mais sobre a operação de API `CreateSession`, consulte a [CreateSession](#) na Referência da API do Amazon Simple Storage Service.

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. Ao usar uma ação em uma política, você geralmente permite ou nega acesso à operação de API com o mesmo nome. No entanto, em alguns casos, uma única ação controla o acesso a mais de uma operação de API. O acesso às ações no nível do bucket só pode ser concedido em políticas baseadas em identidade do IAM (usuário ou perfil), não em políticas de bucket.

Ações e chaves de condição da classe S3 Express One Zone

Ação	API	Descrição	Nível de acesso	Chaves de condição
<code>s3express:CreateBucket</code>	<code>CreateBucket</code>	Concede permissão para criar um bucket.	Escrever	<code>s3express:authType</code>

Ação	API	Descrição	Nível de acesso	Chaves de condição
				s3express :Location Name s3express :Resource Account s3express :signatur eversion s3express :TlsVersi on s3express :x-amz-co ntent-sha 256

Ação	API	Descrição	Nível de acesso	Chaves de condição
s3express:CreateSession	CreateSession	Concede permissão para criar um token de sessão, que é usado para conceder acesso a todas as operações de API zonais (nível de objeto), como PutObject, GetObject e assim por diante.	Escrever	s3express:authType s3express:SessionMode s3express:ResourceAccount s3express:signatureversion s3express:signatureAge s3express:TlsVersion s3express:x-amz-content-sha256

Ação	API	Descrição	Nível de acesso	Chaves de condição
s3express:DeleteBucket	DeleteBucket	Concede permissão para excluir o bucket nomeado no URI.	Escrever	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Ação	API	Descrição	Nível de acesso	Chaves de condição
s3express:DeleteBucketPolicy	DeleteBucketPolicy	Concede permissão para excluir a política de um bucket especificado.	Gerenciamento de permissões	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Ação	API	Descrição	Nível de acesso	Chaves de condição
s3express:GetBucketPolicy	GetBucketPolicy	Concede permissão para retornar a política do bucket especificado.	Leitura	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Ação	API	Descrição	Nível de acesso	Chaves de condição
s3express:ListAllMyDirectoriesBuckets	ListDirectoryBuckets	Concede permissão para listar todos os buckets de diretório de propriedade do remetente autenticado da solicitação.	Lista	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Ação	API	Descrição	Nível de acesso	Chaves de condição
<code>s3express:PutBucketPolicy</code>	PutBucketPolicy	Concede permissão para adicionar ou substituir uma política de um bucket.	Gerenciamento de permissões	<code>s3express:authType</code> <code>s3express:ResourceAccount</code> <code>s3express:signatureversion</code> <code>s3express:TlsVersion</code> <code>s3express:x-amz-content-sha256</code>

Chaves de condição da classe S3 Express One Zone

A classe S3 Express One Zone define as chaves de condição a seguir 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.

Chave de condição	Descrição	Tipo
<code>s3express:authType</code>	Filtra o acesso por método 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	String

Chave de condição	Descrição	Tipo
	<p>de condição para permitir que somente o cabeçalho HTTP <code>Authorization</code> seja usado na autenticação de solicitação.</p> <p>Valores válidos: <code>REST-HEADER</code> , <code>REST-QUERY-STRING</code></p>	
<code>s3express:LocationName</code>	<p>Filtra o acesso à operação de API <code>CreateBucket</code> pelo ID de uma zona de disponibilidade específica (ID de AZ), como <code>usw2-az1</code>.</p> <p>Exemplo de valor: <code>usw2-az1</code></p>	String
<code>s3express:ResourceAccount</code>	<p>Filtra o acesso pelo ID da Conta da AWS do proprietário do recurso.</p> <p>Para restringir o acesso de usuários, perfis ou aplicações aos buckets de diretório pertencentes ao ID de uma Conta da AWS específica, é possível usar a chave de condição <code>aws:ResourceAccount</code> ou <code>s3express:ResourceAccount</code> . Você pode usar essa chave de condição em políticas de identidade do AWS Identity and Access Management (IAM) ou políticas de endpoint de nuvem privada virtual (VPC). Por exemplo, você pode usar essa chave de condição para restringir o acesso dos clientes dentro da VPC aos buckets que não pertencem a você.</p> <p>Exemplo de valor: <code>111122223333</code></p>	String

Chave de condição	Descrição	Tipo
<code>s3express:SessionMode</code>	<p>Filtra o acesso pela permissão solicitada pela operação de API <code>CreateSession</code> . Por padrão, a sessão é <code>ReadWrite</code> . Você pode usar essa chave de condição para limitar o acesso a <code>ReadOnly</code> ou negar explicitamente o acesso a <code>ReadWrite</code> . Para obter mais informações, consulte Exemplos de políticas de bucket de diretório para a classe S3 Express One Zone e CreateSession na Referência da API do Amazon Simple Storage Service.</p> <p>Valores válidos: <code>ReadWrite</code> , <code>ReadOnly</code></p>	String
<code>s3express:signatureAge</code>	<p>Filtra o acesso pela idade em milissegundos da assinatura da solicitação. Essa condição funciona somente para URLs pré-assinados.</p> <p>No AWS 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>Exemplo de valor: <code>600000</code></p>	Numérico

Chave de condição	Descrição	Tipo
<code>s3express:signatureversion</code>	<p>Identifica a versão do AWS Signature à qual você deseja oferecer suporte para solicitações autenticadas. Para solicitações autenticadas, a classe S3 Express One Zone oferece suporte ao Signature Version 4.</p> <p>Valor válido: "AWS4-HMAC-SHA256" (identifica o Signature Version 4)</p>	String
<code>s3express:TlsVersion</code>	<p>Filtra o acesso pela versão do TLS usada pelo cliente.</p> <p>Você pode usar a chave de condição <code>s3:TlsVersion</code> para gravar políticas do IAM, de endpoint de nuvem privada virtual (VPCE) ou de bucket que restrinjam o acesso de usuários ou aplicações aos buckets de diretório com base na versão do TLS usada pelo cliente. Você também pode usar essa chave de condição para gravar políticas que exigem uma versão mínima do TLS.</p> <p>Exemplo de valor: 1.3</p>	Numérico

Chave de condição	Descrição	Tipo
s3express:x-amz-content-sha256	<p>Filtra o acesso por conteúdo não assinado no bucket.</p> <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>, adicione o cabeçalho <code>x-amz-content-sha256</code> no cálculo da assinatura e defina o valor como a carga útil do hash.</p> <p>Você pode usar essa chave de condição na política de bucket para negar todo 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 usam 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-PAYLOAD</code></p>	String

Como as operações de API são autorizadas e autenticadas

A tabela a seguir lista as informações de autorização e autenticação para as operações de API da classe S3 Express One Zone. Para cada operação de API, a tabela mostra o nome da operação de API, a ação do IAM, o tipo de endpoint (regional ou zonal) e o mecanismo de autorização (IAM ou baseada em sessão). Essa tabela também indica onde o acesso entre contas é compatível. O acesso às ações no nível do bucket só pode ser concedido em políticas baseadas em identidade do IAM (usuário ou perfil), não em políticas de bucket.

API	Tipo de endpoint	Ação do IAM	Acesso entre contas
CreateBucket	Regional	s3express:CreateBucket	Não
DeleteBucket	Regional	s3express>DeleteBucket	Não
ListDirectoryBuckets	Regional	s3express:ListAllMyDirectoryBuckets	Não
PutBucketPolicy	Regional	s3express:PutBucketPolicy	Não
GetBucketPolicy	Regional	s3express:GetBucketPolicy	Não
DeleteBucketPolicy	Regional	s3express>DeleteBucketPolicy	Não
CreateSession	Zonal	s3express:CreateSession	Sim
CopyObject	Zonal	s3express:CreateSession	Sim
DeleteObject	Zonal	s3express:CreateSession	Sim
DeleteObjects	Zonal	s3express:CreateSession	Sim
HeadObject	Zonal	s3express:CreateSession	Sim
PutObject	Zonal	s3express:CreateSession	Sim

API	Tipo de endpoint	Ação do IAM	Acesso entre contas
GetObjectAttributes	Zonal	s3express:CreateSession	Sim
ListObjectsV2	Zonal	s3express:CreateSession	Sim
HeadBucket	Zonal	s3express:CreateSession	Sim
CreateMultipartUpload	Zonal	s3express:CreateSession	Sim
UploadPart	Zonal	s3express:CreateSession	Sim
UploadPartCopy	Zonal	s3express:CreateSession	Sim
CompleteMultipartUpload	Zonal	s3express:CreateSession	Sim
AbortMultipartUpload	Zonal	s3express:CreateSession	Sim
ListParts	Zonal	s3express:CreateSession	Sim
ListMultipartUploads	Zonal	s3express:CreateSession	Sim

Políticas baseadas em identidade do IAM para a classe S3 Express One Zone

Antes de criar buckets de diretório ou usar a classe de armazenamento Amazon S3 Express One Zone, você deve conceder as permissões necessárias para os perfis ou usuários do AWS Identity and Access Management (IAM). Esse exemplo de política permite acesso à operação de API `CreateSession` (para uso com as operações de API do endpoint zonal [nível de objeto]) e a todas as operações de API do endpoint regional (nível de bucket). Essa política permite que a operação de

API `CreateSession` seja usada com todos os buckets de diretório, mas as operações de API de endpoint regional só são permitidas para uso com o bucket de diretório especificado. Para usar esse exemplo de política, substitua os *user input placeholders* por suas próprias informações.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessRegionalEndpointAPIs",
      "Effect": "Allow",
      "Action": [
        "s3express:DeleteBucket",
        "s3express:DeleteBucketPolicy",
        "s3express:CreateBucket",
        "s3express:PutBucketPolicy",
        "s3express:GetBucketPolicy",
        "s3express:ListAllMyDirectoryBuckets"
      ],
      "Resource": "arn:aws:s3express:region:account_id:bucket/bucket-base-
name--azid--x-s3/*"
    },
    {
      "Sid": "AllowCreateSession",
      "Effect": "Allow",
      "Action": "s3express:CreateSession",
      "Resource": "*"
    }
  ]
}
```

Exemplos de políticas de bucket de diretório para a classe S3 Express One Zone

Esta seção fornece exemplos de políticas de bucket de diretório para usar com a classe de armazenamento Amazon S3 Express One Zone. Para usar essas políticas, substitua os *user input placeholders* por suas próprias informações.

O exemplo de política de bucket a seguir permite que o ID de Conta da AWS **111122223333** use a operação de API `CreateSession` com a sessão padrão `ReadWrite` para o bucket de diretório especificado. Essa política concede acesso às operações de API de endpoint zonal (nível de objeto).

Example : política de bucket para permitir chamadas de **CreateSession** com a sessão padrão **ReadWrite**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReadWriteAccess",
      "Effect": "Allow",
      "Resource": "arn:aws:s3express:us-west-2:account-id:bucket/bucket-base-
name--azid--x-s3",
      "Principal": {
        "AWS": [
          "111122223333"
        ]
      },
      "Action": [
        "s3express:CreateSession"
      ]
    }
  ]
}
```

Example : política de bucket para permitir chamadas de **CreateSession** com uma sessão **ReadOnly**

O exemplo de política de bucket a seguir permite que o ID de Conta da AWS **111122223333** use a operação de API `CreateSession`. Essa política usa a chave de condição `s3express:SessionMode` com o valor `ReadOnly` para definir uma sessão somente leitura.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReadOnlyAccess",
```



```

    "Effect": "Allow",
    "Principal": {
      "AWS": "111122223333"
    },
    "Action": "s3express:CreateSession",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "s3express:SessionMode": "ReadOnly"
      }
    }
  ]
}

```

Example : política de bucket para permitir o acesso entre contas para chamadas de **CreateSession**

O exemplo de política de bucket a seguir permite que o ID de Conta da AWS **111122223333** use a operação de API `CreateSession` para o bucket de diretório especificado que pertence ao ID de Conta da AWS **444455556666**.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CrossAccount",
      "Effect": "Allow",
      "Principal": {
        "AWS": "111122223333"
      },
      "Action": [
        "s3express:CreateSession"
      ],
      "Resource": "arn:aws:s3express:us-west-2:444455556666:bucket/bucket-base-name--azid--x-s3"
    }
  ]
}

```

CreateSession authorization

A classe Amazon S3 Express One Zone oferece suporte à autorização do AWS Identity and Access Management (AWS IAM) e à autorização baseada em sessão:

- Para usar as operações de API do endpoint regional (operações em nível de bucket ou ambiente de gerenciamento) com a classe S3 Express One Zone, use o modelo de autorização do IAM, que não envolve gerenciamento de sessões. As permissões são concedidas individualmente para as ações. Para ter mais informações, consulte [AWS Identity and Access Management \(IAM\) para a classe S3 Express One Zone](#).
- Para usar as operações de API do endpoint zonal (operações em nível de objeto ou plano de dados), use a operação de API CreateSession para criar e gerenciar sessões otimizadas para autorização de solicitações de dados com baixa latência. Para recuperar e usar um token de sessão, você deve permitir a ação `s3express:CreateSession` para o bucket de diretório em uma política baseada em identidade ou em uma política de bucket. Para ter mais informações, consulte [AWS Identity and Access Management \(IAM\) para a classe S3 Express One Zone](#). Se você acessar a classe S3 Express One Zone pelo console do Amazon S3, pela AWS Command Line Interface (AWS CLI) ou usando os AWS SDKs, a classe S3 Express One Zone criará uma sessão em seu nome.

Se você usar a API REST do Amazon S3, poderá usar a operação de API CreateSession para obter credenciais de segurança temporárias que incluem um ID de chave de acesso, uma chave de acesso secreta, um token de sessão e um prazo de expiração. As credenciais temporárias fornecem as mesmas permissões que as credenciais de segurança de longo prazo, como as credenciais de usuário do IAM, mas as credenciais de segurança temporárias precisam incluir um token de sessão.

Modo de sessão

O modo de sessão define o escopo da sessão. Na política de bucket, você pode especificar a chave de condição `s3express:SessionMode` para controlar quem pode criar uma sessão de ReadWrite ou ReadOnly. Para obter mais informações sobre sessões de ReadWrite ou ReadOnly, consulte o parâmetro `x-amz-create-session-mode` para [CreateSession](#) na Referência da API do Amazon S3. Para obter mais informações sobre a política de bucket que deve ser criada, consulte [Exemplos de políticas de bucket de diretório para a classe S3 Express One Zone](#).

Token de sessão

Ao fazer uma chamada usando credenciais de segurança temporárias, ela deve incluir um token de sessão. O token de sessão é retornado junto com as credenciais temporárias. Um token de sessão tem como escopo o bucket de diretório e é usado para verificar se as credenciais de segurança são válidas e não expiraram. Para proteger as sessões, as credenciais de segurança temporárias expiram após 5 minutos.

CopyObject e HeadBucket

As credenciais de segurança temporárias têm como escopo um bucket de diretório específico e são habilitadas automaticamente para todas as chamadas de API de operação zonal (nível de objeto) para determinado bucket de diretório. Ao contrário de outras operações de API de endpoint zonal, CopyObject e HeadBucket não usam a autenticação de CreateSession. Todas as solicitações CopyObject e HeadBucket devem ser autenticadas e assinadas usando credenciais do IAM. No entanto, CopyObject e HeadBucket ainda são autorizados por `s3express:CreateSession`, como outras operações de API de endpoint zonal.

Para obter mais informações, consulte [CreateSession](#) na Referência da API do Amazon Simple Storage Service.

Práticas recomendadas de segurança para a classe S3 Express One Zone

A classe Amazon S3 Express One Zone 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 para o seu ambiente, trate-as como recomendações úteis em vez de requisitos.

Configurações padrão das funcionalidades Bloqueio de Acesso Público e Propriedade de Objetos

Para usar a classe de armazenamento S3 Express One Zone, você deve usar um bucket de diretório do S3. Os buckets de diretório oferecem suporte às funcionalidades Bloqueio de Acesso Público do S3 e Propriedade de Objetos do S3. Esses recursos do S3 são usados para auditoria e gerenciamento de acesso aos buckets e objetos.

Por padrão, todas as configurações da funcionalidade Bloqueio de Acesso Público são habilitadas para buckets de diretório. Além disso, a funcionalidade Propriedade de Objetos está definida como Imposto pelo proprietário do bucket, o que significa que as listas de controle de acesso (ACLs) estão desabilitadas. Essas configurações não podem ser modificadas. Para obter mais informações

sobre esses recursos, consulte [the section called “Bloquear o acesso público”](#) e [the section called “Controlar a propriedade do objeto”](#).

Note

Você não pode conceder acesso a objetos armazenados em buckets de diretório. Você pode conceder acesso somente aos seus buckets de diretório. O modelo de autorização para a classe S3 Express One Zone é diferente do modelo de autorização para o Amazon S3. Para ter mais informações, consulte [CreateSession authorization](#).

Autenticação e autorização

Conforme você fizer solicitações para operações de API de endpoint zonal ou operações de API de endpoint regional, os mecanismos de autenticação e autorização da classe S3 Express One Zone serão diferentes. As operações de API zonais são operações em nível de objeto (plano de dados). As operações de API regionais são operações em nível de bucket (ambiente de gerenciamento).

Com a classe S3 Express One Zone, você autentica e autoriza solicitações para as operações de API de endpoint zonal por meio de um novo mecanismo baseado em sessão que é otimizado para fornecer a menor latência. Com a autenticação baseada em sessão, os SDKs da AWS usam a operação de API `CreateSession` para solicitar credenciais temporárias que fornecem acesso de baixa latência ao bucket de diretório. Essas credenciais temporárias têm como escopo um bucket de diretório específico e expiram depois de 5 minutos. Você pode usar essas credenciais temporárias para assinar chamadas de API zonais (nível de objeto). Para ter mais informações, consulte [CreateSession authorization](#).

Assinar solicitações com credenciais da classe S3 Express One Zone

Use credenciais da classe S3 Express One Zone para assinar solicitações de API de endpoint zonal (nível de objeto) com o AWS Signature Version 4, usando `s3express` como nome de serviço. Ao assinar as solicitações, use a chave secreta retornada por `CreateSession` e também forneça o token de sessão com `x-amzn-s3session-token` header. Para ter mais informações, consulte [CreateSession](#).

Os [AWS SDKs compatíveis](#) com a classe S3 Express One Zone gerenciam as credenciais e a assinatura em seu nome. Recomendamos usar os SDKs da AWS da classe S3 Express One Zone para atualizar as credenciais e assinar solicitações para você.

Assinar solicitações com credenciais do IAM

Todas as chamadas de API regionais (nível de bucket) devem ser autenticadas e assinadas por credenciais do AWS Identity and Access Management (IAM) em vez de credenciais temporárias de sessão. As credenciais do IAM consistem no ID da chave de acesso e na chave de acesso secreta para as identidades do IAM. Todas as solicitações CopyObject e HeadBucket também devem ser autenticadas e assinadas usando credenciais do IAM.

Para alcançar a menor latência para as chamadas de operação zonal (nível de objeto), recomendamos usar as credenciais da classe S3 Express One Zone obtidas da chamada de CreateSession para assinar as solicitações, exceto para as solicitações de CopyObject e HeadBucket.

Usar o AWS CloudTrail

O AWS CloudTrail fornece um registro das ações executadas por um usuário, perfil ou AWS service (Serviço da AWS) no Amazon S3. É possível usar as informações coletadas pelo CloudTrail para determinar o seguinte:

- A solicitação feita ao Amazon S3
- O endereço IP do qual a solicitação foi feita.
- Quem fez a solicitação.
- Quando a solicitação foi feita
- Detalhes adicionais sobre a solicitação

Quando você configura sua Conta da AWS, o CloudTrail é habilitado por padrão. As operações de API de endpoint regional a seguir (operações de API em nível de bucket ou ambiente de gerenciamento) são registradas em log no CloudTrail.

- CreateBucket
- DeleteBucket
- DeleteBucketPolicy
- PutBucketPolicy
- GetBucketPolicy
- ListDirectoryBuckets

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 [Creating a trail](#) no Guia de Usuário AWS CloudTrail.

Note

Para a classe S3 Express One Zone, o registro em log do CloudTrail de operações de API do endpoint zonal (nível de objeto ou plano de dados) (por exemplo, `PutObject` ou `GetObject`) não é compatível.

Implementar o monitoramento usando as ferramentas de monitoramento da AWS

O monitoramento é uma parte importante para manter a confiabilidade, segurança, disponibilidade e 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, é possível monitorar métricas do Amazon CloudWatch para o Amazon S3, particularmente as métricas de armazenamento `BucketSizeBytes` e `NumberOfObjects`.

Os objetos armazenados na classe de armazenamento S3 Express One Zone não serão refletidos nas métricas de armazenamento `BucketSizeBytes` e `NumberOfObjects` do Amazon S3. No entanto, as métricas de armazenamento `BucketSizeBytes` e `NumberOfObjects` são compatíveis com a classe S3 Express One Zone. Para ver as métricas de sua escolha, você pode diferenciar entre as classes de armazenamento do Amazon S3 e a classe de armazenamento S3 Express One Zone especificando uma dimensão `StorageType`. Para ter mais informações, consulte [Monitoramento de métricas com o Amazon CloudWatch](#).

Para ter mais informações, consulte [Monitoramento de métricas com o Amazon CloudWatch](#) e [Monitorar o Amazon S3](#).

Otimizar a performance da classe Amazon S3 Express One Zone

A classe Amazon S3 Express One Zone é uma classe de armazenamento de zona de disponibilidade (AZ) única e alta performance do S3 desenvolvida com o propósito específico de fornecer acesso consistente aos dados e com latência inferior a dez milissegundos para as aplicações mais sensíveis à latência. A classe S3 Express One Zone é a primeira classe de armazenamento do S3 que oferece a opção de manter o armazenamento de objetos de alto desempenho e os recursos de computação da AWS, como o Amazon Elastic Compute Cloud, o Amazon Elastic Kubernetes Service e o Amazon Elastic Container Service, em uma única zona de disponibilidade. A manutenção dos

recursos de armazenamento e computação na mesma localização otimiza a performance e os custos computacionais e acelera o processamento de dados.

A classe S3 Express One Zone oferece elasticidade de performance semelhante a outras classes de armazenamento do S3, mas com latências consistentes de solicitações de leitura e gravação do primeiro byte inferiores a dez milissegundos — até dez vezes mais rápidas do que a classe S3 Standard. A classe S3 Express One Zone foi projetada do zero para oferecer suporte a um throughput expansível até níveis agregados muito altos. A classe de armazenamento S3 Express One Zone usa uma arquitetura personalizada para otimizar a performance e oferecer latência de solicitações consistentemente baixa, armazenando dados em hardware de alta performance. O protocolo de objetos para a classe S3 Express One Zone foi aprimorado a fim de simplificar a autenticação e a sobrecarga de metadados.

Para aumentar ainda mais a velocidade de acesso e oferecer suporte a centenas de milhares de solicitações por segundo, a classe S3 Express One Zone armazena os dados em um novo tipo de bucket: um bucket de diretório do Amazon S3. Cada bucket de diretório do S3 é capaz de oferecer suporte a centenas de milhares de transações por segundo (TPS).

A combinação de hardware e software de alta performance com propósito específico que oferece velocidade de acesso aos dados inferior a dez milissegundos e buckets de diretório escaláveis para um grande número de transações por segundo faz da classe S3 Express One Zone a melhor classe de armazenamento do Amazon S3 para operações com alto consumo de solicitações ou aplicações de performance crítica.

Os tópicos a seguir descrevem as diretrizes de práticas recomendadas e os padrões de design para otimizar a performance de aplicações que usam a classe de armazenamento S3 Express One Zone.

Tópicos

- [Diretrizes de performance e padrões de design para a classe S3 Express One Zone](#)

Diretrizes de performance e padrões de design para a classe S3 Express One Zone

Ao criar aplicações que fazem upload e recuperam objetos da classe Amazon S3 Express One Zone, siga as nossas diretrizes de práticas recomendadas para otimizar a performance. Para usar a classe de armazenamento S3 Express One Zone, é necessário criar um bucket de diretório do S3. A classe de armazenamento S3 Express One Zone não é compatível com buckets de uso geral do S3.

Para consultar as diretrizes de desempenho para todas as outras classes de armazenamento do Amazon S3 e para buckets de uso geral do S3, consulte [Padrões de design de melhores práticas: otimizar a performance do Amazon S3](#).

Para obter a melhor performance de uma aplicação ao usar a classe de armazenamento S3 Express One Zone e os buckets de diretório, recomendamos as diretrizes e os padrões de design a seguir.

Tópicos

- [Implante o armazenamento da classe S3 Express One Zone na mesma localização dos recursos de computação da AWS](#).
- [Buckets de diretório](#)
- [Paralelização de solicitações de ajuste de escala horizontal de um bucket de diretórios](#)
- [Usar a autenticação baseada em sessão](#)
- [Práticas recomendadas adicionais de soma de verificação do S3](#)
- [Usar a versão mais recente dos AWS SDKs e das bibliotecas de tempo de execução comuns](#)
- [Solução de problemas de performance](#)

Implante o armazenamento da classe S3 Express One Zone na mesma localização dos recursos de computação da AWS.

Cada bucket de diretório é armazenado em uma única zona de disponibilidade selecionada ao criar o bucket. Você pode começar criando um bucket de diretório em uma zona de disponibilidade local para as workloads de computação ou os recursos. Depois, você pode iniciar imediatamente leituras e gravações de latência muito baixa. Os buckets de diretório são os primeiros buckets do S3 em que você pode escolher a zona de disponibilidade em uma Região da AWS a fim de reduzir a latência entre computação e armazenamento.

Se você acessar buckets de diretório entre zonas de disponibilidade, a latência aumentará. Para otimizar a performance, recomendamos que você acesse um bucket de diretório por instâncias do Amazon Elastic Container Service, do Amazon Elastic Kubernetes Service e do Amazon Elastic Compute Cloud que estejam localizadas na mesma zona de disponibilidade, sempre que possível.

Buckets de diretório

Cada bucket de diretório é capaz de oferecer suporte a centenas de milhares de transações por segundo (TPS). Ao contrário dos buckets de uso geral, os buckets de diretórios organizam as chaves hierarquicamente em diretórios em vez de prefixos. Um prefixo é uma string no início do nome da

chave do objeto. Você pode pensar nos prefixos como uma forma de organizar seus dados de forma semelhante aos diretórios. No entanto, os prefixos não são diretórios.

Os prefixos organizam os dados em um namespace simples dentro de buckets de uso geral, e não há limites para o número de prefixos em um intervalo de uso geral. Cada prefixo pode atingir pelo menos 3.500 PUT/POST/DELETE ou 5.500 GET/HEAD solicitações por segundo. Você também pode paralelizar solicitações entre vários prefixos para escalar a performance. No entanto, no caso de operações de leitura e gravação, esse ajuste de escala ocorre gradualmente e não é algo instantâneo. Embora os buckets de uso geral escalem para a nova taxa de solicitação mais alta, você poderá receber alguns erros do código de status HTTP 503 (serviço indisponível).

Com um namespace hierárquico, o delimitador na chave de objeto é importante. O único delimitador compatível é a barra (/). Os diretórios são determinados pelos limites de delimitador. Por exemplo, a chave de objeto `dir1/dir2/file1.txt` resulta na criação automática dos diretórios `dir1/` e `dir2/`, além da adição do objeto `file1.txt` ao diretório `/dir2` no caminho `dir1/dir2/file1.txt`.

Os diretórios criados quando os objetos são carregados em buckets de diretório não têm limites de TPS por prefixo e são pré-dimensionados automaticamente para reduzir a chance de erros HTTP 503 (serviço indisponível). Esse ajuste de escala automático permite que as aplicações emparelhem as solicitações de leitura e gravação dentro de um diretório e entre diretórios, conforme necessário.

Paralelização de solicitações de ajuste de escala horizontal de um bucket de diretórios

O melhor desempenho pode ser atingido com a emissão de várias solicitações simultâneas para buckets de diretório a fim de distribuir as solicitações em conexões separadas e maximizar a largura de banda acessível. A classe S3 Express One Zone não tem nenhum limite para o número de conexões feitas ao bucket de diretório. Diretórios individuais podem escalar a performance horizontal e automaticamente quando ocorre um grande número de gravações simultâneas no mesmo diretório.

Quando uma chave de objeto é criada e seu nome de chave inclui um diretório, o diretório é criado automaticamente para o objeto. Os uploads subsequentes de objetos para o mesmo diretório não exigirão que o diretório seja criado, o que reduzirá a latência dos uploads de objetos nos diretórios existentes.

Embora estruturas de diretórios superficiais e profundas sejam compatíveis para armazenar objetos em um bucket de diretório, os buckets de diretório têm a escala horizontal ajustada automaticamente, com menor latência em uploads simultâneos para o mesmo diretório ou para irmãos de diretórios paralelos.

Usar a autenticação baseada em sessão

A classe S3 Express One Zone e os buckets de diretório oferecem suporte a um novo mecanismo de autorização baseado em sessão para autenticar e autorizar solicitações para um bucket de diretório. Com a autenticação baseada em sessão, os AWS SDKs usam automaticamente a operação de API `CreateSession` para criar um token temporário de sessão que pode ser usado para autorização de baixa latência de solicitações de dados a um bucket de diretório.

Os AWS SDKs usam a operação de API `CreateSession` para solicitar credenciais temporárias, depois criam e atualizam tokens para você automaticamente a cada 5 minutos. Para aproveitar os benefícios de performance da classe de armazenamento S3 Express One Zone, recomendamos que você use os AWS SDKs para iniciar e gerenciar a solicitação de API `CreateSession`. Para obter mais informações sobre esse modelo baseado em sessão, consulte [CreateSession authorization](#).

Práticas recomendadas adicionais de soma de verificação do S3

A classe S3 Express One Zone oferece a opção de escolher o algoritmo de soma de verificação usado para validar os 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 cíclica (CRC): CRC32, CRC32C, SHA-1 e SHA-256. As somas de verificação baseadas em MD5 não são compatíveis com a classe de armazenamento S3 Express One Zone.

CRC32 é a soma de verificação padrão usada pelos AWS SDKs ao transmitir dados de ou para a classe S3 Express One Zone. Recomendamos usar CRC32 e CRC32C para obter a melhor performance com a classe de armazenamento S3 Express One Zone.

Usar a versão mais recente dos AWS SDKs e das bibliotecas de tempo de execução comuns

Vários dos AWS SDKs também fornecem as bibliotecas Common Runtime (CRT) da AWS para acelerar ainda mais a performance em clientes do S3. Esses SDKs incluem o AWS SDK for Java 2.x, o AWS SDK for C++ e o AWS SDK for Python (Boto3). O cliente do S3 baseado em CRT transfere objetos de e para a classe S3 Express One Zone com performance e confiabilidade aprimorados usando automaticamente a operação de API de multipart upload e buscas de intervalo de bytes para automatizar as conexões de ajuste de escala horizontal.

Para alcançar a melhor performance com a classe de armazenamento S3 Express One Zone, recomendamos usar a versão mais recente dos AWS SDKs que incluem as bibliotecas CRT ou usar a AWS Command Line Interface (AWS CLI).

Solução de problemas de performance

Solicitações de repetição para aplicações sensíveis à latência

A classe S3 Express One Zone foi desenvolvida com o propósito específico de oferecer níveis consistentes de alta performance sem ajustes adicionais. No entanto, definir valores agressivos de tempo limite e novas tentativas pode ajudar ainda mais a impulsionar níveis consistentes de latência e desempenho. Os AWS SDKs têm valores configuráveis de tempo limite e repetição que podem ser ajustados de acordo com as tolerâncias da aplicação específica.

Bibliotecas Common Runtime (CRT) da AWS e emparelhamento de tipos de instância do Amazon EC2

As aplicações que executam um grande número de operações de leitura e gravação provavelmente precisam de mais memória ou capacidade de computação do que as aplicações que não o fazem. Ao inicializar as instâncias do Amazon Elastic Compute Cloud (Amazon EC2) para uma workload que exige muita performance, escolha tipos de instância com a quantidade desses recursos que a aplicação precisa. O armazenamento de alta performance da classe S3 Express One Zone apresenta combinação ideal com tipos de instância maiores e mais novos, com quantidades maiores de memória do sistema e CPUs e GPUs mais poderosas que podem aproveitar o armazenamento de alta performance. Também recomendamos usar as versões mais recentes dos AWS SDKs habilitados para CRT, que podem acelerar ainda mais as solicitações de leitura e gravação em paralelo.

Usar a autenticação baseada em sessão nos AWS SDKs em vez das APIs REST HTTP

Com o Amazon S3, você também pode otimizar a performance ao usar solicitações da API REST HTTP seguindo as mesmas práticas recomendadas que fazem parte dos AWS SDKs. No entanto, com o mecanismo de autorização e autenticação baseado em sessão usado pela classe S3 Express One Zone, é altamente recomendável que você use os AWS SDKs para gerenciar `CreateSession` e o respectivo token de sessão gerenciada. Os AWS SDKs criam e atualizam tokens automaticamente em seu nome usando a operação de API `CreateSession`. O uso de `CreateSession` gera economia na latência de ida e volta por solicitação para que o AWS Identity and Access Management (IAM) autorize cada solicitação.

Desenvolver com a classe S3 Express One Zone

A classe Amazon S3 Express One Zone é a primeira classe de armazenamento do S3 em que é possível selecionar uma única zona de disponibilidade com a opção de manter o armazenamento de

objetos e os recursos de computação na mesma localização, o que fornece a maior velocidade de acesso possível. Com a classe de armazenamento S3 Express One Zone, use buckets de diretório do S3 para armazenar os dados. Cada bucket de diretório usa a classe de armazenamento S3 Express One Zone para armazenar objetos em uma única zona de disponibilidade que você pode selecionar ao criar o bucket.

Depois de criar o bucket de diretório, você pode começar a desfrutar imediatamente de leituras e gravações de latência muito baixa. Você pode se comunicar com o bucket de diretório usando uma conexão de endpoint em uma nuvem privada virtual (VPC) ou pode usar operações de API zonais e regionais para gerenciar objetos e buckets de diretório. Você também pode usar a classe de armazenamento S3 Express One Zone pelo console do Amazon S3, pela AWS Command Line Interface (AWS CLI), pelos AWS SDKs e pela API REST do Amazon S3.

A classe de armazenamento Amazon S3 Express One Zone foi projetada para oferecer disponibilidade de 99,95% em uma única zona de disponibilidade e tem o respaldo do [Acordo de Serviço do Amazon S3](#). Com a classe S3 Express One Zone, os dados são armazenados de forma redundante em vários dispositivos dentro de uma única zona de disponibilidade. A classe S3 Express One Zone foi projetada para lidar com falhas simultâneas de dispositivos, detectando e reparando rapidamente qualquer redundância perdida. Se o dispositivo existente encontrar uma falha, a classe S3 Express One Zone transferirá automaticamente as solicitações para novos dispositivos dentro da mesma zona de disponibilidade. Essa redundância ajuda a garantir o acesso ininterrupto aos dados dentro de uma zona de disponibilidade.

Tópicos

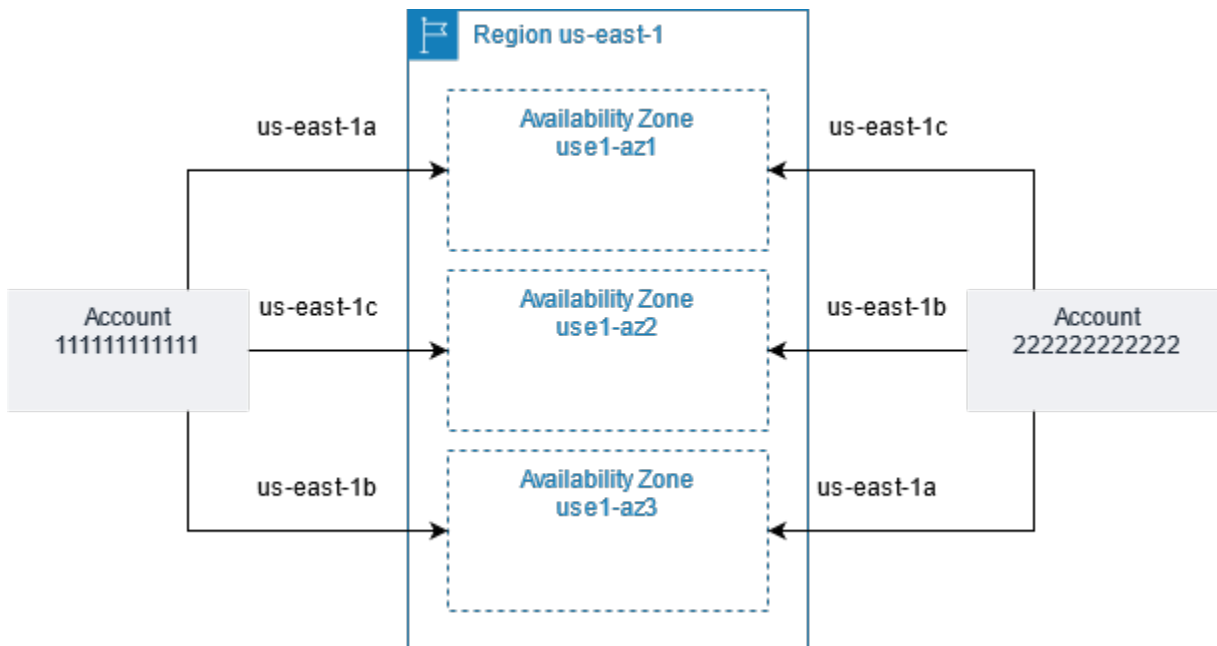
- [Zonas de disponibilidade e regiões da classe S3 Express One Zone](#)
- [Endpoints regionais e zonais](#)
- [Operações de API da classe S3 Express One Zone](#)

Zonas de disponibilidade e regiões da classe S3 Express One Zone

Uma zona de disponibilidade é um ou mais datacenters discretos com energia, redes e conectividade redundantes em uma Região da AWS. Para otimizar as recuperações de baixa latência, os objetos na classe de armazenamento Amazon S3 Express One Zone são armazenados de forma redundante em buckets de diretório do S3 em uma única zona de disponibilidade que seja local para sua workload computacional. Ao criar um bucket de diretório, escolha a zona de disponibilidade e a Região da AWS em que o bucket ficará localizado.

A AWS mapeia as zonas de disponibilidade físicas aleatoriamente em relação aos nomes das zonas de disponibilidade para cada Conta da AWS. Essa abordagem ajuda a distribuir recursos pelas zonas de disponibilidade em uma Região da AWS, em vez de os recursos provavelmente ficarem concentrados na primeira zona de disponibilidade de cada região. Como resultado, a zona de disponibilidade `us-east-1a` para sua Conta da AWS pode não representar a mesma localização física que `us-east-1a` para outra Conta da AWS. Para obter mais informações, consulte [Regiões e zonas de disponibilidade](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Para coordenar as zonas de disponibilidade entre contas, use o ID da AZ que é um identificador exclusivo e consistente para uma zona de disponibilidade. Por exemplo, `use1-az1` é um ID de AZ para a região `us-east-1` e tem o mesmo local físico em cada Conta da AWS. A ilustração a seguir mostra como os IDs da AZ são os mesmos para todas as contas, embora os nomes das zonas de disponibilidade possam ser mapeados de forma diferente para cada conta.



Com a classe S3 Express One Zone, os dados são armazenados de forma redundante em vários dispositivos dentro de uma única zona de disponibilidade. A classe S3 Express One Zone foi projetada para oferecer disponibilidade de 99,95% em uma única zona de disponibilidade e tem o respaldo do [Acordo de Serviço do Amazon S3](#). Para ter mais informações, consulte [Zona de disponibilidade única](#).

A classe S3 Express One Zone é compatível com as seguintes regiões e zonas de disponibilidade:

Zonas de disponibilidade e regiões compatíveis com a classe S3 Express One Zone

Nome da região	Código da região	ID da zona de disponibilidade
Leste dos EUA (Norte da Virgínia)	us-east-1	use1-az4
		use1-az5
		use1-az6
Oeste dos EUA (Oregon)	us-west-2	usw2-az1
		usw2-az3
		usw2-az4
Ásia-Pacífico (Tóquio)	ap-northeast-1	apne1-az1
		apne1-az4
Europa (Estocolmo)	eu-north-1	eun1-az1
		eun1-az2
		eun1-az3

Endpoints regionais e zonais

Para acessar endpoints regionais e zonais da classe Amazon S3 Express One Zone de uma nuvem privada virtual (VPC), você pode usar endpoints de VPC do gateway. Depois de criar um endpoint de gateway, é possível adicioná-lo como um destino na tabela de rotas para o tráfego destinado da VPC à classe S3 Express One Zone. Não há cobrança adicional pelo uso de endpoints do gateway. Para obter mais informações sobre como configurar endpoints de VPC do gateway, consulte [Redes para a classe S3 Express One Zone](#).

Ao trabalhar na classe S3 Express One Zone, as operações de API em nível de bucket (ambiente de gerenciamento) estão disponíveis por meio de um endpoint regional e são chamadas de

operações de API do endpoint regional. Exemplos de operações de API de endpoint regional são `CreateBucket` e `DeleteBucket`.

Depois de criar um bucket de diretório, você pode usar operações de API de endpoint zonal (nível de objeto ou plano de dados) para fazer upload e gerenciar objetos no bucket de diretório. As operações de API de endpoint zonal estão disponíveis por meio de um endpoint zonal. Exemplos de operações de API zonais são `PutObject` e `CopyObject`.

Operações de API da classe S3 Express One Zone

A classe de armazenamento Amazon S3 Express One Zone oferece suporte a operações de API de endpoint regional (nível de bucket ou ambiente de gerenciamento) e zonal (nível de objeto ou plano de dados). Para obter mais informações, consulte [Redes para a classe S3 Express One Zone](#) e [Endpoints e endpoints de VPC do gateway](#).

Operações de API de endpoint regional

As seguintes operações de API de endpoint regional são compatíveis com a classe S3 Express One Zone:

- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteBucketPolicy](#)
- [GetBucketPolicy](#)
- [ListDirectoryBuckets](#)
- [PutBucketPolicy](#)

Operações de API de endpoint zonal

As seguintes operações de API de endpoint zonal são compatíveis com a classe S3 Express One Zone:

- [CreateSession](#)
- [CopyObject](#)
- [DeleteObject](#)
- [DeleteObjects](#)
- [GetObject](#)

- [GetObjectAttributes](#)
- [HeadBucket](#)
- [HeadObject](#)
- [ListObjectsV2](#)
- [PutObject](#)
- [AbortMultipartUpload](#)
- [CompleteMultiPartUpload](#)
- [CreateMultipartUpload](#)
- [ListMultipartUploads](#)
- [ListParts](#)
- [UploadPart](#)
- [UploadPartCopy](#)

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](#).
- 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](#).

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](#). Para obter mais informações sobre como trabalhar com objetos, consulte [Visão geral de objetos Amazon S3](#).

Tópicos

- [Configurar políticas do IAM para uso de pontos de acesso](#)
- [Criar pontos de acesso](#)

- [Usar pontos de acesso](#)
- [Restrições e limitações de pontos de acesso](#)

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 do S3 a um bucket não altera o comportamento do bucket quando ele é acessado diretamente pelo nome do bucket ou pelo nome do 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 se aplicam somente a solicitações feitas por meio desse ponto de acesso.

Ao usar políticas de recursos do IAM, solucione 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 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 do 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](#).

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](#).
2. Adicione as mesmas permissões contidas na política de ponto de acesso à política do bucket subjacente. O Exemplo 1 de política de ponto de acesso demonstra como modificar a política de bucket subjacente para permitir o acesso necessário.

Example 1: Concessão da política de ponto de acesso

A política de ponto de acesso a seguir concede ao usuário do IAM *Jane* na conta *123456789012* permissões para objetos GET e PUT com o prefixo *Jane/* por meio do ponto de acesso *my-access-point* na conta *123456789012*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Jane"
      },
      "Action": ["s3:GetObject", "s3:PutObject"],
      "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point/object/Jane/*"
    }
  ]
}
```

Note

Para que a política de ponto de acesso conceda efetivamente acesso ao usuário *Jane*, o bucket subjacente também deve permitir o mesmo acesso a *Jane*. 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](#). Como alternativa, é possível adicionar a seguinte política ao

bucket subjacente para conceder as permissões necessárias à Jane. 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/Jane"
      },
      "Action": ["s3:GetObject", "s3:PutObject"],
      "Resource": "arn:aws:s3::DOC-EXAMPLE-BUCKET1/Jane/*"
    }
  ]
}
```

Example 2: Política de ponto de acesso com condição de etiqueta

A política de ponto de acesso a seguir concede ao usuário do IAM *Mateo* na conta *123456789012* permissões para objetos GET por meio do ponto de acesso *my-access-point* na conta *123456789012* que possui a chave de etiqueta *data* definida com um valor *finance*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Mateo"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3::us-west-2:123456789012:accesspoint/my-access-point/  
object/*",
      "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/data": "finance"
        }
      }
    }
  ]
}
```

Example 3: Política de ponto de acesso que permite a listagem de buckets

A política de ponto de acesso a seguir permite que o usuário do IAM Arnav na conta **123456789012** visualize os objetos contidos no bucket subjacente ao ponto de acesso **my-access-point** na conta **123456789012**.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Arnav"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point"
    }
  ]
}
```

Example 4: 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 de nuvem privada virtual (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",
      "Action": "s3:CreateAccessPoint",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "s3:AccessPointNetworkOrigin": "VPC"
        }
      }
    }
  ]
}
```

Example 5: Política de bucket para limitar operações do S3 às origens de rede de VPC

A política de bucket a seguir limita o acesso a todas as operações de objeto do S3 para o bucket *DOC-EXAMPLE-BUCKET* para pontos de acesso com uma origem de rede de VPC.

Important

Antes de usar uma instrução como essa do 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",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging",
        "s3:ListMultipartUploadParts",
        "s3:PutObject",
        "s3:PutObjectAcl",
        "s3:PutObjectLegalHold",
        "s3:PutObjectRetention",
        "s3:PutObjectTagging",
        "s3:PutObjectVersionAcl",
        "s3:PutObjectVersionTagging",
        "s3:RestoreObject"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
    "Condition": {
      "StringNotEquals": {
        "s3:AccessPointNetworkOrigin": "VPC"
      }
    }
  }
]
}

```

Chaves de condição

Os pontos de acesso do S3 têm chaves de condição que podem ser usadas em políticas do IAM para controlar o acesso aos recursos. As chaves de condição a seguir representam somente parte de uma política do IAM. Para obter exemplos de políticas completas, consulte [Exemplos de política de ponto de acesso](#), [the section called “Delegar controle de acesso a pontos de acesso”](#) e [the section called “Conceder permissões para pontos de acesso entre contas”](#).

s3:DataAccessPointArn

Este exemplo mostra uma string que você pode usar para estabelecer correspondência em 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 exemplo mostra um operador de string que você pode utilizar para estabelecer correspondência com o ID da conta do proprietário de um ponto de acesso. O exemplo a seguir estabelece correspondência com todos os pontos de acesso pertencentes à Conta da AWS *123456789012*.

```

"Condition" : {
  "StringEquals": {
    "s3:DataAccessPointAccount": "123456789012"
  }
}

```

```
}
}
```

s3:AccessPointNetworkOrigin

Este exemplo mostra um operador de string que você pode utilizar para estabelecer correspondência com 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"
  }
}
```

Consulte mais informações sobre como usar chaves de condição com o Amazon S3 em [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço.

Delegar controle de acesso a pontos de acesso

Você pode delegar o controle de acesso de um bucket aos pontos de acesso do bucket. O exemplo de política de bucket 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 6: Política de bucket que delega 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" }
      }
    }
  ]
}
```



```
    ]]
  }
```

Conceder permissões para pontos de acesso entre contas

Para criar um ponto de acesso para um bucket pertencente a outra conta, primeiro você deve criar o ponto de acesso especificando o nome do bucket e o ID do proprietário da conta. Depois, o proprietário do bucket deve atualizar a política do bucket para autorizar solicitações do ponto de acesso. Criar um ponto de acesso é semelhante à criação de um DNS CNAME, pois o ponto de acesso não fornece acesso ao conteúdo do bucket. Todo acesso ao bucket é controlado pela política de bucket. O exemplo de política de bucket a seguir permite solicitações GET e LIST no bucket de um ponto de acesso pertencente a uma Conta da AWS confiável.

Substitua *Bucket ARN* pelo ARN do bucket.

Example 7: Política de bucket que delega permissões a outra Conta da AWS

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect": "Allow",
      "Principal" : { "AWS": "*" },
      "Action" : ["s3:GetObject","s3:ListBucket"],
      "Resource" : [ "Bucket ARN", "Bucket ARN/*" ],
      "Condition": {
        "StringEquals" : { "s3:DataAccessPointAccount" : "Access point owner's
account ID" }
      }
    }
  ]
}
```

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 como solicitar um aumento, consulte [AWS Service Quotas](#) 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 pode começar nem terminar com hífen (-)
- Não pode 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 ter mais informações, consulte [Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3](#).

Para criar um ponto de acesso, consulte os tópicos a seguir.

Tópicos

- [Criar um ponto de acesso](#)
- [Criar pontos de acesso restritos a uma nuvem privada virtual](#)
- [Gerenciar o acesso público a pontos de acesso](#)

Criar um ponto de acesso

Um ponto de acesso está associado a exatamente um bucket do Amazon S3. Para usar um bucket em sua Conta da AWS, primeiro crie um bucket. Para obter mais informações sobre a criação de buckets, consulte [Criar, configurar e trabalhar com buckets do Amazon S3](#).

Você também pode criar um ponto de acesso entre contas associado a um bucket em outra Conta da AWS, desde que você saiba o nome do bucket e o ID da conta do proprietário do bucket. No entanto, a criação de pontos de acesso entre contas não concederá acesso aos dados no bucket até que você receba as permissões do proprietário do bucket. O proprietário do bucket deve conceder à conta do proprietário do ponto de acesso (sua conta) acesso ao bucket por meio da política do bucket. Para ter mais informações, consulte [Conceder permissões para pontos de acesso entre contas](#).

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 como solicitar um aumento, consulte [AWS Service Quotas](#) 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. Na barra de navegação na parte superior da página, escolha o nome da Região da AWS exibida no momento. Em seguida, escolha a região na qual você deseja criar um ponto de acesso.
3. No painel de navegação, escolha Access Points (Pontos de acesso).
4. Na página Access Points (Pontos de acesso), escolha Create access point (Criar ponto de acesso).
5. No campo Nome do ponto de acesso, insira o nome do ponto de acesso. Para obter mais informações sobre nomenclatura de pontos de acesso, consulte [Regras para nomear pontos de acesso do Amazon S3](#).
6. Em Nome do bucket, especifique o bucket do S3 que você deseja usar com o ponto de acesso.

Para usar um bucket em sua conta, selecione Escolha um bucket nesta conta e insira ou pesquise o nome do bucket.

Para usar um bucket em outra Conta da AWS, escolha Especificar um bucket em outra conta e insira o ID da Conta da AWS e o nome do bucket.


 Note

Se você estiver usando um bucket em outra Conta da AWS, o proprietário do bucket deverá atualizar a política do bucket para autorizar solicitações do ponto de acesso. Para ver um exemplo de política de bucket, consulte [Conceder permissões para pontos de acesso entre contas](#).

7. Escolha uma 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](#).

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 do bloqueio de acesso público são habilitadas por padrão para novos pontos de acesso. Recomendamos que você mantenha todas as configurações ativadas, a menos que saiba que tem uma necessidade específica de desativar qualquer uma delas.

 Note

Depois de criar um ponto de acesso, não será possível alterar as configurações de bloqueio de acesso público.

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](#).

9. (Opcional) Em Access point policy - optional (Política de ponto de acesso - opcional), especifique a política de ponto de acesso. Antes de salvar a política, solucione todos os avisos de segurança, os erros, os avisos gerais e as sugestões. Para obter mais informações sobre como especificar uma política de ponto de acesso, consulte [Exemplos de política de ponto de acesso](#).

10. Selecione Criar ponto de acesso.

Usando a AWS CLI

O exemplo de comando a seguir cria um ponto de acesso chamado *example-ap* para o bucket *DOC-EXAMPLE-BUCKET* na conta *111122223333*. Para criar o ponto de acesso, envie uma solicitação ao Amazon S3 que especifica o seguinte:

- O nome do ponto de acesso. Para obter informações sobre regras de nomeação, consulte [the section called “Regras para nomear pontos de acesso do Amazon S3”](#).
- O nome do bucket ao qual você deseja associar o ponto de acesso.
- O ID da Conta da AWS à qual o bucket pertence.

```
aws s3control create-access-point --name example-ap --account-id 111122223333 --  
bucket DOC-EXAMPLE-BUCKET
```

Ao criar um ponto de acesso usando um bucket em outra Conta da AWS, inclua o parâmetro `--bucket-account-id`. O exemplo de comando a seguir cria um ponto de acesso na Conta da AWS *111122223333*, usando o bucket *DOC-EXAMPLE-BUCKET2*, que está na Conta da AWS *444455556666*.

```
aws s3control create-access-point --name example-ap --account-id 111122223333 --  
bucket DOC-EXAMPLE-BUCKET --bucket-account-id 444455556666
```

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 endpoint da VPC especificado. Os VPC endpoints permitem que o tráfego flua 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](#).

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](#).

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
BlockPublicAcls=false,IgnorePublicAcls=false,BlockPublicPolicy=true,RestrictPublicBuckets=t
```

```
aws s3control get-access-point --name example-ap --account-id 123456789012
```

```
{
  "Name": "example-ap",
  "Bucket": "example-bucket",
  "NetworkOrigin": "Internet",
  "PublicAccessBlockConfiguration": {
    "BlockPublicAcls": false,
    "IgnorePublicAcls": 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 ARNs, consulte [Nomes de recurso da Amazon \(ARNs\)](#) no 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`.

Acessar um bucket por meio de pontos de acesso do S3

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.

Tópicos

- [Monitorar e registrar de pontos de acesso](#)
- [Como usar pontos de acesso do Amazon S3 com o console do Amazon S3](#)
- [Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3](#)
- [Usar pontos de acesso com operações compatíveis com o Amazon S3](#)

Se você tiver uma nuvem privada virtual (VPC), consulte [Gerenciar o acesso ao Amazon S3 com endpoints da VPC e pontos de acesso do S3](#).

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](#)
- [Logs da solicitação](#)

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](#).

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](#). 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](#)
- [Listar pontos de acesso para um bucket](#)
- [Visualizar detalhes de configuração de um ponto de acesso](#)
- [Usar um ponto de acesso](#)
- [Visualizar configurações de bloqueio de acesso público para um ponto de acesso](#)
- [Editar uma política de ponto de acesso](#)
- [Excluir um ponto de acesso](#)

Listar pontos de acesso para sua conta

Como listar todos os pontos de acesso criados em sua 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. Na barra de navegação na parte superior da página, escolha o nome da Região da AWS exibida no momento. Em seguida, escolha a região para a qual você deseja listar os pontos de acesso.
3. No painel de navegação, no lado esquerdo do console, escolha access points (pontos de acesso).
4. Na página Pontos de acesso, em Pontos de acesso, visualize os pontos de acesso em sua Região da AWS.
5. (Opcional) Pesquise pontos de acesso por nome inserindo um nome no campo de texto ao lado do menu suspenso de Região.
6. Escolha o nome do ponto de acesso que você deseja gerenciar ou usar.

Listar pontos de acesso para um bucket

Como listar todos os pontos de acesso em sua Conta da AWS para 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. Na barra de navegação na parte superior da página, escolha o nome da Região da AWS exibido atualmente e selecione a região para a qual você deseja listar os pontos de acesso.
3. No painel de navegação, no lado esquerdo do console, selecione Buckets.
4. Na página Buckets, selecione o nome do bucket cujos pontos de acesso você deseja listar.
5. Na página de detalhes do bucket, escolha a guia access points (pontos de acesso).
6. 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](#).
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](#).
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

you can execute on these objects, but not if you can't see that they exist in the 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](#).
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](#).
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](#).
2. Selecione o botão de opção ao lado do nome do ponto de acesso que você deseja excluir.

3. Escolha Delete.
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 de bucket do S3

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 operações de plano de dados do ponto de acesso. Para obter uma lista dessas operações, consulte [Compatibilidade com ponto de acesso com produtos da AWS](#).

Veja a seguir um exemplo de ARN e alias 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-hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias`

Para obter mais informações sobre ARNs, consulte [Nomes de recurso da Amazon \(ARNs\)](#) no 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](#).

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. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
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-aqfqprnstn7aefdfbarligizwgyfouse1a-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](#).

O exemplo a seguir da AWS CLI para o comando `get-object` usa o alias do ponto de acesso do bucket para retornar informações sobre o objeto especificado. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3api get-object --bucket my-access-point-aqfqprnstn7aefdfbarligizwgyfouse1a-
s3alias --key dir/my_data.rtf my_data.rtf
{
  "AcceptRanges": "bytes",
  "LastModified": "2020-01-08T22:16:28+00:00",
  "ContentLength": 910,
  "ETag": "\"00751974dc146b76404bb7290f8f51bb\"",
  "VersionId": "null",
  "ContentType": "text/rtf",
  "Metadata": {}
}
```

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](#).
- 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.
- No console do Amazon S3, não é possível usar aliases de ponto de acesso do S3 como origem ou destino para operações Mover.
- Não é possível usar alias em políticas do AWS Identity and Access Management (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 é compatível com alias de pontos 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](#)
- [Compatibilidade de ponto de acesso com operações do S3](#)
- [Solicitar um objeto por meio de um ponto de acesso](#)
- [Fazer upload de um objeto por meio de um alias de ponto de acesso](#)
- [Excluir um objeto por meio de um ponto de acesso](#)
- [Listar objetos por meio de um alias de ponto de acesso](#)
- [Adicionar um conjunto de tags a um objeto por meio de um ponto de acesso](#)
- [Conceder permissões de acesso por meio de um ponto de acesso usando uma ACL](#)

Compatibilidade com ponto de acesso com produtos da AWS

Os aliases de Pontos de Acesso Amazon S3 permitem que as aplicações que exijam um nome de bucket do S3 use um ponto de acesso com facilidade. É possível usar aliases de ponto de acesso

do S3 onde você usa nomes de bucket do S3 para acessar dados no S3. Para ter mais informações, consulte [Casos de uso e limitações de alias de ponto de acesso](#).

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](#)
- [GetObjectAttributes](#)
- [GetObjectLegalHold](#)
- [GetObjectRetention](#)
- [GetObjectTagging](#)
- [HeadBucket](#)
- [HeadObject](#)
- [ListMultipartUploads](#)
- [ListObjects](#)

- [ListObjectsV2](#)
- [ListObjectVersions](#)
- [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-hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias` pertencente ao ID de conta `123456789012` na região `us-west-2`.

AWS CLI

```
aws s3api put-object --bucket my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-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:

- 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, depois criar outro com o mesmo nome e associá-lo 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](#).
- 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 como solicitar um aumento, consulte [AWS Service Quotas](#) na Referência geral da AWS.
- Nas Regiões da AWS onde você tem mais de mil pontos de acesso, não é possível pesquisar um ponto de acesso pelo nome no console do Amazon S3.
- 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](#).
- No console do Amazon S3, não é possível usar aliases de ponto de acesso do S3 como origem ou destino para operações Mover.
- 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 [Acessar e listar um bucket do Amazon S3](#).
- As operações de API que controlam a funcionalidade do ponto de acesso (por exemplo, PutAccessPoint e GetAccessPointPolicy) não oferecem suporte a chamadas entre contas.

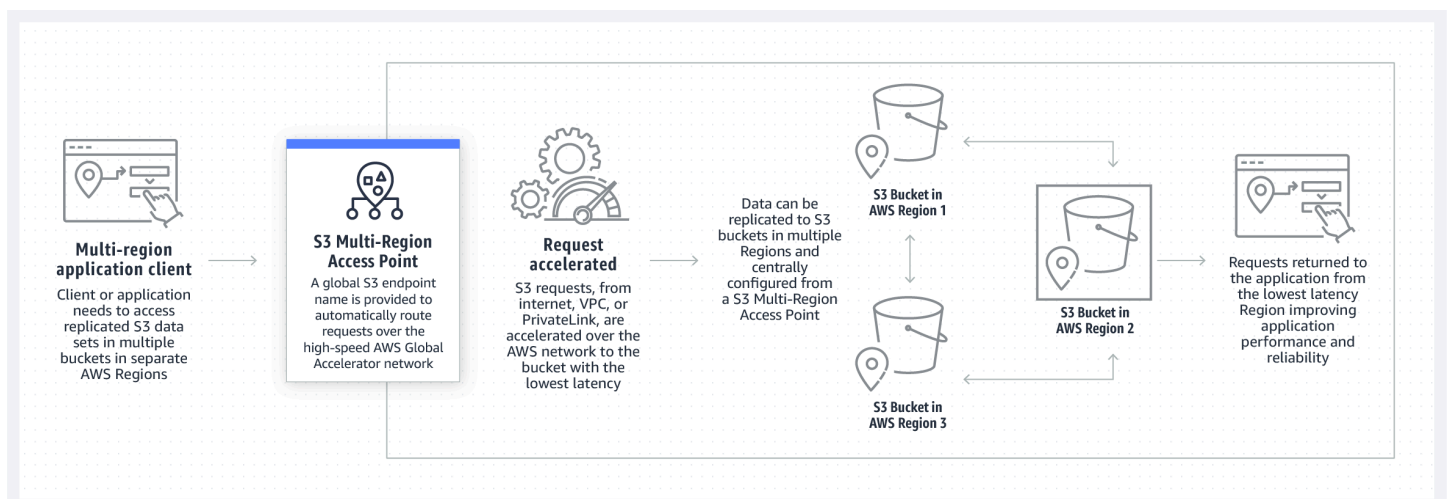
- É necessário usar o AWS Signature Version 4 ao fazer solicitações a um ponto de acesso usando as APIs REST. Para obter mais informações sobre como autenticar solicitações, consulte [Autenticação de solicitações \(AWS Signature Version 4\)](#) na Referência de APIs do Amazon Simple Storage Service.
- Os pontos de acesso permitem solicitações apenas por HTTPS. O Amazon S3 responderá automaticamente com um redirecionamento HTTP em todas as solicitações feitas por HTTP, para atualizar a solicitação para HTTPS.
- Os pontos de acesso não oferecem suporte ao acesso anônimo.
- Os pontos de acesso entre contas não concederão acesso aos dados no bucket até que você receba as permissões do proprietário do bucket. O proprietário do bucket sempre mantém o controle total sobre o acesso aos dados e deve atualizar a política do bucket para autorizar solicitações do ponto de acesso entre contas. Para ver um exemplo de política de bucket, consulte [Configurar políticas do IAM para uso de pontos de acesso](#).
- Quando você está visualizando um ponto de acesso entre contas no console Amazon S3, a coluna Acesso exibe Desconhecido. O console do Amazon S3 não pode determinar se o acesso público é concedido para o bucket e os objetos associados. A menos que você exija uma configuração pública para um caso de uso específico, recomendamos que você e o proprietário do bucket bloqueiem todo o acesso público ao ponto de acesso e ao bucket. Para ter mais informações, consulte [Bloquear o acesso público ao armazenamento do Amazon S3](#).

Pontos de acesso multirregionais no Amazon S3

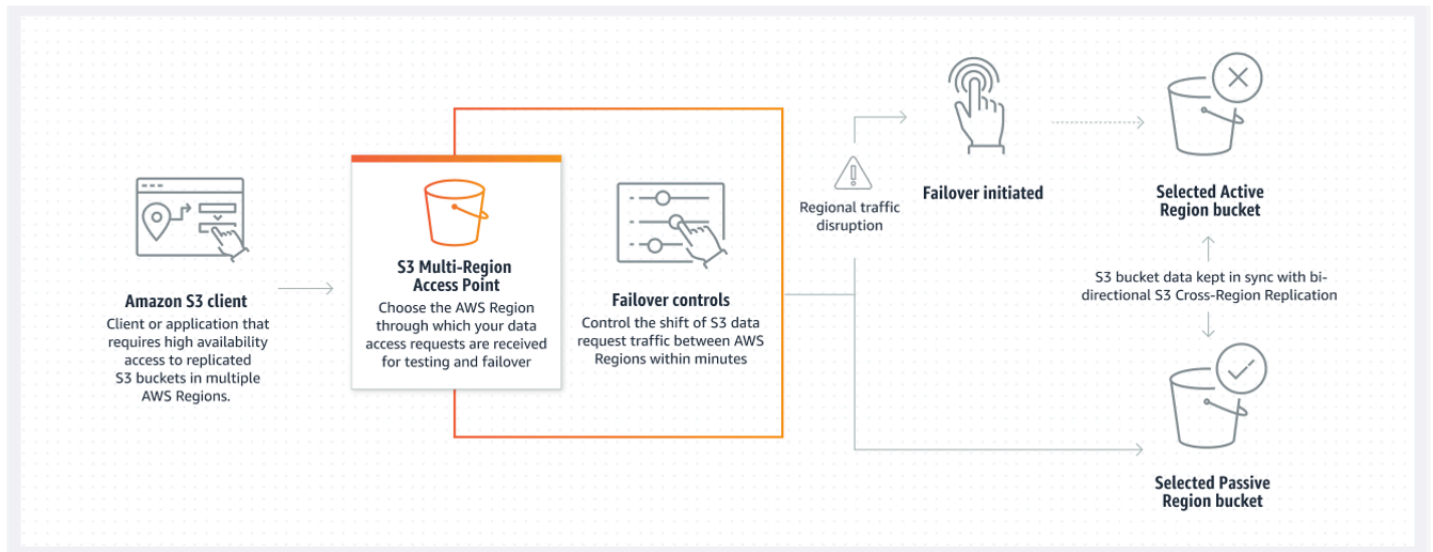
Os pontos de acesso multirregionais do Amazon S3 fornecem um endpoint global que as aplicações podem usar para atender a solicitações de buckets do S3 localizados em várias Regiões da AWS. Você pode usar pontos de acesso multirregionais para criar aplicações de várias regiões com a mesma arquitetura usada em uma única região, depois executar essas aplicações em qualquer lugar do mundo. Em vez de enviar solicitações pela Internet pública congestionada, os pontos de acesso multirregionais 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 ponto de acesso multirregional usam o [AWS Global Accelerator](#) para rotear automaticamente pela rede global da AWS para o bucket do S3 mais próximo com status de roteamento ativo.

Ao criar um ponto de acesso multirregional, você especifica um conjunto de Regiões da AWS onde deseja armazenar dados que serão fornecidos por meio desse ponto de acesso multirregional. 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 multirregional. O Amazon S3 atende automaticamente às solicitações para o conjunto de dados replicado da região mais próxima disponível. Os pontos de acesso multirregionais também são compatíveis com aplicações executadas em nuvens privadas virtuais (VPCs) da Amazon, incluindo aquelas que usam [AWS PrivateLink for Amazon S3](#).

A imagem a seguir é uma representação gráfica de um ponto de acesso multirregional do Amazon S3 em uma representação ativa-ativa. O gráfico mostra como as solicitações do Amazon S3 são roteadas automaticamente para buckets no Região da AWS ativo mais próximo.



A imagem a seguir é uma representação gráfica de um ponto de acesso multirregional do Amazon S3 em uma representação ativa-passiva. O gráfico mostra como você pode controlar o tráfego de acesso a dados do Amazon S3 para realizar failover entre Regiões da AWS ativas e passivas.



Para saber mais sobre como usar pontos de acesso multirregiões, consulte o [Tutorial: Introdução aos pontos de acesso multirregionais do Amazon S3](#).

Tópicos

- [Criação de pontos de acesso multirregionais](#)
- [Configurar um ponto de acesso multirregional para uso com AWS PrivateLink](#)
- [Como fazer solicitações por meio de um ponto de acesso multirregional](#)

Criação de pontos de acesso multirregionais

Para criar um ponto de acesso multirregional no Amazon S3, faça o seguinte:

- Especifique nome do ponto de acesso multirregional.
- Selecione um bucket em cada Região da AWS na qual deseja atender às solicitações para o ponto de acesso multirregional.
- Defina as configurações de bloqueio de acesso público do Amazon S3 para o ponto de acesso multirregional.

Você fornece todas 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.

Solucione 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 [Usar pontos de acesso multirregionais com operações de API compatíveis](#).

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 você pode usar para limitar o uso do ponto de acesso multirregional por recurso, usuário ou outras condições.

Note

Para que uma aplicação ou um usuário possa acessar um objeto por meio de um ponto de acesso multirregional, as duas políticas a seguir devem permitir a solicitação:

- A política de acesso para o ponto de acesso multirregional
- A política de acesso para o bucket subjacente que contém o objeto

Quando as duas políticas são diferentes, a política mais restritiva tem precedência.

Para simplificar o gerenciamento de permissões para pontos de acesso multirregionais, você pode delegar o controle de acesso do bucket para o ponto de acesso multirregional. Para

obter mais informações, consulte [the section called “Exemplos de políticas de pontos de acesso multirregionais”](#).

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 do 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. No entanto, você pode atualizar a política de pontos de acesso multirregionais para negar todas as permissões.

Tópicos

- [Regras para nomear pontos de acesso multirregionais do Amazon S3](#)
- [Regras para escolher buckets para pontos de acesso multirregionais do Amazon S3](#)
- [Criar um ponto de acesso multirregional do Amazon S3](#)
- [Bloqueio de acesso público de pontos de acesso multirregionais do Amazon S3](#)
- [Visualizar detalhes de configuração de pontos de acesso multirregionais do Amazon S3](#)
- [Excluir um ponto de acesso multirregional](#)

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 multirregionais](#). 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 utilizando 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 `.mr.ap`. 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 por meio de um ponto de acesso multirregional](#).

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. Os buckets que são compatíveis com o ponto de acesso multirregional podem estar na mesma Conta da AWS que tem o ponto de acesso multirregional, ou em outras Contas da AWS.

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.
- Se você adicionar um bucket de outra conta ao seu ponto de acesso multirregional, o proprietário do bucket também deverá atualizar a política do bucket para conceder permissões de acesso ao ponto de acesso multirregional. Caso contrário, o ponto de acesso multirregional não conseguirá recuperar dados desse bucket. Por ver exemplos de política que mostram como conceder esse acesso, consulte [Exemplos de políticas de pontos de acesso multirregionais](#).
- Nem todas as regiões suportam pontos de acesso multirregionais. Para visualizar a lista de regiões compatíveis, consulte [Restrições e limitações de pontos de acesso multirregionais](#).

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.

Important

Quando você faz uma solicitação para um ponto de acesso multirregional, o ponto de acesso multirregional não está ciente dos conteúdos dos dados dos buckets no ponto de acesso multirregional. Portanto, o bucket que recebe a solicitação pode não conter os dados solicitados. Para criar conjuntos de dados consistentes nos buckets do Amazon S3 associados a um ponto de acesso multirregional, recomendamos que você configure a Replicação do S3 Entre Regiões Diferentes (CRR). Para obter mais informações, consulte [Configurar a replicação para uso com pontos de acesso multirregionais](#).

Criar um ponto de acesso multirregional do Amazon S3

Os exemplos a seguir demonstram como criar um ponto de acesso multirregional usando o console do Amazon S3.

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 esquerdo, escolha Multi-Region Access Points (Pontos de acesso multirregionais).
3. Selecione Criar pontos de acesso multirregionais para começar a criar seu ponto de acesso multirregional.
4. Na página Ponto de acesso multirregional, forneça um nome para o ponto de acesso multirregional no campo Nome do ponto de acesso multirregional.
5. Selecione os buckets que serão associados a esse ponto de acesso multirregional. Você pode selecionar buckets que estão em sua conta ou buckets de outras contas.

Note

Você deve adicionar pelo menos um bucket de sua conta ou de outras. Além disso, esteja ciente de que os pontos de acesso multirregionais são compatíveis apenas com um bucket por Região da AWS. Portanto, você não pode adicionar dois buckets da mesma região. [Regiões da AWS que estão desativados por padrão](#) não são compatíveis.


- Para adicionar um bucket que esteja em sua conta, selecione Adicionar buckets. Uma lista de todos os buckets em sua conta é exibida. Você pode pesquisar seu bucket pelo nome ou classificar os nomes dos buckets em ordem alfabética.
- Para adicionar um bucket de outra conta, selecione Adicionar bucket de outras contas. É necessário saber o nome exato do bucket e o ID da Conta da AWS, pois não é possível pesquisar nem procurar buckets em outras contas.

Note

Você deve inserir um ID da Conta da AWS e um nome de bucket válidos. O bucket também deve estar em uma região compatível, ou você encontrará um erro ao tentar criar seu ponto de acesso multirregional. Para ver a lista de regiões que são


compatíveis com pontos de acesso multirregionais, consulte [Restrições e limitações de pontos de acesso multirregionais](#).

6. (Opcional) Se você precisar remover um bucket adicionado, selecione Remover.

 Note


Você não pode adicionar nem remover buckets desse ponto de acesso multirregional depois de terminar de criá-lo.

7. 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

Você não pode alterar as configurações de bloqueio de acesso público de um ponto de acesso multirregional após a criação dele. Portanto, se você quiser bloquear o acesso público, certifique-se de que suas aplicações funcionem corretamente sem acesso público antes de criar um ponto de acesso multirregional.

8. Selecione Create Multi-Region Access Point (Criar ponto de acesso multirregional).

 Important

Quando você adicionar um bucket de outra conta ao seu ponto de acesso multirregional, o proprietário do bucket também deverá atualizar a política do bucket para conceder permissões de acesso ao ponto de acesso multirregional. Caso contrário, o ponto de acesso multirregional não conseguirá recuperar dados desse bucket. Por ver exemplos de política que mostram como conceder esse acesso, consulte [Exemplos de políticas de pontos de acesso multirregionais](#).

Usar a AWS CLI

Você pode usar a AWS CLI para criar um ponto de acesso multirregional. Ao criar o ponto de acesso multirregional, você deve fornecer todos os buckets aos quais ele oferecerá suporte. Você não pode 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. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
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
```


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 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 o 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. Se o bloqueio de acesso público estiver ativado, o ponto de acesso multirregional não poderá aceitar solicitações baseadas na Internet.

 Important

Você não pode alterar as configurações de bloqueio de acesso público para um ponto de acesso multirregional após a criação do ponto de acesso.

Para obter mais informações sobre o bloqueio de acesso público do Amazon S3, consulte [Bloquear o acesso público ao armazenamento do Amazon S3](#).

Visualizar detalhes de configuração de pontos de acesso multirregionais do Amazon S3

O exemplo a seguir demonstra como visualizar detalhes de configuração de pontos de acesso multirregionais usando o console do Amazon S3.

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 esquerdo, escolha Multi-Region Access Points (Pontos de acesso multirregionais).
3. Selecione o nome do ponto de acesso multirregional cujos detalhes de configuração você deseja visualizar.
 - A guia Propriedades lista todos os buckets associados ao seu ponto de acesso multirregional, a data de criação, o nome do recurso da Amazon (ARN) e o alias. A coluna ID da Conta da AWS também lista todos os buckets pertencentes a contas externas associadas ao seu ponto de acesso multirregional.
 - A guia Permissões lista as configurações de Bloqueio de Acesso Público que são aplicadas aos buckets associados a esse ponto de acesso multirregional. Você também pode visualizar a política de pontos de acesso multirregionais para o seu ponto de acesso multirregional, se tiver criado uma. O alerta de informações na página Permissões também lista todos

os buckets (em sua conta e em outras) desse ponto de acesso multirregional que tem a configuração Acesso público está bloqueado ativada.

- A guia Replicação e failover fornece uma visualização do mapa dos buckets associados ao seu ponto de acesso multirregional e das regiões nas quais eles residem. Se houver buckets de outra conta dos quais você não tenha permissão para extrair dados, a região será marcada em vermelho no mapa Resumo da replicação, indicando que está em uma Região da AWS com erros ao obter o status de replicação.

Note

Para recuperar informações de status de replicação de um bucket em uma conta externa, o proprietário do bucket deve conceder a você a permissão `s3:GetBucketReplication` em sua política de bucket.

Essa guia também fornece as métricas de replicação, as regras de replicação e os status de failover para as regiões que são usadas com seu ponto de acesso multirregional.

Usar a AWS CLI

Você pode usar a AWS CLI para visualizar os detalhes de configuração de um ponto de acesso multirregional.

O exemplo de AWS CLI a seguir obtém sua configuração de ponto de acesso multirregional atual. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control get-multi-region-access-point --account-id 111122223333 --name DOC-EXAMPLE-BUCKET1
```

Excluir um ponto de acesso multirregional

O procedimento a seguir explica como excluir um ponto de acesso multirregional usando o console do Amazon S3.

A exclusão de um ponto de acesso multirregional não exclui os buckets associados ao ponto de acesso multirregional, apenas o próprio ponto de acesso multirregional.

Uso do console do S3

Como excluir 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 esquerdo, escolha Multi-Region Access Points (Pontos de acesso multirregionais).
3. Selecione o botão de opção ao lado do nome do seu ponto de acesso multirregional.
4. Escolha Delete (Excluir).
5. Na caixa de diálogo Excluir ponto de acesso multirregional, insira o nome do bucket da AWS que você deseja excluir.

Note

Insira um nome de bucket válido. Caso contrário, o botão Excluir será desativado.

6. Selecione Excluir para confirmar a exclusão de seu ponto de acesso multirregional.

Usar a AWS CLI

Você pode usar a AWS CLI para criar um ponto de acesso multirregional. A ação não exclui os buckets associados ao ponto de acesso multirregional, apenas o próprio ponto de acesso multirregional. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control delete-multi-region-access-point --account-id 123456789012 --details  
Name=example-multi-region-access-point-name
```

Configurar um ponto de acesso multirregional para uso com AWS PrivateLink

Você pode usar pontos de acesso multirregionais para direcionar o tráfego de solicitação do Amazon S3 entre Regiões da AWS. Cada endpoint global de ponto de acesso multirregional roteia o tráfego de solicitação de dados do Amazon S3 de várias fontes sem que você precise criar configurações de rede complexas com endpoints separados. Essas fontes de tráfego de solicitação de dados incluem:

- Tráfego originado de uma nuvem privada virtual (VPC)
- Tráfego de datacenters on-premises que percorre o AWS PrivateLink
- Tráfego da Internet pública

Se você estabelecer uma conexão do AWS PrivateLink com um ponto de acesso multirregional do S3, poderá rotear solicitações do S3 para a AWS, ou entre várias Regiões da AWS, por meio de uma conexão privada usando uma arquitetura e configuração de rede simples. Ao usar o AWS PrivateLink, você não precisa configurar uma conexão de emparelhamento da VPC.

Tópicos

- [Configurar um ponto de acesso multirregional para uso com AWS PrivateLink](#)
- [Remoção do acesso a um ponto de acesso multirregional de um endpoint da VPC](#)

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 associados podem pertencer à mesma conta ou a outra 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-EXAMPLE-BUCKET1` e `DOC-EXAMPLE-BUCKET2`, todos pertencentes à conta `123456789012` da AWS. Nesse caso, a política de endpoint da VPC a seguir permitiria que solicitações `GetObject` da VPC feitas para `mfzwi23gnjvgw.mrap` fossem atendidas 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-EXAMPLE-BUCKET1/*",
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET2/*",
      "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. O exemplo de política a seguir concederia acesso a qualquer solicitante que tentasse usar o ponto de acesso multirregional para as solicitações `GetObject`.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Open-read-MRAP-policy",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/*"
    }
  ]
}

```

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-EXAMPLE-BUCKET1",
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET2/*"]
}]
}
```

Para obter mais informações sobre como editar uma política de endpoint da VPC, 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 tiver um ponto de acesso multirregional e quiser remover o acesso dele de um endpoint de interface, você deverá fornecer uma nova política de acesso para o ponto de acesso multirregional que impede o acesso para solicitações que chegam por meio de endpoints da VPC. No entanto, se os buckets em seu ponto de acesso multirregional forem compatíveis com solicitações por meio de endpoints da VPC, eles continuarão a oferecer compatibilidade com 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, não a buckets subjacentes.

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.

Em vez de atualizar a política de acesso para o ponto de acesso multirregional, você pode atualizar as políticas de bucket para evitar solicitações por meio de endpoints da VPC. Nesse caso, os usuários ainda podem acessar o ponto de acesso multirregional por meio do endpoint da VPC. Porém, se a solicitação de ponto de acesso multirregional for encaminhada para um bucket no qual a política de bucket impeça o acesso, ela gerará uma mensagem de erro.

Como fazer solicitações por meio de um ponto de acesso multirregional

Assim como outros recursos, os pontos de acesso multirregionais do Amazon S3 têm nomes de recurso da Amazon (ARNs). Você pode usar esses ARNs para direcionar solicitações para pontos de acesso multirregionais usando a AWS Command Line Interface (AWS CLI), os AWS SDKs ou a API do Amazon S3. Você também pode usar esses ARNs para identificar pontos de acesso multirregionais nas políticas de controle de acesso. Um ARN de ponto de acesso multirregional não inclui nem divulga o nome do ponto de acesso multirregional. Para obter mais informações sobre ARNs, consulte [Nomes de recurso da Amazon \(ARNs\)](#) no Referência geral da AWS.

Note

O alias do ponto de acesso multirregional e o ARN não podem ser usados de forma intercambiável.

Os ARNs de ponto de acesso multirregional usam o seguinte formato:

```
arn:aws:s3::account-id:accesspoint/MultiRegionAccessPoint_alias
```

Veja a seguir alguns exemplos de ARNs de pontos de acesso multirregionais:

- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap` representa o ponto de acesso multirregional com o alias `mfzwi23gnjvgw.mrap`, de propriedade da Conta da AWS 123456789012.
- `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 Amazon S3 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 Amazon S3 na região `us-west-2` para a conta 123456789012, mas não corresponde a nenhum ponto de acesso multirregional.

Os ARNs para objetos que são acessados por meio de um ponto de acesso multirregional usam o seguinte formato:

```
arn:aws:s3::account_id:accesspoint/MultiRegionAccessPoint_alias//key
```


Assim como no caso de 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//01` representa o `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/*` 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//01/finance/*` representa todos os objetos que podem ser acessados sob o `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 de host do ponto de acesso multirregional. As solicitações podem ser direcionadas para esse nome de host pela Internet pública. Se você tiver configurado um ou mais gateways da Internet para o ponto de acesso multirregional, as solicitações também poderão ser direcionadas de uma nuvem privada virtual (VPC) para esse nome de host. 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](#).

Para fazer solicitações por meio de um ponto de acesso multirregional de uma VPC usando um endpoint da VPC, use AWS PrivateLink. Ao fazer solicitações para um ponto de acesso multirregional usando o AWS PrivateLink, você não pode usar diretamente um nome de SISTEMA DE NOMES DE DOMÍNIO (DNS) regional específico do endpoint que termine com `region.vpce.amazonaws.com`. Este nome de host não terá um certificado associado a ele, portanto ele não poderá ser usado diretamente. Você ainda pode usar o nome de SISTEMA DE NOMES DE DOMÍNIO (DNS) público do endpoint da VPC como um destino de CNAME ou ALIAS. Como alternativa, você pode ativar o SISTEMA DE NOMES DE DOMÍNIO (DNS) privado no endpoint e usar o nome de SISTEMA DE NOMES DE DOMÍNIO (DNS) `MultiRegionAccessPoint_alias.accesspoint.s3-global.amazonaws.com` do ponto de acesso multirregional padrão, conforme descrito nesta seção.

Quando você faz solicitações à API para operações de dados do Amazon S3 (por exemplo, `GetObject`) por meio de um ponto de acesso multirregional, o nome de host para a solicitação é o seguinte:

MultiRegionAccessPoint_alias.accesspoint.s3-global.amazonaws.com

Por exemplo, para fazer uma solicitação `GetObject` por meio do ponto de acesso multirregional com alias `mfzwi23gnjvgw.mrap`, faça uma solicitação para o nome de host `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`. A parte `s3-global` do nome de host 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. No entanto, é importante estar ciente das seguintes diferenças:

- Os ARNs de pontos de acesso multirregionais não incluem uma Região da AWS. Eles seguem o formato `arn:aws:s3::account-id:accesspoint/MultiRegionAccessPoint_alias`.
- Para solicitações feitas por meio de APIs (essas solicitações não exigem o uso de um ARN), os pontos de acesso multirregionais usam um esquema de endpoint diferente. O esquema é *MultiRegionAccessPoint_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 de pontos de acesso multirregionais não incluem o ID da Conta da AWS do proprietário.
 - Os nomes de host de pontos de acesso multirregionais 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 de pontos de acesso multirregionais precisam ser assinadas usando o Signature Version 4A (SigV4A). Quando você usa os AWS SDKs, o SDK converte automaticamente uma assinatura SigV4 em SigV4A. Portanto, verifique se seu [AWS SDK oferece compatibilidade](#) com o SigV4A como a implementação de assinatura usada para assinar as solicitações globais da Região da AWS. Para obter mais informações sobre o SigV4A, consulte [Assinar 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. A Aceleração de Transferência é configurada em cada bucket. Para obter mais informações sobre o Transfer Acceleration, consulte [Configurar transferências de arquivos rápidas e seguras usando o Amazon S3 Transfer Acceleration](#).

Os pontos de acesso multirregionais usam um mecanismo de transferência acelerada semelhante ao da Aceleração de Transferência para enviar objetos grandes pela rede da AWS. Por isso, você não precisa usar a Aceleração de Transferência ao enviar solicitações por meio de um ponto de acesso multirregional. Essa performance de transferência aprimorada é incorporada automaticamente ao ponto de acesso multirregional.

Tópicos

- [Permissões](#)
- [Restrições e limitações de pontos de acesso multirregionais](#)
- [Roteamento de solicitação de pontos de acesso multirregionais](#)
- [Controles de failover para pontos de acesso multirregionais do Amazon S3](#)
- [Configurar a replicação para uso com pontos de acesso multirregionais](#)
- [Usar pontos de acesso multirregionais com operações de API compatíveis](#)
- [Monitoramento e registro de solicitações feitas por meio de um ponto de acesso multirregional para recursos subjacentes](#)

Permissões

Os pontos de acesso multirregionais do Amazon S3 podem simplificar o acesso a dados para buckets do Amazon S3 em várias Regiões da AWS. Os pontos de acesso multirregionais são endpoints globais que você pode usar para realizar operações de objeto de acesso a dados no Amazon S3, como `GetObject` e `PutObject`. Cada ponto de acesso multirregional pode ter permissões e controles de rede distintos para todas as solicitações feitas por meio do endpoint global.

Cada ponto de acesso multirregional também pode impor uma política de acesso personalizada que funciona em conjunto com a política de bucket anexada ao bucket subjacente. Para que uma solicitação seja bem-sucedida, todos os itens a seguir devem permitir a operação:

- A política de ponto de acesso multirregional

- A política subjacente do AWS Identity and Access Management (IAM)
- A política de bucket subjacente (para onde a solicitação é encaminhada)

Você pode configurar qualquer política de ponto de acesso multirregional para aceitar solicitações somente de usuários ou grupos específicos do IAM. Para obter um exemplo de como fazer isso, consulte o Exemplo 2 em [the section called “Exemplos de políticas de pontos de acesso multirregionais”](#). Você pode configurar a política de ponto de acesso multirregional para aceitar solicitações somente de uma nuvem privada virtual (VPC) para restringir o acesso a dados do Amazon S3 a uma rede privada.

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` do ponto de acesso multirregional e de cada bucket subjacente ao ponto de acesso multirregional. O `AppDataReader` não será capaz de recuperar dados de todo bucket que não conceder essa permissão.

Important

Delegar o controle de acesso de um bucket a uma política de ponto de acesso multirregional não altera o comportamento do bucket quando o bucket é acessado diretamente por meio de seu nome ou do nome do recurso da Amazon (ARN). Todas as operações feitas diretamente no bucket continuarão a funcionar como antes. As restrições que você incluir em uma política de ponto de acesso multirregional se aplicam somente às solicitações feitas por meio desse ponto de acesso multirregional.

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.

Note

Todas as configurações de bloqueio de acesso público ativadas em **Bloquear acesso público** para esta conta (em sua própria conta) ou **Bloquear configurações públicas para buckets**

externos ainda se aplicam, mesmo que as configurações independentes de bloqueio de acesso público para seu ponto de acesso multirregional estejam desativadas.

Para qualquer solicitação feita por meio de um ponto de acesso multirregional, o Amazon S3 avaliará as configurações de Bloqueio de Acesso Público para:

- O ponto de acesso multirregional
- Os buckets subjacentes (incluindo buckets externos)
- A conta que detém o ponto de acesso multirregional
- A conta que detém os buckets subjacentes (incluindo contas externas)

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](#).

Important

Por padrão, todas as configurações de Bloqueio de Acesso Público são habilitadas para pontos de acesso multirregionais. Você deve desativar explicitamente todas as configurações que não deseja aplicar a um ponto de acesso multirregional.

Você não pode alterar as configurações de bloqueio de acesso público de um ponto de acesso multirregional após a criação dele.

Visualizar configurações de Bloqueio de Acesso Público para um ponto de acesso multirregional

Como visualizar as configurações de Bloqueio de Acesso Público para 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 esquerdo, escolha Multi-Region Access Points (Pontos de acesso multirregionais).
3. Escolha o nome do ponto de acesso multirregional que você deseja revisar.

4. Escolha a guia Permissions (Permissões).
5. Em Block Public Access settings for this Multi-Region Access Point (Configurações do Bloqueio de Acesso Público para este ponto de acesso multirregional), revise as configurações de Bloqueio de Acesso Público que você deseja aplicar ao ponto de acesso multirregional.

 Note

Não é possível editar as configurações de Bloqueio de Acesso Público após a criação do ponto de acesso multirregional. Portanto, se você quiser bloquear o acesso público, certifique-se de que suas aplicações funcionem corretamente sem acesso público antes de criar um ponto de acesso multirregional.

Como usar uma política de ponto de acesso multirregional

O exemplo de política de ponto de acesso multirregional a seguir concede a um usuário do IAM acesso para listar e baixar arquivos do ponto de acesso multirregional. Para usar esse exemplo de política, substitua os *user input placeholders* por suas próprias informações.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/JohnDoe"
      },
      "Action": [
        "s3:ListBucket",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias",
        "arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias/object/"
      ]
    }
  ]
}
```

Para associar sua política de ponto de acesso multirregional ao ponto de acesso multirregional especificado usando a AWS Command Line Interface (AWS CLI), use o comando `put-multi-region-access-point-policy` a seguir. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações. Cada ponto de acesso multirregional pode ter somente uma política, portanto uma solicitação feita para a ação `put-multi-region-access-point-policy` substitui qualquer política existente associada ao ponto de acesso multirregional especificado.

AWS CLI

```
aws s3control put-multi-region-access-point-policy
--account-id 111122223333
--details { "Name": "DOC-EXAMPLE-BUCKET-MultiRegionAccessPoint",
  "Policy": "{ \"Version\": \"2012-10-17\", \"Statement\": { \"Effect\":
  \"Allow\", \"Principal\": { \"AWS\": \"arn:aws:iam::111122223333:root
  \", \"Action\": [\"s3:ListBucket\", \"s3:GetObject\"], \"Resource\":
  [ \"arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias\",
  \"arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias/object/*
  \"] ] } }" }
```

Para consultar os resultados da operação anterior, use o seguinte comando:

AWS CLI

```
aws s3control describe-multi-region-access-point-operation
--account-id 111122223333
--request-token-arn requestArn
```

Para recuperar sua política de ponto de acesso multirregional, use o seguinte comando:

AWS CLI

```
aws s3control get-multi-region-access-point-policy
--account-id 111122223333
--name=DOC-EXAMPLE-BUCKET-MultiRegionAccessPoint
```

Editar a política de ponto de acesso multirregional

A política de ponto de acesso multirregional (escrita em JSON) fornece acesso de armazenamento aos buckets do Amazon S3 que são usados com esse ponto de acesso multirregional. Você pode permitir ou negar que entidades principais específicas realizem várias ações em seu ponto de acesso multirregional. Quando uma solicitação é roteada para um bucket por meio do ponto de acesso multirregional, as políticas de acesso tanto do ponto de acesso multirregional como do bucket se aplicam. A política de acesso mais restritiva sempre tem precedência.

Note

Se um bucket contiver objetos pertencentes a outras contas, a política de ponto de acesso multirregional não se aplicará aos objetos que pertencem a outras Contas da AWS.

Depois que você aplicar uma política de ponto de acesso multirregional, esta política não poderá ser excluída. Você poderá editar a política ou criar uma política que substitua a existente.

Como editar a política de 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 esquerdo, escolha Multi-Region Access Points (Pontos de acesso multirregionais).
3. Escolha o nome do ponto de acesso multirregional para o qual deseja editar a política.
4. Escolha a guia Permissions (Permissões).
5. Role para baixo até a seção Multi-Region Access Point policy (Política de ponto de acesso multirregional). Escolha Edit (Editar) para atualizar a política (em JSON).
6. A página Edit Multi-Region Access Point policy (Editar política de ponto de acesso multirregional) é exibida. Você pode inserir a política diretamente no campo de texto ou escolher Add statement (Adicionar instrução) para selecionar elementos de política em uma lista suspensa.

Note

O console exibe automaticamente o nome do recurso da Amazon (ARN) do ponto de acesso multirregional, que você pode usar na política. Para conferir exemplos de

políticas de pontos de acesso multirregionais, consulte [the section called “Exemplos de políticas de pontos de acesso multirregionais”](#).

Exemplos de políticas de pontos de acesso multirregionais

Os pontos de acesso multirregionais do Amazon S3 são compatíveis com políticas de recursos do AWS Identity and Access Management (IAM). Você pode usar essas políticas para controlar o uso do ponto de acesso multirregional por recurso, usuário ou outras condições. Para que uma aplicação ou um usuário possa acessar objetos por meio de um ponto de acesso multirregional, tanto o ponto de acesso multirregional quanto o bucket subjacente devem permitir o mesmo acesso.

Para permitir o mesmo acesso ao ponto de acesso multirregional e ao bucket subjacente, faça o seguinte:

- (Recomendado) Para simplificar os controles de acesso ao usar um ponto de acesso multirregional do Amazon S3, delegue o controle de acesso do bucket do Amazon S3 ao ponto de acesso multirregional. Para obter um exemplo de como fazer isso, consulte o Exemplo 1 desta seção.
- Adicione as mesmas permissões contidas na política de ponto de acesso multirregional à política do bucket subjacente.

Important

Delegar o controle de acesso de um bucket a uma política de ponto de acesso multirregional não altera o comportamento do bucket quando o bucket é acessado diretamente por meio de seu nome ou do nome do recurso da Amazon (ARN). Todas as operações feitas diretamente no bucket continuarão a funcionar como antes. As restrições que você incluir em uma política de ponto de acesso multirregional se aplicam somente às solicitações feitas por meio desse ponto de acesso multirregional.

Example 1: Delegar acesso a pontos de acesso multirregionais específicos em sua política de bucket (para a mesma conta ou por outras contas)

O exemplo de política de bucket a seguir concede acesso total a um ponto de acesso multirregional específico. Isso significa que todo acesso a esse bucket é controlado pelas políticas anexadas ao ponto de acesso multirregional. Recomendamos configurar seus buckets dessa maneira para todos

os casos de uso que não exigem acesso direto ao bucket. Você pode usar essa estrutura de política de bucket para pontos de acesso multirregionais na mesma conta ou em outra.

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect": "Allow",
      "Principal" : { "AWS": "*" },
      "Action" : "*",
      "Resource" : [ "Bucket ARN", "Bucket ARN/*" ],
      "Condition": {
        "StringEquals" : { "s3:DataAccessPointArn" : "MultiRegionAccessPoint_ARN" }
      }
    }
  ]
}
```

Note

Se houver vários pontos de acesso multirregionais aos quais você está concedendo acesso, liste cada ponto de acesso multirregional.

Example 2: Conceder acesso a um ponto de acesso multirregional em sua política de ponto de acesso multirregional

A política de ponto de acesso multirregional a seguir permite que a conta **123456789012** liste e leia os objetos contidos no ponto de acesso multirregional definido pelo **MultiRegionAccessPoint_ARN**.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/JohnDoe"
      },
      "Action": [
        "s3:ListBucket",
        "s3:GetObject"
      ]
    }
  ]
}
```

```
    ],
    "Resource": [
      "MultiRegionAccessPoint_ARN",
      "MultiRegionAccessPoint_ARN/object/*"
    ]
  }
]
```

Example 3: Política de ponto de acesso multirregional que permite a listagem de buckets

A política de ponto de acesso multirregional a seguir permite que a conta *123456789012* liste os objetos contidos no ponto de acesso multirregional definido pelo *MultiRegionAccessPoint_ARN*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/JohnDoe"
      },
      "Action": "s3:ListBucket",
      "Resource": "MultiRegionAccessPoint_ARN"
    }
  ]
}
```

Restrições e limitações de pontos de acesso multirregionais

Os pontos de acesso multirregionais do Amazon S3 têm as seguintes restrições e limitações:

- Nomes de pontos de acesso multirregionais:
 - Devem ser exclusivos em uma única conta da AWS
 - Devem começar com um número ou uma letra minúscula
 - Devem conter entre 3 e 50 caracteres
 - Não pode começar nem terminar com hífen (-)
 - Não pode conter sublinhados (_), letras maiúsculas ou pontos (.)
 - Não podem ser editados depois que são criados

- Os aliases de pontos de acesso multirregionais são gerados pelo Amazon S3 e não podem ser editados nem reutilizados.
- Não é possível acessar dados por meio de um ponto de acesso multirregional usando endpoints de gateway. No entanto, é possível acessar dados por meio de um ponto de acesso multirregional usando endpoints de interface. Para usar AWS PrivateLink, você deve criar endpoints da VPC. Para ter mais informações, consulte [Configurar um ponto de acesso multirregional para uso com AWS PrivateLink](#).
- Para usar pontos de acesso multirregionais com o Amazon CloudFront, você deve configurar o ponto de acesso multirregional como um tipo de distribuição Custom Origin. Para obter mais informações sobre vários tipos de origem, consulte [Usar várias origens com distribuições do CloudFront](#). Para obter mais informações sobre como usar pontos de acesso multirregionais com o Amazon CloudFront, consulte [Criar uma aplicação ativa-ativa baseada em proximidade em várias regiões](#) no Blog de armazenamento da AWS.
- Requisitos mínimos para pontos de acesso multirregionais:
 - Transport Layer Security (TLS) v1.2
 - Signature versão 4 (SigV4A)

Os pontos de acesso multirregionais são compatíveis com o Signature versão 4A. Esta versão do SigV4 permite que as solicitações sejam assinadas para várias Regiões da AWS. Esse recurso é útil em operações de API que podem resultar em acesso a dados de uma das várias regiões. Ao usar um AWS SDK, forneça suas credenciais, e as solicitações para pontos de acesso multirregionais usarão o Signature Version 4A sem configuração adicional. Verifique a [compatibilidade de AWS SDKs](#) com o algoritmo SigV4A. Para obter mais informações sobre o SigV4A, consulte [Assinar solicitações de API da AWS](#) na Referência geral da AWS.

Note

Para usar o SigV4A com credenciais de segurança temporárias — por exemplo, ao usar perfis do AWS Identity and Access Management (IAM), solicite as credenciais temporárias de um endpoint regional no AWS Security Token Service (AWS STS), em vez de um endpoint global. Se você usar o endpoint global para o AWS STS (`sts.amazonaws.com`), o AWS STS vai gerar credenciais temporárias de um endpoint global, o que não é compatível com o SigV4A. Como resultado, você receberá um erro. Para resolver esse problema, use qualquer um dos [Endpoints regionais para AWS STS](#) listados.

- Os pontos de acesso multirregionais não são compatíveis com solicitações anônimas.
- Limitações do ponto de acesso multirregional:
 - O IPv6 não é compatível.
 - Não há suporte para buckets do Amazon S3 on Outposts.
 - Os pontos de acesso multirregionais permitem operações de cópia que os utilizem somente como destino ao usar o ARN do ponto de acesso multirregional.
 - O recurso Operações em Lote do S3 não é compatível.
- Alguns AWS SDKs não são compatíveis. Para confirmar quais AWS SDKs são compatíveis com pontos de acesso multirregionais, consulte [Compatibilidade com AWS SDKs](#).
- As cotas de serviço para pontos de acesso multirregionais são as seguintes:
 - Há no máximo 100 pontos de acesso multirregionais por conta.
 - Há um limite de 17 regiões para um único ponto de acesso multirregional.
- Depois de criar um ponto de acesso multirregional, você não pode adicionar, modificar nem 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. Se um bucket entre contas em seu ponto de acesso multirregional for excluído, a única maneira de reconectar esse bucket é recriar o bucket usando o mesmo nome e região nessa conta.
- Os buckets subjacentes (na mesma conta) que são usados em um ponto de acesso multirregional só podem ser excluídos após a exclusão de um ponto de acesso multirregional.
- Todas as solicitações do ambiente de gerenciamento para criar ou manter pontos de acesso multirregionais devem ser encaminhadas para a região US West (Oregon). Com relação a solicitações de plano de dados de pontos de acesso multirregionais, as regiões não precisam ser especificadas.
- Para o ambiente de gerenciamento de failover do ponto de acesso multirregional, as solicitações devem ser encaminhadas para uma das cinco regiões compatíveis:
 - US East (N. Virginia)
 - US West (Oregon)
 - Asia Pacific (Sydney)
 - Asia Pacific (Tokyo)
 - Europe (Ireland)
- Seu ponto de acesso multirregional só é compatível com buckets das seguintes Regiões da AWS:
 - US East (N. Virginia)

- US East (Ohio)
- US West (N. California)
- US West (Oregon)
- Asia Pacific (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seoul)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Asia Pacific (Tokyo)
- Canada (Central)
- Europe (Frankfurt)
- Europe (Ireland)
- Europe (London)
- Europe (Paris)
- Europe (Stockholm)
- South America (São Paulo)

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 qual dos buckets associados ao ponto de acesso multirregional está mais próximo. 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 mais próximo, o Amazon S3 processará a solicitação como se você a tivesse feito diretamente para esse bucket. Pontos de acesso multirregionais não estão cientes do conteúdo de dados de um bucket do Amazon S3. Portanto, o bucket que recebe a solicitação pode não conter os dados solicitados. Para criar conjuntos de dados consistentes nos buckets do Amazon S3 associados a um ponto de acesso multirregional, você pode configurar a Replicação do S3 Entre Regiões Diferentes (CRR). Em seguida, qualquer bucket pode cumprir a solicitação com êxito.

O Amazon S3 direciona solicitações de pontos de acesso multirregionais de acordo com as seguintes regras:

- O Amazon S3 otimiza as solicitações para que sejam atendidas de acordo com a proximidade. Ele examina os buckets compatíveis com o ponto de acesso multirregional e repassa a solicitação para o bucket mais próximo.
- 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 mesmo se houver um objeto em um bucket no ponto de acesso multirregional, sua solicitação será roteada para um bucket que não contém o objeto. Essa situação gerará uma mensagem de erro 404 retornada ao cliente.

Para evitar erros 404, recomendamos que você configure a Replicação do S3 Entre Regiões Diferentes (CRR) para seus buckets. A replicação 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 [Configurar a replicação para uso com pontos de acesso multirregionais](#).

Para garantir que suas solicitações sejam atendidas usando os objetos específicos que você deseja, recomendamos que você ative o versionamento do bucket e inclua IDs de versão em suas solicitações. Essa abordagem ajuda a garantir que você tenha a versão correta do objeto que você está procurando. Buckets habilitados para versionamento também podem ajudar a recuperar objetos de uma substituição acidental. Para ter mais informações, consulte [Usar o versionamento em buckets do S3](#).

- Se a solicitação for para criar um recurso (por exemplo, `PutObject` ou `CreateMultipartUpload`), o Amazon S3 atenderá à solicitação usando o bucket mais próximo. Por exemplo, pense em uma empresa de vídeo que queira oferecer compatibilidade com uploads de vídeo de qualquer lugar do mundo. Quando um usuário faz uma solicitação PUT para o ponto de acesso multirregional, o objeto é colocado no bucket mais próximo. Para então disponibilizar o vídeo carregado para outras pessoas ao redor do mundo para download com a menor latência, você pode usar a CRR com replicação bidirecional. O uso de CRR com replicação bidirecional mantém o conteúdo de todos os buckets associados ao ponto de acesso multirregional sincronizado. Para ter mais informações sobre o uso de replicação com pontos de acesso multirregionais, consulte [Configurar a replicação para uso com pontos de acesso multirregionais](#).

Controles de failover para pontos de acesso multirregionais do Amazon S3.

Com os controles de failover para pontos de acesso multirregionais do Amazon S3, você pode manter a continuidade dos negócios durante interrupções no tráfego regional e, ao mesmo tempo, oferecer às aplicações uma arquitetura multirregional para atender às necessidades de conformidade e redundância. Se seu tráfego regional for interrompido, você poderá usar controles de failover para pontos de acesso multirregionais para selecionar quais Regiões da AWS por trás de um ponto de acesso multirregional do Amazon S3 vão processar as solicitações de acesso e armazenamento de dados.

Para oferecer suporte a failover, você pode configurar seu ponto de acesso multirregional em uma configuração ativa-passiva, com tráfego fluindo para a região ativa em condições normais e uma região passiva em espera para failover.

Por exemplo, para realizar o failover para uma Região da AWS de sua escolha, o tráfego é transferido da região primária (ativa) para a região secundária (passiva). Em uma configuração ativa-passiva como essa, um bucket é ativo e aceita tráfego, enquanto o outro bucket é passivo e não aceita tráfego. O bucket passivo é usado para recuperação de desastres. Ao iniciar o failover, todo o tráfego (como solicitações GET ou PUT) é direcionado para o bucket ativo (em uma região) e para longe do bucket passivo (em outra região).

Se você tiver a Replicação do S3 Entre Regiões Diferentes (CRR) habilitada com regras de replicação bidirecional, poderá manter seus buckets sincronizados durante um failover. Além disso, se você tiver a CRR habilitada em uma configuração ativa-ativa, os pontos de acesso multirregionais do Amazon S3 também poderão buscar dados do local do bucket mais próximo, o que melhora a performance da aplicação.

Suporte a Região da AWS

Com os controles de failover para pontos de acesso multirregionais do Amazon S3, seus buckets do S3 podem estar em qualquer uma das [17 regiões](#) em que os pontos de acesso multirregionais são compatíveis. Você pode iniciar o failover entre duas regiões quaisquer por vez.

Note

Embora o failover seja iniciado apenas entre duas regiões por vez, você pode atualizar separadamente os status de roteamento para várias regiões ao mesmo tempo em seu ponto de acesso multirregional.

Os tópicos a seguir demonstram como usar e gerenciar os controles de failover para pontos de acesso multirregionais do Amazon S3.

Tópicos

- [Estados de roteamento de pontos de acesso multirregionais do Amazon S3](#)
- [Como usar controles de failover para pontos de acesso multirregionais do Amazon S3](#)
- [Erros de controles de failover para pontos de acesso multirregionais do Amazon S3](#)

Estados de roteamento de pontos de acesso multirregionais do Amazon S3

Sua configuração de failover para pontos de acesso multirregionais do Amazon S3 determina o status de roteamento das Regiões da AWS que são usadas com o ponto de acesso multirregional. Você pode configurar seu ponto de acesso multirregional do Amazon S3 para que esteja em um estado ativo-ativo ou ativo-passivo.

- **Ativo-ativo:** em uma configuração ativa-ativa, todas as solicitações são enviadas automaticamente para a Região da AWS mais próxima em seu ponto de acesso multirregional. Depois que o ponto de acesso multirregional for configurado para ficar em um estado ativo-ativo, todas as regiões poderão receber tráfego. Se ocorrer interrupção do tráfego em uma configuração ativa-ativa, o tráfego da rede será automaticamente redirecionado para uma das regiões ativas.
- **Ativo-passivo:** em uma configuração ativa-passiva, as regiões ativas em seu ponto de acesso multirregional recebem tráfego e as passivas não. Se você pretende usar os controles de failover do S3 para iniciar um failover em uma situação de desastre, configure os pontos de acesso multirregionais em um estado ativo-passivo enquanto estiver testando e realizando o planejamento de recuperação de desastres.

Como usar controles de failover para pontos de acesso multirregionais do Amazon S3

Esta seção explica como gerenciar e usar os controles de failover para pontos de acesso multirregionais do Amazon S3 usando o AWS Management Console.

Há dois controles de failover na seção Failover configuration (Configuração de failover) na página de detalhes do ponto de acesso multirregional no AWS Management Console: Edit routing status (Editar status de roteamento) e Failover. Você pode usar esses controles da seguinte maneira:

- **Edit routing status (Editar status de roteamento):** você pode editar manualmente os status de roteamento de até 17 Regiões da AWS em uma única solicitação para seu ponto de acesso

multirregional escolhendo Edit routing status (Editar status de roteamento). Você pode usar a opção Edit routing status (Editar status de roteamento) para as seguintes finalidades:

- Para definir ou editar os status de roteamento de uma ou mais regiões em seu ponto de acesso multirregional
- Para criar uma configuração de failover para seu ponto de acesso multirregional configurando duas regiões em um estado ativo-passivo
- Para fazer o failover manualmente em suas regiões
- Para alternar manualmente o tráfego entre regiões
- Failover: ao iniciar o failover escolhendo Failover, você atualiza apenas os status de roteamento de duas regiões que já estejam configuradas em um estado ativo-passivo. Durante um failover que você iniciou escolhendo Failover, os status de roteamento entre as duas regiões são automaticamente alterados.

Editar os status de roteamento das regiões em seu ponto de acesso multirregional

Você pode atualizar manualmente os status de roteamento de até 17 Regiões da AWS em uma única solicitação para seu ponto de acesso multirregional escolhendo Edit routing status (Editar status de roteamento) na seção Failover configuration (Configuração de failover) na página de detalhes do ponto de acesso multirregional. No entanto, ao iniciar o failover escolhendo Failover, você atualiza apenas os status de roteamento de duas regiões que já estejam configuradas em um estado ativo-passivo. Durante um failover que você iniciou escolhendo Failover, os status de roteamento entre as duas regiões são automaticamente alterados.

Você pode usar Edit routing status (Editar status de roteamento) (conforme descrito no procedimento a seguir) para as seguintes finalidades:

- Para definir ou editar os status de roteamento de uma ou mais regiões em seu ponto de acesso multirregional
- Para criar uma configuração de failover para seu ponto de acesso multirregional configurando duas regiões em um estado ativo-passivo
- Para fazer o failover manualmente em suas regiões
- Para alternar manualmente o tráfego entre regiões

Uso do console do S3

Como atualizar os status de roteamento das regiões em seu ponto de acesso multirregional

1. Faça login no Console de Gerenciamento AWS.
2. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
3. No painel de navegação esquerdo, escolha Multi-Region Access Points (Pontos de acesso multirregionais).
4. Escolha o ponto de acesso multirregional que você deseja atualizar.
5. Escolha a guia Replication and failover (Replicação e failover).
6. Selecione uma ou mais regiões para as quais deseja editar o status do roteamento.

Note

Para iniciar um failover, pelo menos uma Região da AWS deve ser designada como Active (Ativa) e uma região deve ser designada como Passive (Passiva) em seu ponto de acesso multirregional.

7. Escolha Edit routing status (Editar status de roteamento).
8. Na caixa de diálogo exibida, selecione Active (Ativo) ou Passive (Passivo) para o Routing status (Status de roteamento) de cada região.

Um estado ativo permite que o tráfego seja roteado para a região. Um estado passivo impede que qualquer tráfego seja direcionado para a região.

Se você estiver criando uma configuração de failover para o ponto de acesso multirregional ou iniciando um failover, pelo menos uma Região da AWS deverá ser designada como Active (Ativa) e uma região deverá ser designada como Passive (Passiva) em seu ponto de acesso multirregional.

9. Escolha Save routing status (Salvar status de roteamento). O redirecionamento do tráfego leva cerca de 2 minutos.

Depois de enviar o status de roteamento das Regiões da AWS para seu ponto de acesso multirregional, você poderá verificar as alterações no status de roteamento. Para verificar essas alterações, acesse o Amazon CloudWatch em <https://console.aws.amazon.com/cloudwatch/> para monitorar a mudança do tráfego de solicitações de dados do Amazon S3 (por exemplo, solicitações

GET e PUT) entre regiões ativas e passivas. Nenhuma conexão existente será encerrada durante o failover. As conexões existentes continuarão até receberem um status de sucesso ou falha.

Usar a AWS CLI

Note

Você pode executar comandos de roteamento da AWS CLI de pontos de acesso multirregionais em qualquer uma destas cinco regiões:

- `ap-southeast-2`
- `ap-northeast-1`
- `us-east-1`
- `us-west-2`
- `eu-west-1`

O exemplo de comando a seguir atualiza sua configuração de roteamento do ponto de acesso multirregional atual. Para atualizar o status ativo ou passivo de um bucket, defina o valor de `TrafficDialPercentage` como `100` para ativo e como `0` para passivo. Neste exemplo, `DOC-EXAMPLE-BUCKET-1` é definido como ativo e `DOC-EXAMPLE-BUCKET-2` é definido como passivo. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control submit-multi-region-access-point-routes
--region ap-southeast-2
--account-id 111122223333
--mrp MultiRegionAccessPoint_ARN
--route-updates Bucket=DOC-EXAMPLE-BUCKET-1,TrafficDialPercentage=100
                Bucket=DOC-EXAMPLE-BUCKET-2,TrafficDialPercentage=0
```

O exemplo de comando a seguir obtém sua configuração de roteamento do ponto de acesso multirregional atualizada. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 111122223333
```

```
--map MultiRegionAccessPoint_ARN
```

Iniciar failover

Quando você inicia um failover escolhendo Failover na seção Failover configuration (Configuração de failover) na página de detalhes do ponto de acesso multirregional, o tráfego de solicitações do Amazon S3 é automaticamente transferido para uma Região da AWS alternativa. O processo de failover é concluído em 2 minutos.

Você pode iniciar um failover em duas Regiões da AWS quaisquer por vez (das [17 regiões](#) em que os pontos de acesso multirregionais são compatíveis). Depois, os eventos de failover são registrados em logs do AWS CloudTrail. Depois da conclusão do failover, você pode monitorar o tráfego do Amazon S3 e qualquer atualização de roteamento de tráfego para a nova região ativa no Amazon CloudWatch.

Important

Para manter todos os metadados e objetos sincronizados entre os buckets durante a replicação de dados, recomendamos que você crie regras de replicação bidirecional e habilite a sincronização de modificações da réplica antes de configurar controles de failover. As regras de replicação bidirecional ajudam a garantir que, quando os dados são gravados no bucket do Amazon S3 para o qual o tráfego faz failover, esses dados sejam replicados de volta para o bucket de origem. A sincronização de modificações da réplica ajuda a garantir que os metadados do objeto também sejam sincronizados entre os buckets durante a replicação bidirecional.

Para obter mais informações sobre como configurar a replicação para que ofereça suporte a failover, consulte [the section called “Replicação do bucket”](#).

Como iniciar um failover entre buckets replicados

1. Faça login no Console de Gerenciamento AWS.
2. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
3. No painel de navegação esquerdo, escolha Multi-Region Access Points (Pontos de acesso multirregionais).
4. Escolha o ponto de acesso multirregional que deseja usar para iniciar o failover.
5. Escolha a guia Replication and failover (Replicação e failover).

6. Role para baixo até a seção Failover configuration (Configuração de failover) e selecione duas Regiões da AWS.

 Note

Para iniciar um failover, pelo menos uma Região da AWS deve ser designada como Active (Ativa) e uma região deve ser designada como Passive (Passiva) em seu ponto de acesso multirregional. Um estado ativo permite que o tráfego seja direcionado para uma região. Um estado passivo impede que qualquer tráfego seja direcionado para a região.

7. Escolha Failover.
8. Na caixa de diálogo, selecione Failover novamente para iniciar o processo de failover. Durante esse processo, os status de roteamento das duas regiões são automaticamente alterados. Todo novo tráfego é direcionado para a região que se torna ativa, e o tráfego deixa de ser direcionado para a região que se torna passiva. O redirecionamento do tráfego leva cerca de 2 minutos.

Depois de iniciar o processo de failover, você pode verificar as alterações no tráfego. Para verificar essas alterações, acesse o Amazon CloudWatch em <https://console.aws.amazon.com/cloudwatch/> para monitorar a mudança do tráfego de solicitações de dados do Amazon S3 (por exemplo, solicitações GET e PUT) entre regiões ativas e passivas. Nenhuma conexão existente será encerrada durante o failover. As conexões existentes continuarão até receberem um status de sucesso ou falha.

Visualizar controles de roteamento de pontos de acesso multirregionais do Amazon S3

Uso do console do S3

Como visualizar controles de roteamento de pontos de acesso multirregionais do Amazon S3

1. Faça login no Console de Gerenciamento AWS.
2. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
3. No painel de navegação esquerdo, escolha Multi-Region Access Points (Pontos de acesso multirregionais).
4. Escolha o ponto de acesso multirregional que você deseja revisar.
5. Escolha a guia Replication and failover (Replicação e failover). Esta página exibe os detalhes e o resumo da configuração de roteamento do seu ponto de acesso multirregional, das regras de

replicação associadas e das métricas de replicação. Você pode ver o status de roteamento de suas regiões na seção Failover configuration (Configuração de failover).

Usar a AWS CLI

O exemplo de comando da AWS CLI a seguir obtém sua configuração de roteamento atual do ponto de acesso multirregional para a região especificada. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 111122223333
--mrap MultiRegionAccessPoint_ARN
```

Note

Esse comando só pode ser executado nestas cinco regiões:

- ap-southeast-2
- ap-northeast-1
- us-east-1
- us-west-2
- eu-west-1

Erros de controles de failover para pontos de acesso multirregionais do Amazon S3

Ao atualizar a configuração de failover do seu ponto de acesso multirregional, você pode encontrar um destes erros:

- HTTP 400 Solicitação Inválida: esse erro poderá ocorrer se você inserir um ARN de ponto de acesso multirregional inválido ao atualizar sua configuração de failover. Você pode confirmar o ARN do ponto de acesso multirregional revisando a política de pontos de acesso multirregionais. Para revisar ou atualizar a política de pontos de acesso multirregionais, consulte [Editar a política de pontos de acesso multirregionais](#). Esse erro também pode ocorrer se você usar uma string vazia ou aleatória ao atualizar seus controles de failover para pontos de acesso multirregionais do Amazon S3. Use o seguinte formato de ARN de ponto de acesso multirregional:

```
arn:aws:s3::account-id:accesspoint/MultiRegionAccessPoint_alias
```

- HTTP 503 Desacelerar: este erro ocorrerá se você enviar muitas solicitações em um curto período. Solicitações rejeitadas resultarão em um erro.
- HTTP 409 Conflito: este erro ocorre quando duas ou mais solicitações simultâneas de atualização de configuração de roteamento têm como alvo um único ponto de acesso multirregional. A primeira solicitação é bem-sucedida, mas todas as outras solicitações falham com um erro.
- HTTP 405 Método Não Permitido: este erro ocorre quando você seleciona um ponto de acesso multirregional com apenas uma Região da AWS ao iniciar o failover. Você deve selecionar duas regiões antes de iniciar o failover. Caso contrário, um erro será retornado.

Configurar a replicação para uso com pontos de acesso multirregionais

Quando você faz uma solicitação para um endpoint de ponto de acesso multirregional, o Amazon S3 roteia automaticamente a solicitação para o bucket mais próximo de você. O Amazon S3 não considera o conteúdo da solicitação ao tomar essa decisão. Se você fizer uma solicitação GET para um objeto, sua solicitação poderá ser roteada para um bucket que não tenha uma cópia desse objeto. Se isso acontecer, você receberá um erro do código de status HTTP 404 (Não encontrado). Para ter mais informações sobre direcionamento de solicitações de um ponto de acesso multirregional, consulte [the section called “Roteamento de solicitação”](#).

Se você quiser que o ponto de acesso multirregional consiga recuperar o objeto independentemente de qual bucket receba a solicitação, configure a Replicação do Amazon S3 Entre Regiões Diferentes (CRR).

Por exemplo, considere um ponto de acesso multirregional com três buckets:

- Um bucket chamado `my-bucket-usw2` na região `us-west-2` que contém o objeto `my-image.jpg`
- Um bucket chamado `my-bucket-aps1` na região `ap-south-1` que contém o objeto `my-image.jpg`
- Um bucket chamado `my-bucket-euc1` na região `eu-central-1` que não contém o 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 maior proximidade. Mesmo que seu objeto esteja em um bucket no ponto de acesso multirregional, você receberá um erro HTTP 404 Não Encontrado porque o bucket individual que recebeu sua solicitação não tinha o objeto.

Habilitar a Replicação Entre Regiões Diferentes (CRR) ajuda a evitar esse resultado. Com as regras de replicação apropriadas, o objeto `my-image.jpg` é copiado para o bucket `my-bucket-euc1`. Portanto, se o Amazon S3 rotear sua solicitação para esse bucket, agora você poderá recuperar o objeto.

A replicação funciona normalmente com buckets atribuídos a um ponto de acesso multirregional. O Amazon S3 não executa nenhum tratamento de replicação especial com buckets que estão em pontos de acesso multirregionais. Para obter mais informações sobre como configurar a replicação em seus buckets, consulte [Configuração da replicação](#).

Recomendações para usar a replicação com pontos de acesso multirregionais

Para obter a melhor performance de replicação ao trabalhar com pontos de acesso multirregionais, recomendamos o seguinte:

- Configure o Controle de Tempo de Replicação do S3 (S3 RTC). Para replicar dados em diferentes regiões dentro de um período previsível, você pode usar o S3 RTC. 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”](#). Não há cobranças adicionais por usar o S3 RTC. Para ter informações, consulte [Definição de preço do Amazon S3](#).
- Use a replicação bidirecional para ajudar a manter os buckets sincronizados quando os buckets são atualizados por meio do ponto de acesso multirregional. Para obter mais informações, consulte [the section called “Criar regras de replicação bidirecional para seu ponto de acesso multirregional”](#).
- Crie pontos de acesso multirregionais entre contas para replicar dados em buckets em Contas da AWS separadas. Essa abordagem oferece separação em nível de conta, para que os dados possam ser acessados e replicados em contas diferentes em regiões distintas, além do bucket de origem. A configuração de pontos de acesso multirregionais entre contas não tem custo adicional. Se você é proprietário de um bucket, mas não tem o ponto de acesso multirregional, paga somente pelos custos de transferência e solicitação de dados. Os proprietários de pontos de acesso multirregionais pagam pelos custos de roteamento de dados e aceleração da Internet. Para obter mais informações, consulte [Definição de preço do Amazon S3](#).

- Habilite a sincronização de modificações da réplica para cada regra de replicação a fim de também manter as alterações de metadados dos objetos sincronizadas. Para obter mais informações, consulte [Habilitar a sincronização de modificações da réplica](#).
- Habilite as métricas do Amazon CloudWatch para [monitorar os eventos de replicação](#). As taxas de métricas do CloudWatch se aplicam. Para obter mais informações, consulte [Preço do Amazon CloudWatch](#).

Tópicos

- [Criar regras de replicação unidirecional para seu ponto de acesso multirregional](#)
- [Criar regras de replicação bidirecional para seu ponto de acesso multirregional](#)
- [Visualizar as regras de replicação para seu ponto de acesso multirregional](#).

Criar regras de replicação unidirecional para seu ponto de acesso multirregional

As regras de replicação possibilitam a cópia assíncrona automática de objetos entre buckets. Uma regra de replicação unidirecional ajuda a garantir que os dados sejam totalmente replicados de um bucket de origem em uma Região da AWS para um bucket de destino em outra região. Quando a replicação unidirecional é configurada, uma regra de replicação do bucket de origem (DOC-EXAMPLE-BUCKET-1) para o bucket de destino (DOC-EXAMPLE-BUCKET-2) é criada. Como todas as regras de replicação, você pode aplicar a regra de replicação unidirecional a todo o bucket do Amazon S3 ou a um subconjunto de objetos filtrados com base em um prefixo ou tags de objeto.

Important

Recomendamos usar a replicação unidirecional se seus usuários consumirem apenas os objetos em seus buckets de destino. Se seus usuários fizerem upload ou modificarem os objetos em seus buckets de destino, use a replicação bidirecional para manter todos os seus buckets sincronizados. Também recomendamos replicação bidirecional se você planeja usar um ponto de acesso multirregional para o failover. Para configurar a replicação bidirecional, consulte [the section called “Criar regras de replicação bidirecional para seu ponto de acesso multirregional”](#).

Como criar uma regra de replicação unidirecional para 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 esquerdo, escolha Multi-Region Access Points (Pontos de acesso multirregionais).
3. Escolha o nome de seu ponto de acesso multirregional.
4. Escolha a guia Replication and failover (Replicação e failover).
5. Role para baixo até Replication rules (Regras de replicação) e escolha Create replication rules (Criar regras de replicação). Verifique se você tem permissões suficientes para criar a regra de replicação, caso contrário, o versionamento será desativado.

Note

Apenas é possível criar regras de replicação para buckets em sua própria conta. Para criar regras de replicação para buckets externos, os proprietários do bucket devem criar as regras de replicação para esses buckets.


6. Na página Criar regras de replicação, selecione o modelo Replicar objetos de um ou mais buckets de origem para um ou mais buckets de destino.

Important

Quando você cria regras de replicação usando esse modelo, todas as regras de replicação existentes que já estejam atribuídas ao bucket são substituídas. Para adicionar ou modificar regras de replicação existentes em vez de substituí-las, acesse a guia Management (Gerenciamento) de cada bucket no console e edite as regras na seção Replication rules (Regras de replicação). Você também pode adicionar ou modificar as regras de replicação existentes usando a AWS CLI, SDKs ou a API REST. Para obter mais informações, consulte [Configuração de replicação](#).


7. Na seção Origem e destino, em Buckets de origem, selecione um ou mais buckets dos quais você deseja replicar objetos. Todos os buckets (origem e destino) escolhidos para replicação devem ter o Versionamento do S3 ativado, e cada bucket deve estar em uma Região da AWS diferente. Para obter mais informações sobre o Versionamento do S3, consulte [Usar o versionamento em buckets do Amazon S3](#).

Em Buckets de destino, selecione um ou mais buckets dos quais você deseja replicar objetos.

 Note

Tenha as permissões de leitura e replicação necessárias para estabelecer a replicação, caso contrário, você encontrará erros. Para ter mais informações, consulte [Criar um perfil do IAM](#).

8. Na seção Replication rule configuration (Configuração da regra de replicação), escolha se a regra de replicação ficará Enabled (Ativada) ou Disabled (Desativada) quando for criada.

 Note

Você não pode inserir um nome na caixa Replication rule name (Nome da regra de replicação). Os nomes de regras de replicação são gerados com base na sua configuração ao criar a regra de replicação.

9. Na seção Scope (Escopo), escolha o escopo apropriado para sua replicação.
 - Para replicar todo o bucket, escolha Apply to all objects in the bucket (Aplicar a todos os objetos no bucket).
 - Para replicar um subconjunto dos objetos do bucket, escolha Limit the scope of this rule using one or more filters (Limitar o escopo desta regra usando um ou mais filtros).

Você pode filtrar os objetos usando um prefixo, etiquetas de objeto ou uma combinação de ambos.

- Para limitar a replicação para todos os objetos que têm nomes que começam com a mesma string (por exemplo, `pictures`), insira um prefixo na caixa Prefix (Prefixo).

Se você inserir um prefixo que seja o nome de uma pasta, será preciso usar um delimitadores, como uma / (barra), para indicar seu nível de hierarquia (por exemplo, `pictures/`). Para ter mais informações sobre prefixos, consulte [Organizar objetos usando prefixos](#).

- Para replicar todos os objetos que têm uma ou mais etiquetas de objeto, selecione Add tag (Adicionar etiqueta) e insira o par de chave/valor nas caixas. Para adicionar outra etiqueta, repita o procedimento. Para obter mais informações sobre etiquetas de objeto, consulte [Categorizando seu armazenamento usando tags](#).

10. Role para baixo até a seção Additional replication options (Opções de replicação adicionais) e selecione as opções de replicação que deseja aplicar.

 Note

Recomendamos aplicar as seguintes opções:

- Replication time control (RTC) (Controle de Tempo de Replicação (RTC)): para replicar seus dados em diferentes regiões dentro de um período previsível, você pode usar o Controle de Tempo de Replicação do S3 (S3 RTC). 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”](#).
- Replication metrics and notifications (Métricas e notificações de replicação): habilite as métricas do Amazon CloudWatch para monitorar eventos de replicação.
- Replicação de marcadores de exclusão: os marcadores de exclusão criados pelas operações de exclusão do S3 serão replicados. Os marcadores de exclusão criados pelas regras do ciclo de vida não são replicados. Para ter mais informações, consulte [Replicar marcadores de exclusão entre buckets](#).

Há cobranças adicionais pelas métricas e notificações de replicação do S3 RTC e do CloudWatch. Para obter mais informações, consulte [Definição de preço do Amazon S3](#) e [Preço do Amazon CloudWatch](#).

11. Se você estiver escrevendo uma nova regra de replicação que substitui uma existente, selecione I acknowledge that by choosing Create replication rules, these existing replication rules will be overwritten (Eu reconheço que, ao escolher Criar regras de replicação, essas regras de replicação existentes serão substituídas).
12. Selecione Criar regras de replicação para criar e salvar suas novas regras de replicação unidirecional.

Criar regras de replicação bidirecional para seu ponto de acesso multirregional

As regras de replicação possibilitam a cópia assíncrona automática de objetos entre buckets. Uma regra de replicação bidirecional garante que os dados sejam totalmente replicados entre dois ou mais buckets em Regiões da AWS diferentes. Quando a replicação bidirecional é configurada, uma

regra de replicação do bucket de origem (DOC-EXAMPLE-BUCKET-1) para o bucket que contém as réplicas (DOC-EXAMPLE-BUCKET-2) é criada. Depois, é criada uma segunda regra de replicação do bucket que contém as réplicas (DOC-EXAMPLE-BUCKET-2) para o bucket de origem (DOC-EXAMPLE-BUCKET-1).

Como todas as regras de replicação, você pode aplicar a regra de replicação bidirecional a todo o bucket do Amazon S3 ou a um subconjunto de objetos filtrados com base em um prefixo ou tags de objeto. Também é possível manter as alterações de metadados dos objetos sincronizadas ao [habilitar a sincronização de modificações da réplica](#) para cada regra de replicação. Você pode ativar a sincronização de modificações da réplica por meio do console do Amazon S3, a AWS CLI, os AWS SDKs, a API REST do Amazon S3 ou o AWS CloudFormation.

Para monitorar o progresso da replicação de objetos e metadados de objetos no Amazon CloudWatch, habilite as métricas e notificações da Replicação do S3. Para obter mais informações, consulte [Monitoramento do progresso com métricas de replicação e notificações de eventos do Amazon S3](#).

Como criar uma regra de replicação bidirecional para 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 esquerdo, escolha Multi-Region Access Points (Pontos de acesso multirregionais).
3. Escolha o nome do ponto de acesso multirregional que você deseja atualizar.
4. Escolha a guia Replication and failover (Replicação e failover).
5. Role para baixo até Replication rules (Regras de replicação) e escolha Create replication rules (Criar regras de replicação).
6. Na página Create replication rules (Criar regras de replicação), escolha o modelo Replicate objects among all specified buckets (Replicar objetos entre todos os buckets especificados). O modelo Replicate objects among all specified buckets (Replicar objetos entre todos os buckets especificados) configura a replicação bidirecional (com recursos de failover) para seus buckets.

 Important

Quando você cria regras de replicação usando esse modelo, todas as regras de replicação existentes que já estejam atribuídas ao bucket são substituídas.


Para adicionar ou modificar regras de replicação existentes em vez de substituí-las, acesse a guia Management (Gerenciamento) de cada bucket no console e edite as regras na seção Replication rules (Regras de replicação). Você também pode adicionar ou modificar as regras de replicação existentes usando a AWS CLI, os AWS SDKs ou a API REST do Amazon S3. Para obter mais informações, consulte [Configuração de replicação](#).

7. Na seção Buckets, selecione pelo menos dois buckets dos quais você deseja replicar objetos. Todos os buckets escolhidos para replicação devem ter o Versionamento do S3 ativado, e cada bucket deve estar em uma Região da AWS diferente. Para obter mais informações sobre o Versionamento do S3, consulte [Usar o versionamento em buckets do Amazon S3](#).

 Note

Tenha as permissões de leitura e replicação necessárias para estabelecer a replicação, caso contrário, você encontrará erros. Para ter mais informações, consulte [Criar um perfil do IAM](#).

8. Na seção Replication rule configuration (Configuração da regra de replicação), escolha se a regra de replicação ficará Enabled (Ativada) ou Disabled (Desativada) quando for criada.

 Note

Você não pode inserir um nome na caixa Replication rule name (Nome da regra de replicação). Os nomes de regras de replicação são gerados com base na sua configuração ao criar a regra de replicação.

9. Na seção Scope (Escopo), escolha o escopo apropriado para sua replicação.
 - Para replicar todo o bucket, escolha Apply to all objects in the bucket (Aplicar a todos os objetos no bucket).
 - Para replicar um subconjunto dos objetos do bucket, escolha Limit the scope of this rule using one or more filters (Limitar o escopo desta regra usando um ou mais filtros).

Você pode filtrar os objetos usando um prefixo, etiquetas de objeto ou uma combinação de ambos.

- Para limitar a replicação para todos os objetos que têm nomes que começam com a mesma string (por exemplo, pictures), insira um prefixo na caixa Prefix (Prefixo).

Se você inserir um prefixo que seja o nome de uma pasta, use uma / (barra) como o último caractere (por exemplo, `pictures/`).

- Para replicar todos os objetos que têm uma ou mais etiquetas de objeto, selecione **Add tag** (Adicionar etiqueta) e insira o par de chave/valor nas caixas. Para adicionar outra etiqueta, repita o procedimento. Para obter mais informações sobre etiquetas de objeto, consulte [Categorizando seu armazenamento usando tags](#).

10. Role para baixo até a seção **Additional replication options** (Opções de replicação adicionais) e selecione as opções de replicação que deseja aplicar.

Note

Recomendamos que você aplique as seguintes opções, especialmente se pretender configurar um ponto de acesso multirregional para oferecer suporte a failover:

- **Replication time control (RTC) (Controle de Tempo de Replicação (RTC))**: para replicar seus dados em diferentes regiões dentro de um período previsível, você pode usar o Controle de Tempo de Replicação do S3 (S3 RTC). 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”](#).
- **Replication metrics and notifications (Métricas e notificações de replicação)**: habilite as métricas do Amazon CloudWatch para monitorar eventos de replicação.
- **Replicação de marcadores de exclusão**: os marcadores de exclusão criados pelas operações de exclusão do S3 serão replicados. Os marcadores de exclusão criados pelas regras do ciclo de vida não são replicados. Para ter mais informações, consulte [Replicar marcadores de exclusão entre buckets](#).
- **Replica modification sync (Sincronização de modificações da réplica)**: Habilite a sincronização de modificações da réplica para cada regra de replicação a fim de também manter as alterações de metadados dos objetos sincronizadas. Para obter mais informações, consulte [Habilitar a sincronização de modificações da réplica](#).

Há cobranças adicionais pelas métricas e notificações de replicação do S3 RTC e do CloudWatch. Para obter mais informações, consulte [Definição de preço do Amazon S3](#) e [Preço do Amazon CloudWatch](#).

11. Se você estiver escrevendo uma nova regra de replicação que substitui uma existente, selecione I acknowledge that by choosing Create replication rules, these existing replication rules will be overwritten (Eu reconheço que, ao escolher Criar regras de replicação, essas regras de replicação existentes serão substituídas).
12. Escolha Create replication rules (Criar regras de replicação) para criar e salvar suas novas regras de replicação bidirecional.

Visualizar as regras de replicação para seu ponto de acesso multirregional.

Com os pontos de acesso multirregionais, você pode configurar regras de replicação unidirecional ou regras de replicação bidirecional (bidirecional). Para ter informações sobre como gerenciar suas regras de replicação, consulte [Gerenciar regras de replicação usando o console do Amazon S3](#).

Como visualizar as regras de replicação para seu 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 esquerdo, escolha Multi-Region Access Points (Pontos de acesso multirregionais).
3. Escolha o nome de seu ponto de acesso multirregional.
4. Escolha a guia Replication and failover (Replicação e failover).
5. Role para baixo até a seção Regras de replicação. Esta seção lista todas as regras de replicação criadas para seu ponto de acesso multirregional.

Note

Se você adicionou um bucket de outra conta a esse ponto de acesso multirregional, deverá ter a permissão `s3:GetBucketReplication` do proprietário do bucket para visualizar as regras de replicação desse bucket.

Usar pontos de acesso multirregionais com operações de API compatíveis

O Amazon S3 fornece um conjunto de operações para o gerenciamento de pontos de acesso multirregionais. 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 Amazon S3 são sempre síncronas. O console aguarda até que a solicitação seja concluída antes de permitir que você envie outra solicitação.

É possível visualizar o status atual e os resultados das operações assíncronas usando o console ou usar `DescribeMultiRegionAccessPointOperation` na AWS CLI, AWS SDKs 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`. Quando você inclui o token de rastreamento, o Amazon S3 retorna o status atual e os resultados da operação especificada, incluindo erros ou informações relevantes de recursos. O Amazon S3 executa operações de `DescribeMultiRegionAccessPointOperation` de forma síncrona.

Todas as solicitações do ambiente de gerenciamento para criar ou manter pontos de acesso multirregionais devem ser encaminhadas para a região US West (Oregon). Com relação a solicitações de plano de dados de pontos de acesso multirregionais, as regiões não precisam ser especificadas. Para o ambiente de gerenciamento de failover do ponto de acesso multirregional, a solicitação deve ser encaminhada a uma das cinco regiões compatíveis. Consulte mais informações sobre as regiões compatíveis com o ponto de acesso multirregional em [Restrições e limitações de pontos de acesso multirregionais](#).

Além disso, é preciso conceder a permissão `s3:ListAllMyBuckets` para o usuário, perfil ou outra entidade do AWS Identity and Access Management (IAM) que faz uma solicitação para gerenciar um ponto de acesso multirregional.

Os exemplos a seguir demonstram como usar pontos de acesso multirregionais com operações compatíveis no Amazon S3.

Tópicos

- [Compatibilidade de pontos de acesso multirregionais com Serviços da AWS e AWS SDKs](#)
- [Compatibilidade de pontos de acesso multirregionais com operações do S3](#)
- [Visualizar a configuração de roteamento de pontos de acesso multirregionais](#)
- [Atualizar a política subjacente de buckets do Amazon S3](#)

- [Atualizar a configuração de roteamento de um ponto de acesso multirregional](#)
- [Adicionar um objeto a um bucket no ponto de acesso multirregional](#)
- [Recuperar objetos do ponto de acesso multirregional](#)
- [Listar objetos que estão armazenados em um bucket subjacente ao ponto de acesso multirregional](#)
- [Usar um URL pré-assinado com pontos de acesso multirregionais](#)
- [Usar um bucket que é configurado com pagamentos do solicitante com pontos de acesso multirregionais](#)

Compatibilidade de pontos de acesso multirregionais com Serviços da AWS e AWS SDKs


Para usar um ponto de acesso multirregional com aplicações que exigem um nome de bucket do Amazon S3, use o nome do recurso da Amazon (ARN) do ponto de acesso multirregional ao fazer solicitações usando um AWS SDK. Para conferir quais AWS SDKs são compatíveis com pontos de acesso multirregionais, consulte [Compatibilidade com AWS SDKs](#).

Compatibilidade de pontos de acesso multirregionais com operações do S3

É possível usar as operações de API a seguir do plano de dados do Amazon S3 para executar ações em objetos em buckets associados ao ponto de acesso multirregional. As seguintes operações do S3 podem aceitar ARNs de pontos de acesso multirregionais:

- [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 permitem operações de cópia que os utilizem somente como destino ao usar o ARN do ponto de acesso multirregional.

É possível usar as seguintes operações do ambiente de gerenciamento do Amazon S3 para criar e gerenciar pontos de acesso multirregionais:

- [CreateMultiRegionAccessPoint](#)
- [DescribeMultiRegionAccessPointOperation](#)
- [GetMultiRegionAccessPoint](#)
- [GetMultiRegionAccessPointPolicy](#)
- [GetMultiRegionAccessPointPolicyStatus](#)
- [GetMultiRegionAccessPointRoutes](#)
- [ListMultiRegionAccessPoints](#)
- [PutMultiRegionAccessPointPolicy](#)
- [SubmitMultiRegionAccessPointRoutes](#)

Visualizar a configuração de roteamento de pontos de acesso multirregionais

AWS CLI

O exemplo de comando a seguir recupera a configuração de roteamento de pontos de acesso multirregionais para que você possa ver os status de roteamento atuais de seus buckets. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 111122223333
--mrap arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap
```

SDK for Java

O código de SDK para Java a seguir recupera a configuração de roteamento de pontos de acesso multirregionais para que você possa ver os status de roteamento atuais de seus buckets. Para usar esse exemplo de sintaxe, substitua os *user input placeholders* por suas próprias informações.

```
S3ControlClient s3ControlClient = S3ControlClient.builder()
    .region(Region.US_EAST_1)
    .credentialsProvider(credentialsProvider)
    .build();

GetMultiRegionAccessPointRoutesRequest request =
    GetMultiRegionAccessPointRoutesRequest.builder()
        .accountId("111122223333")
        .mrap("arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap")
        .build();

GetMultiRegionAccessPointRoutesResponse response =
    s3ControlClient.getMultiRegionAccessPointRoutes(request);
```

SDK for JavaScript

O código de SDK para JavaScript a seguir recupera a configuração de roteamento de pontos de acesso multirregionais para que você possa ver os status de roteamento atuais de seus buckets. Para usar esse exemplo de sintaxe, substitua os *user input placeholders* por suas próprias informações.

```
const REGION = 'us-east-1'

const s3ControlClient = new S3ControlClient({
  region: REGION
})

export const run = async () => {
  try {
    const data = await s3ControlClient.send(
      new GetMultiRegionAccessPointRoutesCommand({
        AccountId: '111122223333',
        Mrap: 'arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap',
      })
    )
    console.log('Success', data)
    return data
  } catch (err) {
    console.log('Error', err)
  }
}

run()
```

SDK for Python

O código de SDK para Python a seguir recupera a configuração de roteamento de pontos de acesso multirregionais para que você possa ver os status de roteamento atuais de seus buckets. Para usar esse exemplo de sintaxe, substitua os *user input placeholders* por suas próprias informações.

```
s3.get_multi_region_access_point_routes(
    AccountId=111122223333,
    Mrap=arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap)['Routes']
```

Atualizar a política subjacente de buckets do Amazon S3

Para conceder acesso adequado, você também deve atualizar a política subjacente de buckets do Amazon S3. Os exemplos a seguir delegam o controle de acesso à política de pontos de acesso multirregionais. Depois de delegar o controle de acesso à política de ponto de acesso multirregional,

a política de bucket não é mais usada para controle de acesso quando as solicitações são feitas por meio desse ponto de acesso multirregional.

Veja a seguir um exemplo de política de bucket que delega o controle de acesso à política de pontos de acesso multirregionais. Para usar esse exemplo de política de bucket, substitua os *user input placeholders* por suas próprias informações. Para aplicar essa política por meio do comando `put-bucket-policy` da AWS CLI, conforme mostrado no próximo exemplo, salve a política em um arquivo, como `policy.json`.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Principal": { "AWS": "*" },
    "Effect": "Allow",
    "Action": ["s3:*"],
    "Resource": ["arn:aws:s3:::111122223333/*", "arn:aws:s3:::DOC-EXAMPLE-BUCKET"],
    "Condition": {
      "StringEquals": {
        "s3:DataAccessPointAccount": "444455556666"
      }
    }
  }
}
```

O seguinte exemplo de comando `put-bucket-policy` associa a política atualizada ao bucket do S3:

```
aws s3api put-bucket-policy
--bucket DOC-EXAMPLE-BUCKET
--policy file:///tmp/policy.json
```

Atualizar a configuração de roteamento de um ponto de acesso multirregional

O exemplo de comando a seguir atualiza a configuração de roteamento do ponto de acesso multirregional. Os comandos de roteamento de pontos de acesso multirregionais podem ser executados nestas cinco regiões:

- `ap-southeast-2`
- `ap-northeast-1`
- `us-east-1`

- `us-west-2`
- `eu-west-1`

Em uma configuração de roteamento de ponto de acesso multirregional, você pode definir buckets com status de roteamento ativo ou passivo. Os buckets ativos recebem tráfego, enquanto os passivos não recebem. Você pode definir o status de roteamento de um bucket definindo o valor de `TrafficDialPercentage` do bucket como `100` para ativo ou como `0` para passivo.

AWS CLI

O exemplo de comando a seguir atualiza a configuração de roteamento do seu ponto de acesso multirregional. Neste exemplo, `DOC-EXAMPLE-BUCKET1` é definido como ativo e `DOC-EXAMPLE-BUCKET2` é definido como passivo. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control submit-multi-region-access-point-routes
--region ap-southeast-2
--account-id 111122223333
--mrap arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap
--route-updates Bucket=DOC-EXAMPLE-BUCKET1,TrafficDialPercentage=100
                Bucket=DOC-EXAMPLE-BUCKET2,TrafficDialPercentage=0
```

SDK for Java

O código de SDK para Java a seguir atualiza a configuração de roteamento do seu ponto de acesso multirregional. Para usar esse exemplo de sintaxe, substitua os *user input placeholders* por suas próprias informações.

```
S3ControlClient s3ControlClient = S3ControlClient.builder()
    .region(Region.ap-southeast-2)
    .credentialsProvider(credentialsProvider)
    .build();

SubmitMultiRegionAccessPointRoutesRequest request =
    SubmitMultiRegionAccessPointRoutesRequest.builder()
        .accountId("111122223333")
        .mrap("arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap")
        .routeUpdates(
            MultiRegionAccessPointRoute.builder()
                .region("eu-west-1")
                .trafficDialPercentage(100)
```



```
        .build(),
    MultiRegionAccessPointRoute.builder()
        .region("ca-central-1")
        .bucket("111122223333")
        .trafficDialPercentage(0)
        .build()
    )
    .build();
```

```
SubmitMultiRegionAccessPointRoutesResponse response =
    s3ControlClient.submitMultiRegionAccessPointRoutes(request);
```

SDK for JavaScript

O código de SDK para JavaScript a seguir atualiza a configuração de roteamento do seu ponto de acesso multirregional. Para usar esse exemplo de sintaxe, substitua os *user input placeholders* por suas próprias informações.

```
const REGION = 'ap-southeast-2'

const s3ControlClient = new S3ControlClient({
  region: REGION
})

export const run = async () => {
  try {
    const data = await s3ControlClient.send(
      new SubmitMultiRegionAccessPointRoutesCommand({
        AccountId: '111122223333',
        Mrap: 'arn:aws:s3:::111122223333:accesspoint/abcdef0123456.mrap',
        RouteUpdates: [
          {
            Region: 'eu-west-1',
            TrafficDialPercentage: 100,
          },
          {
            Region: 'ca-central-1',
            Bucket: 'DOC-EXAMPLE-BUCKET1',
            TrafficDialPercentage: 0,
          },
        ],
      })
    )
  }
}
```

```
    console.log('Success', data)
    return data
  } catch (err) {
    console.log('Error', err)
  }
}

run()
```

SDK for Python

O código de SDK para Python a seguir atualiza a configuração de roteamento do seu ponto de acesso multirregional. Para usar esse exemplo de sintaxe, substitua os *user input placeholders* por suas próprias informações.

```
s3.submit_multi_region_access_point_routes(
    AccountId=111122223333,
    Mrap=arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrp,
    RouteUpdates= [{
        'Bucket': DOC-EXAMPLE-BUCKET,
        'Region': ap-southeast-2,
        'TrafficDialPercentage': 10
    }])
```

Adicionar um objeto a um bucket no ponto de acesso multirregional

Para adicionar um objeto ao bucket associado ao ponto de acesso multirregional, é possível usar a operação [PutObject](#). Para manter todos os buckets no ponto de acesso multirregional sincronizados, habilite a [Replicação entre regiões](#).

Note

Para usar essa operação, é necessário ter a permissão `s3:PutObject` do ponto de acesso multirregional. Para obter mais informações sobre os requisitos de permissão do ponto de acesso multirregional, consulte [Permissões](#).

AWS CLI

O exemplo de plano de dados a seguir carrega *example.txt* para o ponto de acesso multirregional especificado. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
aws s3api put-object --bucket
arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap --key example.txt --
body example.txt
```

SDK for Java

```
S3Client s3Client = S3Client.builder()
    .build();

PutObjectRequest objectRequest = PutObjectRequest.builder()
    .bucket("arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap")
    .key("example.txt")
    .build();

s3Client.putObject(objectRequest, RequestBody.fromString("Hello S3!"));
```

SDK for JavaScript

```
const client = new S3Client({});

async function putObjectExample() {
  const command = new PutObjectCommand({
    Bucket: "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap",
    Key: "example.txt",
    Body: "Hello S3!",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
}
```

SDK for Python

```
import boto3

client = boto3.client('s3')
client.put_object(
    Bucket='arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap',
    Key='example.txt',
    Body='Hello S3!'
)
```

Recuperar objetos do ponto de acesso multirregional

Para recuperar objetos do ponto de acesso multirregional, é possível usar a operação [GetObject](#).

Note

Para usar essa operação de API, é necessário ter a permissão `s3:GetObject` do ponto de acesso multirregional. Para obter mais informações sobre os requisitos de permissões do ponto de acesso multirregional, consulte [Permissões](#).

AWS CLI

O exemplo de solicitação de plano de dados a seguir recupera `example.txt` do ponto de acesso multirregional especificado e o baixa como `downloaded_example.txt`. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
aws s3api get-object --bucket
arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap --
key example.txt downloaded_example.txt
```

SDK for Java

```
S3Client s3 = S3Client
    .builder()
    .build();

GetObjectRequest getObjectRequest = GetObjectRequest.builder()
```

```
.bucket("arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap")
.key("example.txt")
.build();

s3Client.getObject(getObjectRequest);
```

SDK for JavaScript

```
const client = new S3Client({})

async function getObjectExample() {
  const command = new GetObjectCommand({
    Bucket: "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap",
    Key: "example.txt"
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
}
```

SDK for Python

```
import boto3

client = boto3.client('s3')
client.get_object(
    Bucket='arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap',
    Key='example.txt'
)
```

Listar objetos que estão armazenados em um bucket subjacente ao ponto de acesso multirregional

Para retornar uma lista de objetos que estão armazenados em um bucket subjacente ao ponto de acesso multirregional, use a operação [ListObjectsV2](#). No exemplo de comando a seguir, todos os objetos do ponto de acesso multirregional especificado são listados usando o ARN do ponto de acesso multirregional. Nesse caso, o ARN do ponto de acesso multirregional é:

```
arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrp
```

Note

Para usar essa operação de API, é necessário ter a permissão `s3:ListBucket` para o ponto de acesso multirregional e o bucket subjacente. Para obter mais informações sobre os requisitos de permissões do ponto de acesso multirregional, consulte [Permissões](#).

AWS CLI

O exemplo de solicitação de plano de dados a seguir lista os objetos no bucket subjacente ao ponto de acesso multirregional especificado pelo ARN. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
aws s3api list-objects-v2 --bucket  
arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrp
```

SDK for Java

```
S3Client s3Client = S3Client.builder()  
    .build();  
  
String bucketName = "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrp";  
  
ListObjectsV2Request listObjectsRequest = ListObjectsV2Request  
    .builder()  
    .bucket(bucketName)  
    .build();  
  
s3Client.listObjectsV2(listObjectsRequest);
```

SDK for JavaScript

```
const client = new S3Client({});  
  
async function listObjectsExample() {  
    const command = new ListObjectsV2Command({  
        Bucket: "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrp",  
    });
```

```
try {
  const response = await client.send(command);
  console.log(response);
} catch (err) {
  console.error(err);
}
}
```

SDK for Python

```
import boto3

client = boto3.client('s3')
client.list_objects_v2(
  Bucket='arn:aws:s3:::123456789012:accesspoint/abcdef0123456.mrap'
)
```

Usar um URL pré-assinado com pontos de acesso multirregionais

É possível usar um URL pré-assinado para gerar um URL que permite acessar os buckets do Amazon S3 por meio de um ponto de acesso multirregional do Amazon S3. Ao criar um URL pré-assinado, você o associa a uma ação de objeto específica, como um upload no S3 (`PutObject`) ou um download do S3 (`GetObject`). É possível compartilhar o URL pré-assinado, e qualquer pessoa com acesso a ele pode executar a ação incorporada no URL como se fosse o usuário de assinatura original.

Os URLs pré-assinados têm uma data de validade. Quando o prazo de validade for atingido, o URL não funcionará mais.

Antes de usar os pontos de acesso multirregionais do S3 com URLs pré-assinados, confira a [compatibilidade de AWS SDKs](#) com o algoritmo SigV4A. Verifique se sua versão de SDK oferece suporte ao SigV4A como a implementação de assinatura usada para assinar as solicitações globais da Região da AWS. Para obter mais informações sobre como usar URLs pré-assinados com o Amazon S3, consulte [Compartilhar objetos usando URLs pré-assinados](#).

Os exemplos a seguir mostram como você pode usar pontos de acesso multirregionais com URLs pré-assinados. Para usar esses exemplos, substitua os *user input placeholders* por suas próprias informações.

AWS CLI

```
aws s3 presign
arn:aws:s3::123456789012:accesspoint/MultiRegionAccessPoint_alias/example-file.txt
```

SDK for Python

```
import logging
import boto3
from botocore.exceptions import ClientError

s3_client = boto3.client('s3',aws_access_key_id='xxx',aws_secret_access_key='xxx')
s3_client.generate_presigned_url(HttpMethod='PUT',ClientMethod="put_object",
    Params={'Bucket':'arn:aws:s3::123456789012:accesspoint/
abcdef0123456.mrap','Key':'example-file'})
```

SDK for Java

```
S3Presigner s3Presigner = S3Presigner.builder()
    .credentialsProvider(StsAssumeRoleCredentialsProvider.builder()
        .refreshRequest(assumeRole)
        .stsClient(stsClient)
        .build())
    .build();

GetObjectRequest getObjectRequest = GetObjectRequest.builder()
    .bucket("arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap")
    .key("example-file")
    .build();

GetObjectPresignRequest preSignedReq = GetObjectPresignRequest.builder()
    .getObjectRequest(getObjectRequest)
    .signatureDuration(Duration.ofMinutes(10))
    .build();

PresignedGetObjectRequest presignedGetObjectRequest =
    s3Presigner.presignGetObject(preSignedReq);
```


Note

Para usar o SigV4A com credenciais de segurança temporárias, por exemplo, ao usar perfis do IAM, solicite as credenciais temporárias de um endpoint regional no AWS Security Token Service (AWS STS), em vez de um endpoint global. Se você usar o endpoint global para o AWS STS (`sts.amazonaws.com`), o AWS STS vai gerar credenciais temporárias de um endpoint global, o que não é compatível com o Sig4A. Como resultado, você receberá um erro. Para resolver esse problema, use qualquer um dos [Endpoints regionais para AWS STS](#) listados.

Usar um bucket que é configurado com pagamentos do solicitante com pontos de acesso multirregionais

Se o bucket do S3 associado aos pontos de acesso multirregionais estiver [configurado para usar pagamentos do solicitante](#), o solicitante pagará pela solicitação do bucket, pelo download e por todos os custos relacionados aos pontos de acesso multirregionais. Para obter mais informações, consulte [Preço do Amazon S3](#).

Veja um exemplo de uma solicitação de plano de dados para um ponto de acesso multirregional conectado a um bucket de pagamentos do solicitante.

AWS CLI

Para baixar objetos de um ponto de acesso multirregional que está conectado a um bucket de pagamentos do solicitante, é necessário especificar `--request-payer requester` como parte da solicitação [get-object](#). Também é necessário especificar o nome do arquivo no bucket e o local onde o arquivo baixado deve ser armazenado.

```
aws s3api get-object --bucket MultiRegionAccessPoint_ARN --request-payer requester  
--key example-file-in-bucket.txt example-location-of-downloaded-file.txt
```

SDK for Java

Para baixar objetos de um ponto de acesso multirregional que está conectado a um bucket de pagamentos do solicitante, você deve especificar `RequestPayer.REQUESTER` como parte da solicitação `GetObject`. Você também deve especificar o nome do arquivo no bucket, bem como o local onde ele deve ser armazenado.

```
GetObjectResponse getObjectResponse = s3Client.getObject(GetObjectRequest.builder()
    .key("example-file.txt")
    .bucket("arn:aws:s3::
123456789012:accesspoint/abcdef0123456.mrap")
    .requestPayer(RequestPayer.REQUESTER)
    .build()
).response();
```

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 operações de API 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 denominado `my-bucket-usw2` na região `us-west-2` que contém o objeto `my-image.jpg`.
- Um bucket denominado `my-bucket-aps1` na região `ap-south-1` que contém o objeto `my-image.jpg`.
- Um bucket denominado `my-bucket-euc1` na região `eu-central-1` que não contém um objeto denominado `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, em vez disso, você fizer a solicitação por meio do ponto de acesso multirregional, primeiro o Amazon S3 determinará quais dos buckets nas diferentes regiões é o mais próximo. 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 retransmitir 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`.

Important

Pontos de acesso multirregionais não estão cientes do conteúdo de dados de buckets subjacentes. Portanto, o bucket que recebe a solicitação pode não conter os dados solicitados. Por exemplo, se o Amazon S3 determinar que o bucket `my-bucket-euc1` é o mais próximo, 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](#). Para obter mais informações sobre o AWS CloudTrail, consulte [What is AWS CloudTrail? \(O que é?\)](#) no AWS CloudTrail Guia do usuário.

Monitoramento e registro em log de solicitações feitas para operações de API de gerenciamento de pontos de acesso multirregionais

O Amazon S3 fornece várias operações de API para gerenciar pontos de acesso multirregionais, como `CreateMultiRegionAccessPoint` e `GetMultiRegionAccessPointPolicy`. Quando você faz solicitações para essas operações de API usando a AWS Command Line Interface (AWS CLI), AWS SDKs ou a API REST do Amazon S3, 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 Amazon S3 processa solicitações de `DescribeAsyncOperation` de forma síncrona. Para visualizar o status de solicitações assíncronas, você pode usar o console do Amazon S3, a AWS CLI, SDKs ou a API REST.

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.

Usar o 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 os pontos de acesso multirregionais e o registro em log do CloudTrail, você pode identificar o seguinte:

- Quem ou o que realizou qual ação
- Quais recursos foram utilizados
- Quando ocorreu o evento
- Outros detalhes sobre o evento

Você pode usar essas informações de registro em log para ajudar a analisar e responder às atividades que ocorreram por meio de seus pontos de acesso multirregionais.

Como configurar o AWS CloudTrail para pontos de acesso multirregionais

Para ativar o registro em log do CloudTrail para quaisquer operações relacionadas à criação ou à manutenção de pontos de acesso multirregionais, você deve configurar o registro em log do CloudTrail para registrar os eventos na região Oeste dos EUA (Oregon). Você deve definir a configuração de registro em log dessa forma independentemente da região em que você esteja ao fazer a solicitação ou a quais regiões o ponto de acesso multirregional oferece suporte. Todas as solicitações para criar ou manter um ponto de acesso multirregional são encaminhadas para a região Oeste dos EUA (Oregon). Recomendamos que você adicione essa região a uma trilha existente ou crie uma trilha que contenha essa região e todas as regiões associadas ao ponto de acesso multirregional.

O Amazon S3 registra todas as solicitações feitas por meio de pontos de acesso multirregionais e solicitações feitas às operações de API 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 terão um nome de host de `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`.

Os pontos de acesso multirregionais encaminham as solicitações para o bucket mais próximo. Devido a esse comportamento, quando você estiver observando os logs do CloudTrail de um ponto de acesso multirregional, você verá solicitações sendo feitas aos buckets subjacentes. Algumas dessas solicitações podem ser solicitações diretas para o bucket e não roteadas pelo ponto de acesso multirregional. Lembre-se disso 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 em log de solicitações feitas para operações de API de gerenciamento de pontos de acesso multirregionais](#).

Para obter mais informações sobre o AWS CloudTrail, consulte [What is AWS CloudTrail? \(O que é?\)](#) no AWS CloudTrail Guia do usuário.

Segurança do Amazon S3

A segurança na nuvem da AWS é 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, como a confidencialidade de seus dados, os requisitos da sua organização, 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.

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Tópicos

- [Proteção de dados no Amazon S3](#)
- [Proteger dados com criptografia](#)
- [Privacidade do tráfego entre redes](#)
- [AWS PrivateLink for Amazon S3](#)
- [Gerenciamento de identidade e acesso para o Amazon S3](#)
- [Gerenciar o acesso com a funcionalidade Concessões de Acesso do S3](#)
- [Gerenciar o acesso com ACLs](#)
- [Usar o compartilhamento de recursos de origem cruzada \(CORS\)](#)
- [Bloquear o acesso público ao armazenamento do Amazon S3](#)
- [Revisar o acesso de bucket usando o IAM Access Analyzer para S3](#)
- [Verificar a propriedade do bucket com a condição de proprietário do bucket](#)
- [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#)
- [Registrar em log e monitorar no Amazon S3](#)
- [Validação de conformidade para o Amazon S3](#)
- [Resiliência no Amazon S3](#)
- [Segurança da infraestrutura no Amazon S3](#)
- [Análise de configuração e vulnerabilidade no Amazon S3](#)
- [Práticas recomendadas de segurança para o Amazon S3](#)
- [Monitorar a segurança dos dados com serviços de segurança gerenciados da AWS](#)

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:

- Respaldado pelo [Acordo de Nível de Serviço do Amazon S3](#).
- Projetado 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.

Além do Versionamento do S3, você também pode usar o Bloqueio de Objeto do Amazon S3 e a Replicação do S3 para proteger seus dados. Para obter mais informações, consulte [Tutorial: Proteger dados no Amazon S3 contra exclusão acidental ou bugs de aplicações usando o Versionamento do S3, o Bloqueio de Objeto do S3 e a Replicação do S3](#).

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 comandos 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](#)

Proteger dados com criptografia

Important

O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).

A proteção de dados refere-se à proteção enquanto eles estão 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 do lado do cliente. Para proteger dados em repouso no Amazon S3, você tem as seguintes opções:

- Criptografia do lado do servidor: o Amazon S3 criptografa os objetos antes de salvá-los em discos em seus datacenters da AWS e descriptografa-os quando você os baixa.

Todos os buckets do Amazon S3 têm criptografia configurada por padrão e todos os novos objetos que são carregados em um bucket do S3 são automaticamente criptografados em repouso. A criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) é a configuração de criptografia padrão para todos os buckets no Amazon S3. Para usar um tipo diferente de criptografia, você pode especificar a criptografia do lado do servidor a ser usada nas solicitações PUT do S3 ou definir a configuração de criptografia padrão no bucket de destino.

Se quiser especificar um tipo de criptografia diferente nas solicitações PUT, você pode usar a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS), criptografia de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS) ou criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C). Se quiser definir uma configuração de criptografia padrão diferente no bucket de destino, você pode usar SSE-KMS ou DSSE-KMS.

Para obter mais informações sobre cada opção de criptografia do lado do servidor, consulte [Proteger os dados usando criptografia do lado do servidor](#).

Para configurar a criptografia do lado do servidor, consulte:

- [Especificar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#)
 - [Especificando criptografia no lado do servidor com o AWS KMS \(SSE-KMS\)](#)
 - [the section called “Especificar a DSSE-KMS”](#)
 - [Especificação de criptografia no lado do servidor com chaves fornecidas pelo cliente \(SSE-C\)](#).
- Criptografia do lado do cliente: você criptografa dados do lado do cliente e faz 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 do lado do cliente](#).

Para ver qual porcentagem dos seus bytes de armazenamento estão criptografados, você pode usar as métricas da Lente de Armazenamento do Amazon S3. A Lente de Armazenamento do S3 é um recurso de análise de armazenamento em nuvem que você pode usar para obter visibilidade em toda a organização sobre o uso e a atividade do armazenamento de objetos. Para obter mais informações, consulte [Avaliar a atividade e o uso do armazenamento com o S3 Storage Lens](#). Para

obter uma lista completa de métricas, consulte o [Glossário de métricas da Lente de Armazenamento do S3](#).

Para obter mais informações sobre criptografia do lado do servidor e criptografia do lado do cliente, consulte os tópicos a seguir.

Tópicos

- [Proteger os dados usando criptografia do lado do servidor](#)
- [Proteger dados usando a criptografia do lado do cliente](#)

Proteger os dados usando criptografia do lado do servidor

Important


O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).

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 da AWS 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 URL pré-assinado, esse URL funcionará da mesma forma para objetos criptografados e não criptografados. Além disso, quando você lista objetos no bucket, as operações de API de lista retornam uma lista de todos os objetos, independentemente de estarem ou não criptografados.

Todos os buckets do Amazon S3 têm criptografia configurada por padrão e todos os novos objetos que são carregados em um bucket do S3 são automaticamente criptografados em

repouso. A criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) é a configuração de criptografia padrão para todos os buckets no Amazon S3. Para usar um tipo diferente de criptografia, você pode especificar a criptografia do lado do servidor a ser usada nas solicitações PUT do S3 ou definir a configuração de criptografia padrão no bucket de destino.

Se quiser especificar um tipo de criptografia diferente nas solicitações PUT, você pode usar a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS), criptografia de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS) ou criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C). Se quiser definir uma configuração de criptografia padrão diferente no bucket de destino, você pode usar SSE-KMS ou DSSE-KMS.

 Note

Não é possível aplicar diferentes tipos de criptografia de servidor ao mesmo objeto simultaneamente.

Se você precisar criptografar os objetos existentes, use as operações em lote do S3 e o inventário do S3. Para obter mais informações, consulte [Criptografar objetos com as operações em lote do Amazon S3](#) e [Executar operações em lote de grande escala em objetos do Amazon S3](#).

Há três opções mutuamente excludentes de criptografia do lado do servidor, dependendo de como você opta por gerenciar as chaves de criptografia e do número de camadas de criptografia que deseja aplicar.

Criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3)

Todos os buckets do Amazon S3 têm criptografia configurada por padrão. A opção padrão para a criptografia do lado do servidor é com as chaves gerenciadas pelo Amazon S3 (SSE-S3). Cada objeto é criptografado com uma chave exclusiva. Como uma proteção adicional, a SSE-S3 criptografa a própria chave com uma chave-raiz que alterna regularmente. A SSE-S3 usa uma das cifras de bloco mais fortes disponíveis, o padrão de criptografia avançada de 256 bits (AES-256), para criptografar os dados. Para ter mais informações, consulte [Usar a criptografia do 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)

A criptografia do lado do servidor com o AWS KMS keys (SSE-KMS) é fornecida por meio de uma integração do serviço AWS KMS com o Amazon S3. Com o AWS KMS, você tem mais controle sobre suas chaves. Por exemplo, você pode visualizar chaves separadas, editar políticas de controle e seguir as chaves no AWS CloudTrail. 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 ter mais informações, consulte [Usar criptografia do lado do servidor com o AWS KMS \(SSE-KMS\)](#).

Criptografia de camada dupla do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (DSSE-KMS)

A criptografia de camada dupla do lado do servidor com AWS KMS keys (DSSE-KMS) é semelhante à SSE-KMS, mas a DSSE-KMS aplica duas camadas individuais de criptografia em nível de objeto em vez apenas uma. Como as duas camadas de criptografia são aplicadas a um objeto do lado do servidor, você pode usar uma grande variedade de ferramentas e Serviços da AWS para analisar dados no S3 enquanto usa um método de criptografia capaz de atender aos seus requisitos de conformidade. Para ter mais informações, consulte [Usar criptografia de camada dupla do lado do servidor com chaves do AWS KMS \(DSSE-KMS\)](#).

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, ao gravar em discos, e a descriptografia, quando você acessa os objetos. Para ter mais informações, consulte [Como usar criptografia do lado do servidor com chaves fornecidas pelo cliente \(SSE-C\)](#).

Agora o Amazon S3 criptografa automaticamente todos os objetos novos

O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. A SSE-S3, que usa criptografia AES-256 (Advanced Encryption Standard de 256 bits), é aplicada automaticamente a todos os novos buckets e a todos os buckets do S3 existentes que ainda não tenham a criptografia padrão configurada. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface (AWS CLI) e nos AWS SDKs.

As seções a seguir respondem a perguntas sobre essa atualização.

O Amazon S3 altera as configurações de criptografia padrão para meus buckets existentes que já têm a criptografia padrão configurada?

Não. Não há nenhuma alteração na configuração de criptografia padrão para um bucket existente que já tenha a criptografia SSE-S3 ou a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS) configuradas. Para obter mais informações sobre como definir o comportamento de criptografia padrão para buckets, consulte [Definir o comportamento padrão da criptografia para os buckets do Amazon S3](#). Para obter mais informações sobre configurações de criptografia SSE-S3 e SSE-KMS, consulte [Proteger os dados usando criptografia do lado do servidor](#).

A criptografia padrão é ativada em meus buckets existentes que não têm a criptografia padrão configurada?

Sim. O Amazon S3 agora configura a criptografia padrão em todos os buckets não criptografados para aplicar a criptografia do lado do servidor com chaves gerenciadas do S3 (SSE-S3) como o nível básico de criptografia para novos objetos carregados nesses buckets. Objetos que já estão em um bucket não criptografado existente não serão criptografados automaticamente.

Como posso ver o status de criptografia padrão de novos uploads de objetos?

Atualmente, é possível visualizar o status de criptografia padrão de novos uploads de objetos em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface (AWS CLI) e em AWS SDKs.

- Para ver seus eventos do CloudTrail, consulte [Visualizar eventos do CloudTrail no console do CloudTrail](#) no Guia do usuário do AWS CloudTrail. Os logs do CloudTrail fornecem monitoramento de API para solicitações PUT e POST para o Amazon S3. Quando a criptografia padrão está sendo usada para criptografar objetos nos buckets, os logs do CloudTrail para solicitações de API PUT e POST incluem o seguinte campo como o par de nome-valor: "SSEApplied": "Default_SSE_S3".
- Para visualizar o status de criptografia automática de novos uploads de objetos no Inventário do S3, configure um relatório do Inventário do S3 para incluir o campo de metadados Encryption (Criptografia), depois veja o status de criptografia de cada novo objeto no relatório. Para obter mais informações, consulte [Configurar o Inventário do Amazon S3](#).
- Para ver o status de criptografia automática de novos uploads de objetos na Lente de Armazenamento do S3, configure um painel da Lente de Armazenamento do S3 e veja as métricas Encrypted bytes (Bytes criptografados) Encrypted object count (Contagem de objetos

criptografados) na categoria Data protection (Proteção de dados) do painel. Para obter mais informações, consulte [Criação de um painel do Amazon S3 Storage Lens](#) e [Exibição das métricas da lente de armazenamento do S3 nos painéis](#).

- Para ver o status da criptografia automática no nível do bucket no console do Amazon S3, verifique a criptografia padrão dos buckets do Amazon S3 no console do Amazon S3. Para ter mais informações, consulte [Configurar a criptografia padrão](#).
- Para ver o status de criptografia automática como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface (AWS CLI) e em AWS SDKs, verifique o cabeçalho de resposta `x-amz-server-side-encryption` ao usar APIs de ação do objeto, como [PutObject](#) e [GetObject](#).

O que preciso fazer para aproveitar essa mudança?

Você não precisa fazer nenhuma alteração em suas aplicações existentes. Como a criptografia padrão está habilitada para todos os seus buckets, todos os novos objetos enviados para o Amazon S3 serão criptografados automaticamente.

Posso desativar a criptografia para os novos objetos que serão gravados no meu bucket?

Não. A SSE-S3 é o novo nível básico de criptografia aplicado a todos os novos objetos que são enviados para o bucket. Não é mais possível desativar a criptografia para novos uploads de objetos.

Minhas cobranças serão afetadas?

Não. A criptografia padrão com SSE-S3 está disponível sem nenhum custo adicional. Serão cobrados o armazenamento, as solicitações e outros recursos do S3, como de costume. Para obter informações sobre preços, consulte [Definição de preços do Amazon S3](#).

O Amazon S3 vai criptografar meus objetos existentes que não estão criptografados?

Não. A partir de 5 de janeiro de 2023, o Amazon S3 criptografa automaticamente somente novos uploads de objetos. Para criptografar objetos existentes, você pode usar o recurso Operações em Lote do S3 para criar cópias criptografadas de seus objetos. Essas cópias criptografadas reterão os dados e o nome do objeto existente e serão criptografadas usando as chaves de criptografia que você especificar. Para obter mais detalhes, consulte [Criptografia de objetos com o recurso Operações em Lote do Amazon S3](#) no blog de armazenamento da AWS.

Eu não habilitei a criptografia para meus buckets antes desse lançamento. Preciso mudar a forma como acesso os objetos?

Não. A criptografia padrão com SSE-S3 criptografa automaticamente seus dados à medida que são gravados no Amazon S3 e os descriptografa para você quando os acessa. Não há nenhuma alteração na forma como você acessa objetos que são criptografados automaticamente.

Preciso mudar a forma como acesso meus objetos com criptografia do lado do cliente?

Não. Todos os objetos criptografados do lado do cliente antes de serem carregados no Amazon S3 chegam como objetos de texto cifrado criptografados no Amazon S3. Esses objetos agora terão uma camada adicional de criptografia SSE-S3. Suas workloads que usam objetos criptografados do lado do cliente não exigirão nenhuma alteração nos serviços do cliente ou nas configurações de autorização.

Note

Os usuários da Terraform da HashiCorp que não usam uma versão atualizada do AWS Provider talvez vejam uma mudança inesperada depois de criar buckets do S3 sem nenhuma configuração de criptografia definida pelo cliente. Para evitar esse desvio, atualize sua versão do AWS Provider para Terraform para uma das seguintes versões: qualquer versão 4.x, 3.76.1 ou 2.70.4.

Usar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3)

Important

O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).

Todos os uploads de novos objetos para buckets do Amazon S3 são criptografados por padrão com criptografia do lado do servidor com chaves 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 do lado do servidor do Amazon S3 usa o Advanced Encryption Standard Galois/Counter Mode (AES-GCM) de 256 bits para criptografar todos os objetos carregados.

Não existem taxas adicionais pelo uso da criptografia do 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 preços, consulte [Preços do Amazon S3](#).

Se você precisar que os uploads de dados sejam criptografados usando somente chaves gerenciadas pelo Amazon S3, poderá usar a política de bucket a seguir. 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": "DenyObjectsThatAreNotSSES3",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "AES256"
        }
      }
    }
  ]
}
```

Note

A criptografia no lado do servidor criptografa somente os dados de objeto, não os metadados de objeto.

Suporte de API para criptografia do lado do servidor


Todos os buckets do Amazon S3 têm criptografia configurada por padrão e todos os novos objetos que são carregados em um bucket do S3 são automaticamente criptografados em repouso. A criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) é a configuração de criptografia padrão para todos os buckets no Amazon S3. Para usar um tipo diferente de criptografia, você pode especificar a criptografia do lado do servidor a ser usada nas solicitações PUT do S3 ou definir a configuração de criptografia padrão no bucket de destino.

Se quiser especificar um tipo de criptografia diferente nas solicitações PUT, você pode usar a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS), criptografia de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS) ou criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C). Se quiser definir uma configuração de criptografia padrão diferente no bucket de destino, você pode usar SSE-KMS ou DSSE-KMS.

Para configurar a criptografia do lado do servidor usando as APIs REST de criação de objetos, é necessário fornecer o cabeçalho de solicitação `x-amz-server-side-encryption`. Para obter informações sobre as APIs REST, consulte [Uso da API REST](#).

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 upload de várias partes: especifique o cabeçalho na solicitação de inicialização ao fazer upload de objetos grandes usando a operação de API de upload de várias partes. 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.

Para obter mais informações gerais, consulte [Conceitos do AWS KMS](#) no Guia do desenvolvedor do AWS Key Management Service.

Tópicos

- [Especificar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#)

Especificar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3)

 Important

O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).

Todos os buckets do Amazon S3 têm criptografia configurada por padrão e todos os novos objetos que são carregados em um bucket do S3 são automaticamente criptografados em repouso. A criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) é a configuração de criptografia padrão para todos os buckets no Amazon S3. Para usar um tipo

diferente de criptografia, você pode especificar a criptografia do lado do servidor a ser usada nas solicitações PUT do S3 ou definir a configuração de criptografia padrão no bucket de destino.

Se quiser especificar um tipo de criptografia diferente nas solicitações PUT, você pode usar a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS), criptografia de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS) ou criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C). Se quiser definir uma configuração de criptografia padrão diferente no bucket de destino, você pode usar SSE-KMS ou DSSE-KMS.

Você pode especificar o SSE-S3 usando o console do S3, APIs REST, AWS SDKs e a AWS Command Line Interface (AWS CLI). Para ter mais informações, consulte [Definir o comportamento padrão da criptografia para os buckets do Amazon S3](#).

Uso do console do S3

Este tópico descreve como definir ou alterar o tipo de criptografia de um objeto usando o AWS Management Console. Ao copiar um objeto usando o console, o Amazon S3 copia o objeto como está. Isso significa que se o objeto de origem é criptografado, o objeto de destino também será criptografado. É possível usar o console para adicionar ou alterar a criptografia para 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. O perfil que altera a propriedade também se torna o proprietário do novo objeto (ou versão do objeto).

Para 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. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, escolha o nome do bucket que contém o objeto.
4. Na lista Name (Nome), escolha o nome do objeto ao qual você deseja adicionar ou no qual deseja alterar a criptografia.

A página de detalhes do objeto é exibida com várias seções que exibem as propriedades do objeto.

5. Escolha a guia Properties (Propriedades).
6. Role para baixo até a seção Configurações da criptografia do lado do servidor e escolha Editar.
7. Em Configurações da criptografia, escolha Usar configurações de criptografia padrão do bucket ou Substituir configurações de criptografia padrão do bucket.
8. Se você escolher Substituir configurações do bucket para criptografia padrão, defina as configurações de criptografia a seguir.
 - Em Tipo de criptografia, escolha Chaves gerenciadas pelo Amazon S3 (SSE-S3). A SSE-S3 usa uma das criptografias de bloco mais fortes: o padrão de criptografia avançada de 256 bits (AES-256) para criptografar cada objeto. Para ter mais informações, consulte [Usar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#).
9. 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 da API REST

No momento da criação do objeto (quando você faz upload de um objeto novo ou faz uma cópia de um objeto existente), é possível especificar se deseja que o Amazon S3 criptografe seus dados com as chaves gerenciadas pelo Amazon S3 (SSE-S3) adicionando o cabeçalho `x-amz-server-side-encryption` à solicitação. Defina o valor do cabeçalho como o algoritmo de criptografia AES256, que é compatível com o Amazon S3. O Amazon S3 confirma que o objeto foi armazenado com SSE-S3 retornando o cabeçalho de resposta `x-amz-server-side-encryption`.

As seguintes operações de 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 operação de API de upload de várias partes, você pode especificar a criptografia do lado do servidor adicionando o cabeçalho `x-amz-server-side-encryption` à solicitação Iniciar upload de várias partes. 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 do lado do servidor.

Os cabeçalhos de resposta das operações de API REST a seguir retornam o cabeçalho `x-amz-server-side-encryption` quando um objeto é armazenado usando SSE-S3.

- [Objeto PUT](#)
- [Objeto PUT - Copiar](#)
- [Objeto POST](#)
- [Iniciar multipart upload](#)
- [Carregar parte](#)
- [Carregamento de parte: copiar](#)
- [Concluir carregamento fracionado](#)
- [Objeto Get](#)
- [Objeto Head](#)

 Note

Não envie cabeçalhos de solicitação de criptografia para solicitações GET e HEAD se o objeto usa SSE-S3, ou você receberá um erro 400 (Solicitação inválida) de status HTTP.

Uso dos AWS SDKs

Ao usar AWS SDKs, é possível solicitar que o Amazon S3 use criptografia do lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 (SSE-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 fazer upload de um objeto, você pode usar SSE-S3 para criptografá-lo. 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 SSE-S3 ao fazer upload de objetos com a operação de API de upload de várias partes:

- Ao usar a operação de API de alto nível de upload de várias partes, 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 ter mais informações, consulte [Fazer upload de um objeto usando multipart upload](#).
- Ao usar a operação de API de upload de várias partes de baixo nível, você especifica a criptografia de servidor ao iniciar o upload de várias partes. Você adiciona a propriedade `ObjectMetadata` chamando o método `InitiateMultipartUploadRequest.setObjectMetadata()`. Para obter mais informações, consulte [Usar os AWS SDKs \(API de baixo nível\)](#).

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 do lado do servidor usando o AWS SDK for Java. Ele mostra como executar as seguintes tarefas:

- Faça upload de um novo objeto usando SSE-S3.
- 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 da API REST](#). Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3](#).

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

import java.io.ByteArrayInputStream;

public class SpecifyServerSideEncryption {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "**** Bucket name ****";
        String keyNameToEncrypt = "**** Key name for an object to upload and encrypt
****";
        String keyNameToCopyAndEncrypt = "**** Key name for an unencrypted object to
be encrypted by copying ****";
        String copiedObjectKeyName = "**** Key name for the encrypted copy of the
unencrypted object ****";

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

            // Upload an object and encrypt it with SSE.
            uploadObjectWithSSEEncryption(s3Client, bucketName, keyNameToEncrypt);

            // Upload a new unencrypted object, then change its encryption state
            // to encrypted by making a copy.
            changeSSEEncryptionStatusByCopying(s3Client,
```



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

private static void uploadObjectWithSSEEncryption(AmazonS3 s3Client, String
bucketName, String keyName) {
    String objectContent = "Test object encrypted with SSE";
    byte[] objectBytes = objectContent.getBytes();

    // Specify server-side encryption.
    ObjectMetadata objectMetadata = new ObjectMetadata();
    objectMetadata.setContentLength(objectBytes.length);

    objectMetadata.setSSEAlgorithm(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);
    PutObjectRequest putRequest = new PutObjectRequest(bucketName,
        keyName,
        new ByteArrayInputStream(objectBytes),
        objectMetadata);

    // Upload the object and check its encryption status.
    PutObjectResult putResult = s3Client.putObject(putRequest);
    System.out.println("Object \"" + keyName + "\" uploaded with SSE.");
    printEncryptionStatus(putResult);
}

private static void changeSSEEncryptionStatusByCopying(AmazonS3 s3Client,
    String bucketName,
    String sourceKey,
    String destKey) {
    // Upload a new, unencrypted object.
    PutObjectResult putResult = s3Client.putObject(bucketName, sourceKey,
"Object example to encrypt by copying");
    System.out.println("Unencrypted object \"" + sourceKey + "\" uploaded.");
    printEncryptionStatus(putResult);
}
```

```
// Make a copy of the object and use server-side encryption when storing the
// copy.
CopyObjectRequest request = new CopyObjectRequest(bucketName,
    sourceKey,
    bucketName,
    destKey);
ObjectMetadata objectMetadata = new ObjectMetadata();

objectMetadata.setSSEAlgorithm(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);
request.setNewObjectMetadata(objectMetadata);

// Perform the copy operation and display the copy's encryption status.
CopyObjectResult response = s3Client.copyObject(request);
System.out.println("Object \"" + destKey + "\" uploaded with SSE.");
printEncryptionStatus(response);

// Delete the original, unencrypted object, leaving only the encrypted copy
in
// Amazon S3.
s3Client.deleteObject(bucketName, sourceKey);
System.out.println("Unencrypted object \"" + sourceKey + "\" deleted.");
}

private static void printEncryptionStatus(SSEResultBase response) {
    String encryptionStatus = response.getSSEAlgorithm();
    if (encryptionStatus == null) {
        encryptionStatus = "Not encrypted with SSE";
    }
    System.out.println("Object encryption status is: " + encryptionStatus);
}
}
```

.NET

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 SSE-S3 no `CopyObjectRequest`, adicione o seguinte:

```
ServerSideEncryptionMethod = ServerSideEncryptionMethod.AES256
```

Para obter um exemplo funcional de como copiar um objeto, consulte [Uso dos AWS SDKs](#).

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](#).

```
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 SSE-S3 a objetos que você carrega no Amazon S3. Pressupõe-se que você já esteja seguindo as instruções para [Usar o AWS SDK for PHP e executar exemplos do PHP](#) 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 da API REST](#).

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

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

// $filepath should be an absolute path to a file on disk.
$filepath = '*** Your File Path ***';

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

// Upload a file with server-side encryption.
$result = $s3->putObject([
    'Bucket' => $bucket,
    'Key' => $keyname,
    'SourceFile' => $filepath,
    'ServerSideEncryption' => 'AES256',
]);
```

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 de objeto.

Ao fazer upload de objetos grandes usando a operação de API de upload de várias partes, você pode especificar a SSE-S3 para os objetos que estiver carregando, conforme segue:

- Ao usar a operação de API de upload de várias partes de baixo nível, especifique a criptografia do lado do 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 operação de API de upload de várias partes de baixo nível, consulte [Usar os AWS SDKs \(API de baixo nível\)](#).
- Ao usar a operação de API de upload de várias partes de alto nível, especifique a criptografia do lado do servidor usando o parâmetro `ServerSideEncryption` da operação de API

[CreateMultipartUpload](#). Para ver um exemplo de uso do método `setOption()` com a operação de API de upload de várias partes de alto nível, consulte [Fazer upload de um objeto usando multipart upload](#).

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 criptografará o destino, a menos que você solicite explicitamente a criptografia do lado do 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',
]);
```

Para obter mais informações, consulte os tópicos a seguir.

- [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 SSE-S3. Ao ler o objeto de volta, ele é descriptografado automaticamente.

O exemplo do 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
end
```

```

end

def put_object_encrypted(object_content, encryption)
  @object.put(body: object_content, server_side_encryption: encryption)
  true
rescue Aws::Errors::ServiceError => e
  puts "Couldn't put your content to #{object.key}. Here's why: #{e.message}"
  false
end
end

# Example usage:
def run_demo
  bucket_name = "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__

```

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

# Example usage:
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 do lado do servidor não for usada para o objeto que é armazenado no Amazon S3, o método retornará `null`.

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 com SSE-S3.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyEncryptWrapper
  attr_reader :source_object
```

```
# @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is
used as the source object for
#
#           copy actions.
def initialize(source_object)
  @source_object = source_object
end

# Copy the source object to the specified target bucket, rename it with the target
key, and encrypt it.
#
# @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
object is copied.
# @param target_object_key [String] The key to give the copy of the object.
# @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
nil.
def copy_object(target_bucket, target_object_key, encryption)
  @source_object.copy_to(bucket: target_bucket.name, key: target_object_key,
server_side_encryption: encryption)
  target_bucket.object(target_object_key)
rescue Aws::Errors::ServiceError => e
  puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
#{e.message}"
end
end

# Example usage:
def run_demo
  source_bucket_name = "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__
```

Usando a AWS CLI

Para especificar a 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 a 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 os exemplos [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 tópico `AWS::S3::Bucket` `ServerSideEncryptionRule` do Guia do usuário do AWS CloudFormation.

Usar criptografia do lado do servidor com o AWS KMS (SSE-KMS)

Important

O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).

A criptografia do lado do servidor é a criptografia de dados em seu destino pela aplicação ou serviço que os recebe.

O Amazon S3 ativa automaticamente a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) para uploads de novos objetos.

A menos que você especifique o contrário, os buckets usam SSE-S3 por padrão para criptografar objetos. No entanto, você pode optar por configurar buckets para usar a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS) em vez disso. Para ter mais informações, consulte [Especificando criptografia no lado do servidor com o AWS KMS \(SSE-KMS\)](#).

O AWS KMS é um serviço que combina hardware e software seguros e altamente disponíveis para fornecer um sistema de gerenciamento de chaves escalado para a nuvem. O Amazon S3 usa criptografia do lado do servidor com AWS KMS (SSE-KMS) para criptografar os dados do objeto do S3. Além disso, quando a SSE-KMS é solicitada para o objeto, a soma de verificação do S3 (como parte dos metadados do objeto) é armazenada em formato criptografado. Para obter mais informações sobre a soma de verificação, consulte [Verificar a integridade do objeto](#).

Se você usar chaves do KMS, poderá usar o AWS KMS por meio do [AWS Management Console](#) ou da [API do AWS KMS](#) para fazer o seguinte:

- Criar, visualizar, editar, monitorar, ativar/desativar, rotacionar e programar a exclusão de chaves KMS, tudo de forma centralizada.
- Definir políticas que controlam como e por quem as chaves do KMS podem ser usadas.
- Auditar o uso para provar que elas estão sendo usadas corretamente. A auditoria é compatível com a [API do AWS KMS](#), mas não com o [AWS Management Console do AWS KMS](#).

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 uploads de várias partes, consulte [API de multipart upload e permissões](#).

Important

Analise cuidadosamente as permissões concedidas em suas políticas de chave do KMS. Restrinja sempre as permissões de política de chave do KMS gerenciada pelo cliente somente às entidades principais do IAM e a serviços da AWS que precisam acessar a ação de chave do AWS KMS pertinente. Para obter mais informações, consulte [Usar políticas de chaves no AWS KMS](#).

Tópicos

- [AWS KMS keys](#)
- [Amazon S3 Bucket Keys](#)
- [Exigir a criptografia no lado do servidor](#)
- [Contexto de criptografia](#)
- [Enviar solicitações para objetos criptografados do AWS KMS](#)
- [Especificando criptografia no lado do servidor com o AWS KMS \(SSE-KMS\)](#)
- [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3](#)

AWS KMS keys

Ao usar criptografia do lado do 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 é compatível com criptografia envelopada. O S3 usa os recursos do AWS KMS de criptografia envelopada para proteger ainda mais seus dados. Criptografia envelopada é a prática de criptografar dados de texto simples com uma chave de dados, depois criptografar essa chave de dados com uma chave do KMS. Para ter mais informações sobre a criptografia de envelope, consulte [Criptografia de envelope](#) no Guia do desenvolvedor do AWS Key Management Service.

Se você não especificar uma chave gerenciada pelo cliente, o Amazon S3 criará automaticamente uma Chave gerenciada pela AWS 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.

Note

Objetos criptografados usando SSE-KMS com [Chaves gerenciadas pela AWS](#) não podem ser compartilhados entre contas. Se você precisar replicar dados de SSE-KMS entre contas, será necessário usar uma [chave gerenciada pelo cliente](#) do AWS KMS.

Se você quiser usar uma chave gerenciada pelo cliente para SSE-KMS, crie uma chave de criptografia simétrica gerenciada pelo cliente antes de configurar a SSE-KMS. Depois, ao configurar o SSE-KMS para seu bucket, especifique a chave gerenciada pelo cliente existente. Para obter mais informações sobre chaves de criptografia simétrica, consulte [Chaves do KMS de criptografia simétrica](#), no Guia do desenvolvedor do AWS Key Management Service.

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 pela AWS, consulte [Chaves de clientes e chaves da AWS](#) no Guia do desenvolvedor do AWS Key Management Service.

Note

Ao usar a criptografia do lado do servidor com uma chave gerenciada pelo cliente que é armazenada em um armazenamento de chaves externo, ao contrário das chaves do KMS padrão, você é responsável por garantir a disponibilidade e a durabilidade do seu material de chaves. Para obter mais informações sobre armazenamentos de chaves externos e como eles mudam o modelo de responsabilidade compartilhada, consulte [Armazenamentos de chaves externos](#) no Guia do desenvolvedor do AWS Key Management Service.

Se você optar por criptografar seus dados usando uma Chave gerenciada pela AWS ou uma chave gerenciada pelo cliente, o AWS KMS e o Amazon S3 realizarão as seguintes ações de criptografia envelopada:

1. O Amazon S3 solicitará uma [chave de dados](#) em texto simples e uma cópia da chave criptografada na chave do KMS especificada.
2. 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.

3. 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.
4. 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:

1. O Amazon S3 envia a chave de dados criptografada ao AWS KMS em uma solicitação Decrypt.
2. O AWS KMS descriptografa a chave dos dados criptografados usando a mesma chave do KMS e retorna a chave de dados de texto simples para o Amazon S3.
3. O Amazon S3 descriptografa os dados criptografados usando a chave de dados de texto simples e remove a chave de dados de texto simples da memória o quanto antes.

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. Para obter mais informações sobre essas chaves, consulte [Chaves do KMS de criptografia simétrica](#) no Guia do desenvolvedor do AWS Key Management Service.

Para identificar solicitações que especificam SSE-KMS, você pode usar as métricas All SSE-KMS requests (Todas as solicitações de SSE-KMS) e % all SSE-KMS requests (% de todas as solicitações de SSE-KMS) da Lente de Armazenamento do Amazon S3. A Lente de Armazenamento do S3 é um recurso de análise de armazenamento em nuvem que você pode usar para obter visibilidade em toda a organização sobre o uso e a atividade do armazenamento de objetos. Para obter mais informações, consulte [Avaliar a atividade e o uso do armazenamento com o S3 Storage Lens](#). Para obter uma lista completa de métricas, consulte o [Glossário de métricas da Lente de Armazenamento do S3](#).

Amazon S3 Bucket Keys

Ao configurar a criptografia do lado do servidor usando o AWS KMS (SSE-KMS), você pode configurar os buckets 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 o bucket para usar uma chave 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 do S3 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](#).

Exigir a criptografia no lado do servidor

Para exigir criptografia do lado do servidor de 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 negará permissão de upload de objeto (`s3:PutObject`) para todos se a solicitação não incluir um cabeçalho `x-amz-server-side-encryption-aws-kms-key-id` que solicita criptografia do lado do servidor com a SSE-KMS.

```
{
  "Version": "2012-10-17",
  "Id": "PutObjectPolicy",
  "Statement": [{
    "Sid": "DenyObjectsThatAreNotSSEKMS",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*",
    "Condition": {
      "Null": {
        "s3:x-amz-server-side-encryption-aws-kms-key-id": "true"
      }
    }
  }
]
```

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 do recurso da Amazon (ARN) da chave no formato `arn:aws:kms:region:acct-id:key/key-id`. O AWS Identity and Access

Management não validará se a string para `s3:x-amz-server-side-encryption-aws-kms-key-id` existe.

Note

Quando você faz upload de um objeto, pode especificar a chave do 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.

Consulte uma lista completa de chaves de condição específicas do Amazon S3 em [Condition keys for Amazon S3](#) na Referência de autorização do serviço.

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.

Por padrão, o Amazon S3 usa o nome do recurso da Amazon (ARN) do objeto ou do 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](#).

```
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"
}
```

Enviar solicitações para objetos criptografados do AWS KMS

- ⚠ Important**
Todas as solicitações GET e PUT para objetos criptografados do AWS KMS devem ser criadas usando 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).

O AWS Signature Version 4 é o processo de 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

Se o objeto usar SSE-KMS, não envie cabeçalhos de solicitação de criptografia para solicitações GET e solicitações HEAD. Caso contrário, você receberá um erro HTTP 400 Solicitação Inválida.

Tópicos

- [Especificando criptografia no lado do servidor com o AWS KMS \(SSE-KMS\)](#)
- [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3](#)

Especificando criptografia no lado do servidor com o AWS KMS (SSE-KMS)

⚠ Important

O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).

Todos os buckets do Amazon S3 têm criptografia configurada por padrão e todos os novos objetos que são carregados em um bucket do S3 são automaticamente criptografados em repouso. A criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) é a configuração de criptografia padrão para todos os buckets no Amazon S3. Para usar um tipo diferente de criptografia, você pode especificar a criptografia do lado do servidor a ser usada nas solicitações PUT do S3 ou definir a configuração de criptografia padrão no bucket de destino.

Se quiser especificar um tipo de criptografia diferente nas solicitações PUT, você pode usar a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS), criptografia de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS) ou criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C). Se quiser definir uma configuração de criptografia padrão diferente no bucket de destino, você pode usar SSE-KMS ou DSSE-KMS.

É possível aplicar criptografia durante o upload de um novo objeto ou ao copiar um objeto existente.

Você pode especificar a SSE-KMS usando o console do Amazon S3, operações da API REST, AWS SDKs 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.

Note

Se quiser usar uma chave do KMS que seja de propriedade de outra conta, você deverá ter permissão para usar a chave. Para obter mais informações sobre permissões entre contas para chaves do KMS, consulte [Criar chaves do KMS que outras contas podem usar](#) 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 para usar criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS) 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. O perfil 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. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, escolha o nome do bucket que contém o objeto.
4. Na lista Name (Nome), escolha o nome do objeto ao qual você deseja adicionar ou no qual deseja alterar a criptografia.

A página de detalhes do objeto é exibida com várias seções que exibem as propriedades do objeto.

5. Escolha a guia Properties (Propriedades).
6. Role para baixo até a seção Configurações da criptografia do lado do servidor e escolha Editar.

A página Edit server-side encryption (Editar criptografia do lado do servidor) é aberta.

7. Em Criptografia do lado do servidor, em Configurações de criptografia, escolha Substituir configurações padrão do bucket de criptografia.
8. Em Tipo de criptografia, escolha Criptografia do lado do servidor com chaves do AWS Key Management Service (SSE-KMS).

 Important

Se você usar a opção SSE-KMS em sua configuração de criptografia padrão, estará sujeito às cotas de solicitações por segundo (RPS) do AWS KMS. Para obter mais informações sobre as cotas do AWS KMS e como solicitar um aumento de cota, consulte [Cotas](#) no Guia do desenvolvedor do AWS Key Management Service.

9. Em Chave do AWS KMS, realize um dos seguintes procedimentos para escolher sua chave do KMS:

- Para escolher entre uma lista de chaves do KMS disponíveis, selecione Escolher de sua AWS KMS keys e escolha a chave do KMS na lista de chaves disponíveis.

As chaves Chave gerenciada pela AWS (aws/s3) e as chaves gerenciadas pelo cliente são exibidas nessa lista. Para ter mais informações sobre chaves gerenciadas pelo cliente, consulte [Chaves de clientes e chaves da AWS](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para inserir o ARN da chave do KMS, selecione Inserir ARN da AWS KMS key e insira o ARN da chave do KMS no campo exibido.
- Para criar uma chave gerenciada pelo cliente no console do AWS KMS, selecione Criar uma chave do KMS.

Para ter mais informações sobre como criar uma AWS KMS key, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

Important

Você só pode usar chaves do KMS disponíveis na mesma Região da AWS que o bucket. O console do Amazon S3 lista somente as primeiras 100 chaves do KMS na mesma região que o bucket. Para usar uma chave do KMS que não esteja listada, você deve inserir o ARN da chave do KMS. Se quiser usar uma chave do KMS que seja de propriedade de outra conta, primeiro você deverá ter permissão para usar a chave e, depois, inserir o ARN da chave do KMS.

O Amazon S3 só é compatível com chaves do KMS de criptografia simétrica, e não com chaves assimétricas do KMS. Para ter mais informações, consulte [Identificar chaves do KMS simétricas e assimétricas](#) no Guia do desenvolvedor do AWS Key Management Service.

10. 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 da API REST

Quando você cria um objeto, ou seja, quando você faz upload de um novo objeto ou copia um objeto existente, é possível especificar a utilização de criptografia do lado do servidor com AWS KMS keys (SSE-KMS) para criptografar os 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](#)
- [Contexto de criptografia \(`x-amz-server-side-encryption-context`\)](#)
- [ID da chave do AWS KMS \(`x-amz-server-side-encryption-aws-kms-key-id`\)](#)
- [Chaves de bucket do S3 \(`x-amz-server-side-encryption-aws-bucket-key-enabled`\)](#)

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`.

- [PutObject](#): ao carregar dados usando a operação de API PUT, é possível especificar esses cabeçalhos de solicitação.
- [CopyObject](#): quando você copia um objeto, há um objeto de origem e um objeto de destino. Ao transmitir cabeçalhos de SSE-KMS com a operação CopyObject, 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 do lado do 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.
- [CreateMultipartUpload](#): ao fazer upload de objetos grandes usando a operação de API de upload de várias partes, você pode especificar esses cabeçalhos. Você especifica esses cabeçalhos na solicitação de iniciar o upload de várias partes.

Os cabeçalhos de resposta das operações de API REST a seguir retornam o cabeçalho `x-amz-server-side-encryption` quando um objeto é armazenado usando criptografia de servidor.

- [PutObject](#)
- [CopyObject](#)
- [Objeto POST](#)
- [CreateMultipartUpload](#)
- [UploadPart](#)
- [UploadPartCopy](#)
- [CompleteMultipartUpload](#)
- [GetObject](#)
- [HeadObject](#)

 Important

- Todas as solicitações GET e PUT para um objeto protegido por AWS KMS falharão se elas não forem feitas usando Secure Sockets Layer (SSL), Transport Layer Security (TLS) ou Signature versão 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](#). 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 cabeçalho `x-amz-server-side-encryption:aws:kms`, mas não fornecer o cabeçalho `x-amz-server-side-encryption-aws-kms-key-id`, o Amazon S3 usará a Chave gerenciada pela AWS (`aws/s3`) 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. Para obter mais informações sobre essas chaves, consulte [Chaves do KMS de criptografia simétrica](#) 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 ter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3](#).

Se você especificar o cabeçalho `x-amz-server-side-encryption:aws:kms`, mas não fornecer o cabeçalho `x-amz-server-side-encryption-aws-bucket-key-enabled`, o objeto usará as configurações da chave de bucket do S3 para o bucket de destino para criptografar seu objeto. Para ter mais informações, consulte [Configurar uma chave de bucket do S3 no nível de objeto](#).

Usar a AWS CLI

Ao fazer upload de um novo objeto ou copiar um objeto existente, ou seja, você pode especificar a utilização de criptografia do lado do servidor com chaves do AWS KMS para criptografar seus dados. Para fazer isso, adicione o cabeçalho `--server-side-encryption aws:kms` à solicitação. Use `--ssekms-key-id example-key-id` para adicionar a [chave do AWS KMS gerenciada pelo cliente](#) que você criou. Se você especificar `--server-side-encryption aws:kms`, mas não fornecer um ID de chave do AWS KMS, o Amazon S3 usará uma chave gerenciada pela AWS.

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key example-object-key --server-side-encryption aws:kms --ssekms-key-id example-key-id --body filepath
```

Além disso, você pode ativar ou desativar as chaves de bucket do Amazon S3 nas operações PUT ou COPY adicionando `--bucket-key-enabled` ou `--no-bucket-key-enabled`. As chaves de bucket do Amazon 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 [Reduzir o custo da SSE-KMS usando chaves de bucket do Amazon S3](#).

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key example-object-key --server-side-encryption aws:kms --bucket-key-enabled --body filepath
```

Você pode criptografar um objeto não criptografado para usar a SSE-KMS copiando o objeto de volta no lugar.

```
aws s3api copy-object --bucket DOC-EXAMPLE-BUCKET --key example-object-key --  
body filepath --bucket DOC-EXAMPLE-BUCKET --key example-object-key --sse aws:kms --sse-  
kms-key-id example-key-id --body filepath
```

Uso dos AWS SDKs

Ao usar AWS SDKs, é possível solicitar que o Amazon S3 use AWS KMS keys para criptografia do lado do servidor. 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. Para obter mais informações sobre essas chaves, consulte [Chaves do KMS de criptografia simétrica](#) 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 [Copiar e mover objetos](#).

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 (`aws/s3`) (consulte [Usar criptografia do lado do servidor com o AWS KMS \(SSE-KMS\)](#)). 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. Para usar esses exemplos, é necessário atualizar os 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](#).
- 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](#).
 - Para usar a API de carregamento fracionado de baixo nível, consulte [Usar os AWS SDKs \(API de baixo nível\)](#).

.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 = DOC-EXAMPLE-BUCKET,  
    Key = keyName,  
    // other properties.  
    ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS  
};
```

Nesse caso, o Amazon S3 usa a Chave gerenciada pela AWS. Para ter mais informações, consulte [Usar criptografia do lado do servidor com o AWS KMS \(SSE-KMS\)](#). É possível criar sua própria chave gerenciada pelo cliente de criptografia simétrica e especificar isso na solicitação.

```
PutObjectRequest putRequest1 = new PutObjectRequest
```

```
{
  BucketName = DOC-EXAMPLE-BUCKET,
  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. Para usar esses exemplos, é necessário atualizar os 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](#).
- 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](#).
 - Para usar a API de carregamento fracionado de baixo nível, consulte [Fazer upload de um objeto usando multipart upload](#).

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 com URLs pré-assinados](#).

.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 com URLs pré-assinados](#).

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 do lado do servidor do Amazon S3 com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS). Usar uma chave no 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. Com alguns cliques no AWS Management Console e sem alterações em aplicações do cliente, você pode configurar o bucket para usar uma chave de bucket do S3 para criptografia SSE-KMS em novos objetos.

Note

As chaves de bucket do S3 não comportam criptografia de camada dupla do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (DSSE-KMS).

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. Nesse caso, o Amazon S3 faz uma chamada para o AWS KMS sempre que uma solicitação é feita contra um objeto criptografado pelo KMS. Para obter informações sobre como o SSE-KMS funciona, consulte [Usar criptografia do lado do servidor com o AWS KMS \(SSE-KMS\)](#).

Quando você configura um bucket para usar uma chave de bucket do S3 para SSE-KMS, a AWS gera uma chave de bucket de curta duração do AWS KMS e a mantém temporariamente no S3. Essa chave de bucket criará chaves de dados para novos objetos durante seu ciclo de vida. As chaves de bucket do S3 são usadas por um período limitado no Amazon S3, reduzindo a necessidade de

o 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 pelo AWS KMS no Amazon S3 por uma fração do custo anterior.

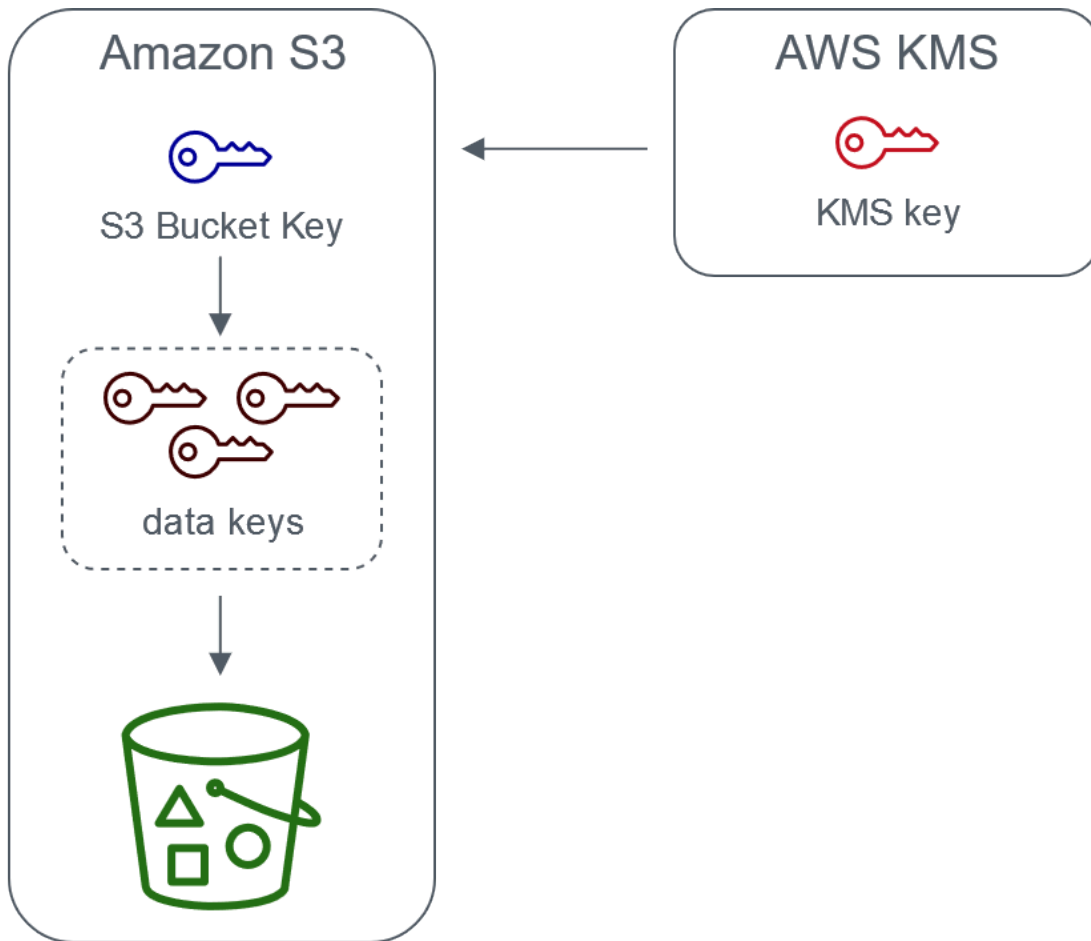
Chaves exclusivas em nível de bucket são buscadas pelo menos uma vez por solicitante para garantir que o acesso do solicitante à chave seja capturado em um evento do CloudTrail no AWS KMS. O Amazon S3 trata os chamadores como solicitantes diferentes quando usam perfis ou contas diferentes, ou o mesmo perfil com políticas de escopo diferentes. A economia de solicitações do AWS KMS reflete o número de solicitantes, os padrões de solicitação e a idade relativa dos objetos solicitados. Por exemplo, um número menor de solicitantes, solicitando vários objetos em um período limitado e criptografados com a mesma chave de bucket, resultará em maior economia.

Note

O uso de chaves de bucket do S3 permite que você reduza os custos de solicitações do AWS KMS ao diminuir as solicitações ao AWS KMS para operações `Encrypt`, `GenerateDataKey` e `Decrypt` por meio do uso de uma chave em nível de bucket. Por definição, as solicitações subsequentes que aproveitam essa chave em nível de bucket não resultam em solicitações de API do AWS KMS nem validam o acesso em relação à política de chave AWS KMS.

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 `CopyObject`. Para ter mais informações, consulte [Configurar uma chave de bucket do S3 no nível de objeto](#).

O Amazon S3 compartilhará apenas uma chave de bucket do S3 para objetos criptografados pela mesma AWS KMS key. As chaves de bucket do S3 são compatíveis com chaves do KMS criadas pelo AWS KMS, [material de chave importado](#) e [material chave baseado em armazenamentos de chaves personalizados](#).



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. Com as chaves de bucket do S3 habilitadas em seu bucket, os objetos carregados com outra chave de SSE-KMS especificada usarão suas próprias chaves de bucket do S3. Independentemente da configuração de chave de bucket do S3, você pode incluir o cabeçalho `x-amz-server-side-encryption-bucket-key-enabled` com um valor `true` ou `false` em sua solicitação para substituir a configuração do bucket.

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](#).

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](#).

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 mais informações, consulte:

- [Configurar uma chave de bucket do S3 no nível de objeto](#)
- [Configurando seu bucket para usar uma chave de bucket do S3 com SSE-KMS para novos objetos](#)

As seguintes operações de API são compatíveis com 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 [POST Object](#)
 - 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 do S3 está ativada ou desativada para um objeto.

Trabalhar com o 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](#).

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 do AWS KMS

Se suas políticas do AWS Identity and Access Management (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](#).

Eventos do CloudTrail para o AWS KMS

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 ter mais informações, consulte [Replicar objetos criados com criptografia do lado do servidor \(SSE-C, SSE-S3, SSE-KMS, DSSE-KMS\)](#).

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 réplica será criptografado usando uma chave de bucket S3 com a 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](#)
- [Configurar uma chave de bucket do S3 no nível de objeto](#)
- [Visualizar as configurações para uma chave de bucket do S3](#)

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 com chaves do AWS Key Management Service (AWS KMS) (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 KMS e reduzem o custo do SSE-KMS. Para ter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3](#).

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, AWS SDKs, a AWS Command Line Interface

(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 `CopyObject`. Para obter mais informações, consulte [Configurar uma chave de bucket do S3 no nível de objeto](#) e [Uso do S3 Batch Operations para criptografar objetos com chaves de bucket do S3](#).

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 a fim de usar o ARN de bucket para o contexto de criptografia. Para ter mais informações, consulte [Chaves de bucket do S3 e replicação](#).

Os exemplos a seguir ilustram como uma chave de bucket do S3 funciona com a replicação. Para ter mais informações, consulte [Replicar objetos criados com criptografia do lado do servidor \(SSE-C, SSE-S3, SSE-KMS, DSSE-KMS\)](#).

Pré-requisitos

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](#).

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 SSE-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

do S3 ativada, mas o bucket de destino tiver a chave de bucket do S3 ativada, uma chave de bucket do 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. No painel de navegação à esquerda, escolha Buckets.
3. Selecione Create bucket (Criar bucket).
4. Insira o nome do bucket e escolha sua Região da AWS.
5. Em Criptografia padrão, para Tipo de chave de criptografia, escolha Chave do AWS Key Management Service (SSE-KMS).
6. Em Chave do AWS KMS, realize um dos seguintes procedimentos para escolher sua chave do KMS:
 - Para escolher entre uma lista de chaves do KMS disponíveis, selecione Escolher de sua AWS KMS keys e escolha a chave do KMS na lista de chaves disponíveis.

As chaves Chave gerenciada pela AWS (aws/s3) e as chaves gerenciadas pelo cliente são exibidas nessa lista. Para ter mais informações sobre chaves gerenciadas pelo cliente, consulte [Chaves de clientes e chaves da AWS](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para inserir o ARN da chave do KMS, escolha Inserir ARN da AWS KMS key e insira o ARN da chave do KMS no campo exibido.
- Para criar uma chave gerenciada pelo cliente no console do AWS KMS, selecione Criar uma chave do KMS.

Para obter mais informações sobre como criar uma AWS KMS key, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

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. Novos objetos que você carregar no bucket usarão uma chave de bucket do S3.

Para desativar uma chave de bucket do S3, siga as etapas anteriores e escolha Disable (Desativar).

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. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets (Buckets), escolha o bucket para o qual você deseja habilitar uma chave de bucket do S3.
4. Escolha a guia Properties (Propriedades).
5. Em Default encryption (Criptografia padrão), escolha Edit (Editar).
6. Em Criptografia padrão, para Tipo de chave de criptografia, escolha Chave do AWS Key Management Service (SSE-KMS).
7. Em Chave do AWS KMS, realize um dos seguintes procedimentos para escolher sua chave do KMS:

- Para escolher entre uma lista de chaves do KMS disponíveis, selecione Escolher de sua AWS KMS keys e escolha a chave do KMS na lista de chaves disponíveis.

As chaves Chave gerenciada pela AWS (aws/s3) e as chaves gerenciadas pelo cliente são exibidas nessa lista. Para ter mais informações sobre chaves gerenciadas pelo cliente, consulte [Chaves de clientes e chaves da AWS](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para inserir o ARN da chave do KMS, escolha Inserir ARN da AWS KMS key e insira o ARN da chave do KMS no campo exibido.
- Para criar uma chave gerenciada pelo cliente no console do AWS KMS, selecione Criar uma chave do KMS.

Para obter mais informações sobre como criar uma AWS KMS key, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

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 configurar uma chave de bucket do S3 para objetos existentes, você pode usar uma operação CopyObject. Para ter mais informações, consulte [Configurar uma chave de bucket do S3 no nível de objeto](#).

Para desativar uma chave de bucket do S3, siga as etapas anteriores e escolha **Disable (Desativar)**.

Uso da API REST

Você pode usar [PutBucketEncryption](#) para habilitar ou desabilitar uma chave de bucket do S3 para o bucket. Para configurar uma chave de bucket do S3 com `PutBucketEncryption`, use o tipo de dados [ServerSideEncryptionRule](#), que inclui criptografia padrão com SSE-KMS. 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);
```

Usando a 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. Substitua os *user input placeholders* por suas próprias informações.

```
aws s3api put-bucket-encryption --bucket DOC-EXAMPLE-BUCKET --server-side-encryption-configuration '{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "aws:kms",
        "KMSEMasterKeyID": "KMS-Key-ARN"
      },
      "BucketKeyEnabled": true
    }
  ]
}'
```

Usar o AWS CloudFormation

Para obter mais informações sobre como configurar uma chave de bucket do S3 com AWS CloudFormation, consulte [AWS::S3::Bucket ServerSideEncryptionRule](#) no Guia do usuário do AWS CloudFormation.

Configurar uma chave de bucket do S3 no nível de objeto

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 para o objeto adicionando o cabeçalho de solicitação `x-amz-server-side-encryption-bucket-key-enabled` com um valor `true` ou `false`. 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](#).

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 enviar uma solicitação PUT ou COPY para um objeto criptografado pelo KMS em um bucket com as chaves de bucket do S3 habilitadas, suas operações no objeto vão usar automaticamente as chaves de bucket do S3, a menos que você

desabilite as chaves no cabeçalho da solicitação. 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](#).

Tópicos

- [Amazon S3 Batch Operations](#)
- [Uso da API REST](#)
- [Usar o AWSSDK Java \(PutObject\)](#)
- [Como usar a AWS CLI \(PutObject\)](#)

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](#) e [Criptografia de objetos existentes com o Amazon S3 Batch Operations](#).

Uso da API REST

Ao usar a SSE-KMS, você pode habilitar uma chave de bucket do S3 para um objeto usando as seguintes operações de API:

- [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.

- [POST Object](#): quando você usa uma operação POST para fazer upload de um objeto e configurar a 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 o objeto.
- [CreateMultipartUpload](#): ao fazer upload de objetos grandes usando a operação de API de `CreateMultipartUpload` e configurar a 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 o 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 da API REST](#).

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 = "DOC-EXAMPLE-BUCKET1";
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 DOC-EXAMPLE-BUCKET --key object key name --server-side-encryption aws:kms --bucket-key-enabled --body filepath
```

Visualizar as 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 Command Line Interface (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 ter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3](#).

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 visualizar a configuração da chave de bucket do S3 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 Buckets (Buckets), escolha o bucket para o qual você deseja habilitar uma chave de bucket do S3.
4. Escolha Properties (Propriedades).
5. 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 Chave de bucket, você vê a configuração da chave de bucket do S3 para o objeto. Você não pode editar essa configuração.

Usando a AWS CLI

Para retornar as configurações da chave de bucket do S3 no nível de bucket

Para usar esse exemplo, substitua cada *user input placeholder* por suas próprias informações.

```
aws s3api get-bucket-encryption --bucket DOC-EXAMPLE-BUCKET1
```

Para obter mais informações, consulte [get-bucket-encryption](#) na Referência de comandos da AWS CLI.

Como retornar configurações de chave de bucket do S3 para objetos

Para usar esse exemplo, substitua cada *user input placeholder* por suas próprias informações.

```
aws s3api head-object --bucket DOC-EXAMPLE-BUCKET1 --key my_images.tar.bz2
```

Para obter mais informações, consulte [head-object](#) na Referência de comandos da AWS CLI.

Uso da API REST

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 elemento `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](#)

Usar criptografia de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS)

O uso da criptografia de camada dupla do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (DSSE-KMS) aplica duas camadas de criptografia aos objetos quando eles são carregados no Amazon S3. A DSSE-KMS ajuda a cumprir mais facilmente os padrões de conformidade que exigem que você aplique a criptografia de várias camadas aos dados e tenha controle total das chaves de criptografia.

Ao usar a criptografia DSSE-KMS com um bucket do Amazon S3, as chaves do AWS KMS devem estar na mesma região que o bucket. Além disso, quando a DSSE-KMS é solicitada para o objeto, a soma de verificação do S3 que faz parte dos metadados do objeto é armazenada em formato criptografado. Para ter mais informações sobre somas de verificação, consulte [Verificar a integridade do objeto](#).

Há custos adicionais para usar a DSSE-KMS e as AWS KMS keys. Para ter mais informações sobre o preço da DSSE-KMS, consulte [Conceitos do AWS KMS key](#) no Guia do desenvolvedor do AWS Key Management Service e [Preços do AWS KMS](#).

Note

As chaves de bucket do S3 não comportam DSSE-KMS.

Exigir criptografia de camada dupla do lado do servidor com chaves do AWS KMS keys (DSSE-KMS)

Para exigir criptografia de camada dupla do lado do servidor de 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 negará permissão de carregamento de objeto (`s3:PutObject`) para todos se a solicitação não incluir um cabeçalho `x-amz-server-side-encryption` que solicita criptografia do lado do servidor com a DSSE-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:dsse"
      }
    }
  ]
}
```

Tópicos

- [Especificar criptografia de camada dupla do lado do servidor com chaves do AWS KMS \(DSSE-KMS\)](#)

Especificar criptografia de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS)

Important

O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).

Todos os buckets do Amazon S3 têm criptografia configurada por padrão e todos os novos objetos que são carregados em um bucket do S3 são automaticamente criptografados em repouso. A criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) é a configuração de criptografia padrão para todos os buckets no Amazon S3. Para usar um tipo diferente de criptografia, você pode especificar a criptografia do lado do servidor a ser usada nas solicitações PUT do S3 ou definir a configuração de criptografia padrão no bucket de destino.

Se quiser especificar um tipo de criptografia diferente nas solicitações PUT, você pode usar a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS), criptografia de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS) ou criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C). Se quiser definir uma configuração de criptografia padrão diferente no bucket de destino, você pode usar SSE-KMS ou DSSE-KMS.

É possível aplicar criptografia durante o upload de um novo objeto ou ao copiar um objeto existente.

Você pode especificar a DSSE-KMS usando o console do Amazon S3, a API REST do Amazon S3 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.

Note

Se quiser usar uma chave do KMS que seja de propriedade de outra conta, você deverá ter permissão para usar a chave. Para obter mais informações sobre permissões entre contas para chaves do KMS, consulte [Criar chaves do KMS que outras contas podem usar](#) no Guia do desenvolvedor do AWS Key Management Service.

Uso do console do S3

Esta seção descreve como definir ou alterar o tipo de criptografia de um objeto para usar criptografia de camada dupla do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (DSSE-KMS) por meio do console do Amazon S3.

Note

Se você alterar o método de criptografia de um objeto, outro 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. O perfil 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. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, escolha o nome do bucket que contém os objetos que você deseja restaurar.
4. Na lista Nome, marque a caixa de seleção ao lado do objeto ao qual você deseja adicionar ou no qual deseja alterar a criptografia.

A página de detalhes do objeto é exibida com várias seções que exibem as propriedades do objeto.

5. Escolha a guia Properties (Propriedades).
6. Role para baixo até a seção Criptografia padrão e escolha Editar.

A página Editar criptografia padrão é aberta.

7. Em Tipo de criptografia, escolha Criptografia de camada dupla do lado do servidor com chaves do AWS Key Management Service (DSSE-KMS).
8. Em Chave do AWS KMS, realize um dos seguintes procedimentos para escolher sua chave do KMS:

- Para escolher entre uma lista de chaves do KMS disponíveis, selecione Escolher de sua AWS KMS keys e escolha a chave do KMS na lista de chaves disponíveis.

As chaves Chave gerenciada pela AWS (aws/s3) e as chaves gerenciadas pelo cliente são exibidas nessa lista. Para ter mais informações sobre chaves gerenciadas pelo cliente, consulte [Chaves de clientes e chaves da AWS](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para inserir o ARN da chave do KMS, selecione Inserir ARN da AWS KMS key e insira o ARN da chave do KMS no campo exibido.

- Para criar uma chave gerenciada pelo cliente no console do AWS KMS, selecione Criar uma chave do KMS.

Para ter mais informações sobre como criar uma AWS KMS key, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

Important

Você só pode usar chaves do KMS disponíveis na mesma Região da AWS que o bucket. O console do Amazon S3 lista somente as primeiras 100 chaves do KMS na mesma região que o bucket. Para usar uma chave do KMS que não esteja listada, você deve inserir o ARN da chave do KMS. Se quiser usar uma chave do KMS que seja de propriedade de outra conta, primeiro você deverá ter permissão para usar a chave e, depois, inserir o ARN da chave do KMS.

O Amazon S3 só é compatível com chaves do KMS de criptografia simétrica, e não com chaves assimétricas do KMS. Para ter mais informações, consulte [Identificar chaves do KMS assimétricas](#) no Guia do desenvolvedor do AWS Key Management Service.

9. Para Chave do bucket, escolha Desabilitar. As chaves de bucket do S3 não comportam DSSE-KMS.
10. 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 da API REST

Quando você cria um objeto, ou seja, quando você carrega um novo objeto ou copia um objeto, é possível especificar a utilização de criptografia de camada dupla do lado do servidor com AWS KMS keys (DSSE-KMS) para criptografar os 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:dsse`. O Amazon S3 confirma que o objeto foi armazenado com a criptografia DSSE-S3 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:dsse`, também poderá usar os seguintes cabeçalhos de solicitação:

- `x-amz-server-side-encryption: AES256 | aws:kms | aws:kms:dsse`
- `x-amz-server-side-encryption-aws-kms-key-id: SSEKMSKeyId`

Tópicos

- [Operações da API REST do Amazon S3 compatíveis com DSSE-KMS](#)
- [Contexto de criptografia \(x-amz-server-side-encryption-context\)](#)
- [ID da chave do AWS KMS \(x-amz-server-side-encryption-aws-kms-key-id\)](#)

Operações da API REST do Amazon S3 compatíveis com DSSE-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`.

- [PutObject](#): ao carregar dados usando a operação de API PUT, é possível especificar esses cabeçalhos de solicitação.
- [CopyObject](#): quando você copia um objeto, há um objeto de origem e um objeto de destino. Ao transmitir cabeçalhos de DSSE-KMS com a operação `CopyObject`, 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 do lado do 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.
- [CreateMultipartUpload](#): ao carregar objetos grandes usando a operação de API de carregamento fracionado, você pode especificar esses cabeçalhos na solicitação `CreateMultipartUpload`.

Os cabeçalhos de resposta das operações da API REST a seguir retornam o cabeçalho `x-amz-server-side-encryption` quando um objeto é armazenado com criptografia do lado do servidor.

- [PutObject](#)
- [CopyObject](#)
- [Objeto POST](#)

- [CreateMultipartUpload](#)
- [UploadPart](#)
- [UploadPartCopy](#)
- [CompleteMultipartUpload](#)
- [GetObject](#)
- [HeadObject](#)

Important

- Todas as solicitações GET e PUT para um objeto protegido por AWS KMS falharão se elas não forem feitas usando Secure Sockets Layer (SSL), Transport Layer Security (TLS) ou Signature versão 4.
- Se o objeto usar DSSE-KMS, os cabeçalhos de solicitação de criptografia não deverão ser enviados para solicitações GET e HEAD; do contrário, será exibido um erro HTTP 400 (BadRequest).

Contexto de criptografia (**x-amz-server-side-encryption-context**)

Se você especificar `x-amz-server-side-encryption:aws:kms:dsse`, 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 como o par de contexto de criptografia, por exemplo `arn:aws:s3:::object_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](#). 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 cabeçalho `x-amz-server-side-encryption:aws:kms:dsse`, mas não fornecer o cabeçalho `x-amz-server-side-encryption-aws-kms-key-id`, o Amazon S3 usará a Chave gerenciada pela AWS (`aws/s3`) 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. Para obter mais informações sobre essas chaves, consulte [Chaves do KMS de criptografia simétrica](#) no Guia do desenvolvedor do AWS Key Management Service.

Usando a AWS CLI

Ao carregar um novo objeto ou copiar um objeto, é possível especificar a utilização de criptografia de DSSE-KMS para criptografar os dados. Para isso, adicione o parâmetro `--server-side-encryption aws:kms:dsse` à solicitação. Use o parâmetro `--ssekms-key-id example-key-id` para adicionar a [chave do AWS KMS gerenciada pelo cliente](#) que você criou. Se você especificar `--server-side-encryption aws:kms:dsse`, mas não fornecer um ID de chave do AWS KMS, o Amazon S3 usará a chave gerenciada pela AWS (`aws/s3`).

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key example-object-key --server-side-encryption aws:kms:dsse --ssekms-key-id example-key-id --body filepath
```

Você pode criptografar um objeto não criptografado para usar a DSSE-KMS copiando o objeto de volta no lugar.

```
aws s3api copy-object --bucket DOC-EXAMPLE-BUCKET --key example-object-key --body filepath --bucket DOC-EXAMPLE-BUCKET --key example-object-key --sse aws:kms:dsse --sse-kms-key-id example-key-id --body filepath
```

Como usar criptografia do lado do servidor com chaves 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. Ao usar criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C), você pode armazenar suas próprias chaves de criptografia. Com a chave de criptografia que você fornece como parte de sua solicitação, o Amazon S3 gerencia a criptografia de dados ao gravar em discos, e a descriptografia dos dados quando você acessa seus objetos. Portanto, você não precisa manter um código para executar a criptografia e a descriptografia de dados. A única coisa que você precisa fazer é gerenciar as chaves de criptografia fornecidas.

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. Depois, o Amazon S3 remove 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, depois decifra o objeto antes de retornar os dados de objeto.

Não há cobranças adicionais pelo uso de SSE-C. No entanto, as solicitações para configurar e usar SSE-C incorrem em cobranças padrão de solicitação do Amazon S3. Para obter mais informações sobre preços, consulte [Preços do Amazon S3](#).

Note

O Amazon S3 não armazena a chave de criptografia que você fornece. Em vez disso, ele armazena um valor de código de autenticação de mensagens por hash (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.

A replicação do S3 oferece suporte a objetos que são criptografados com SSE-C. Para obter mais informações sobre replicação de objetos criptografados, consulte [the section called “Replicar objetos criptografados \(SSE-C, SSE-S3, SSE-KMS, DSSE-KMS\)”](#).

Para obter mais informações sobre SSE-C, consulte os tópicos a seguir.

Tópicos

- [Visão geral do SSE-C](#)
- [Exigência e restrição de SSE-C](#)
- [Pre-signed URLs e SSE-C](#)
- [Especificação de criptografia no lado do servidor com chaves fornecidas pelo cliente \(SSE-C\).](#)

Visão geral do SSE-C

Esta seção fornece uma visão geral da SSE-C. Ao usar SSE-C, tenha as seguintes considerações em mente:

- 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. Descarte a chave e alterne conforme apropriado.

- A etiqueta de entidade (ETag) na resposta não é o hash 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 habilitado, 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 SSE-C de todos os objetos em um bucket específico do Amazon S3, use 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 incluam 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"
        }
      }
    }
  ]
}
```

Você também pode usar uma política para restringir a criptografia do lado do servidor de todos os objetos em um bucket específico do Amazon S3. Por exemplo, a política de bucket a seguir nega permissão de upload de objeto (`s3:PutObject`) para todos se a solicitação inclui o cabeçalho `x-amz-server-side-encryption-customer-algorithm` solicitando 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"
      }
    }
  }
]
```

Important

Se você usar uma política de bucket para exigir SSE-C em `s3:PutObject`, deverá incluir o cabeçalho `x-amz-server-side-encryption-customer-algorithm` em todas as solicitações de carregamento fracionado (`CreateMultipartUpload`, `UploadPart` e `CompleteMultipartUpload`).

Pre-signed URLs e SSE-C

Você pode gerar um URL pré-assinado que pode ser usado para operações, como fazer upload de um objeto novo, recuperar um objeto existente ou recuperar metadados de objeto. Os URLs pré-assinados oferecem suporte a SSE-C da seguinte maneira:

- Ao criar um URL pré-assinado, você deve especificar o algoritmo usando o cabeçalho `x-amz-server-side-encryption-customer-algorithm` no cálculo de assinatura.
- Ao usar o URL pré-assinado 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 na solicitação de aplicação cliente.

Note

Para objetos que não usam SSE-C, é possível gerar um URL pré-assinado e colá-lo diretamente em um navegador para acessar os dados.

No entanto, não é possível fazer isso para objetos SSE-C porque, além do URL pré-assinado, você também precisa incluir cabeçalhos HTTP específicos para objetos SSE-C. Dessa forma, é possível usar URLs pré-assinados para objetos SSE-C somente de maneira programática.


Para obter mais informações sobre pre-signed URLs, consulte [the section called “Trabalhar com URLs pré-assinados”](#).

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 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 da API REST

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.
- Carregamento fracionado: ao fazer upload de objetos grandes usando a API de carregamentos fracionados, você pode especificar esses cabeçalhos. Especifique esses cabeçalhos na solicitação de inicialização (consulte [Iniciar carregamento fracionado](#)) e em cada solicitação de upload de parte subsequente (consulte [Fazer upload de parte](#) ou [Fazer upload de parte: cópia](#)). 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 AES256.
<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:

- Colocar objeto: faz upload de um objeto e solicita a criptografia de servidor, usando uma chave de criptografia fornecida pelo cliente.
- Obter objeto: faz 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.

- 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](#).

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 [Como usar criptografia do lado do servidor com chaves fornecidas pelo cliente \(SSE-C\)](#). Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3](#).

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](#) e [Usar os AWS SDKs \(API de baixo nível\)](#).

Para obter informações sobre SSE-C, consulte [Como usar criptografia do lado do servidor com chaves fornecidas pelo cliente \(SSE-C\)](#). Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código do Amazon S3 .NET](#).

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](#)). 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](#)).

- 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 \(API de baixo nível\)](#).

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 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](#)). 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](#).

```
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 \(API de baixo nível\)](#).

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 do lado do cliente

A criptografia do lado do cliente é o ato de criptografar dados localmente para ajudar a garantir sua segurança em trânsito e em repouso. Para criptografar os objetos antes de enviá-los ao Amazon S3, use o cliente de criptografia do Amazon S3. Quando os objetos são criptografados dessa maneira, eles não são expostos a terceiros, inclusive a AWS. O Amazon S3 recebe os objetos já criptografados. O Amazon S3 não atua na criptografia ou descriptografia dos objetos. Você pode usar o cliente de criptografia do Amazon S3 e a [criptografia do lado do servidor](#) para criptografar os dados. Quando você envia objetos criptografados para o Amazon S3, o Amazon S3 não reconhece os objetos como sendo criptografados, ele detecta apenas objetos típicos.

O cliente de criptografia do Amazon S3 funciona como um intermediário entre você e o Amazon S3. Depois de instanciar o cliente de criptografia do Amazon S3, os objetos são automaticamente criptografados e descriptografados como parte do Amazon S3 e das solicitações `PutObject` e `GetObject`. Todos os objetos são criptografados com uma chave de dados exclusiva. O cliente de criptografia do Amazon S3 não usa nem interage com chaves de bucket, mesmo se você especificar uma chave do KMS como a chave de encapsulamento.

O Guia do desenvolvedor do cliente de criptografia do Amazon S3 concentra-se nas versões 3.0 e posteriores do cliente de criptografia do Amazon S3. Para obter mais informações, consulte [Qual é o cliente de criptografia do Amazon S3?](#) no Guia do desenvolvedor do cliente de criptografia do Amazon S3.

Para obter mais informações sobre versões anteriores do cliente de criptografia do Amazon S3, consulte o Guia do desenvolvedor do SDK AWS de acordo com a sua linguagem de programação.

- [AWS SDK for Java](#)
- [AWS SDK for .NET](#)
- [AWS SDK for Go](#)
- [AWS SDK for PHP](#)
- [AWS SDK for Ruby](#)
- [AWS SDK for C++](#)

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 ter 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 ser compatíveis com o Transport Layer Security (TLS) 1.2. Recomendamos o TLS 1.3. 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](#).

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 para o Amazon S3 por meio de endpoints de interface permanecem 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 [AWS Direct Connect Guia do usuário do](#) e o [AWS Site-to-Site VPNGuia do usuário do](#).

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](#)
- [Restrições e limitações do AWS PrivateLink do Amazon S3](#)
- [Criar um endpoint da VPC](#)
- [Acessar endpoints da interface do Amazon S3](#)
- [DNS privado](#)
- [Acessar buckets, pontos de acesso e operações de API de controle do Amazon S3 de endpoints de interface do S3](#)
- [Atualizar uma configuração de DNS on-premises](#)
- [Criar uma política de VPC endpoint para o Amazon S3](#)

Tipos de VPC endpoints para o Amazon S3

Você pode usar dois tipos de endpoints da VPC para acessar o Amazon S3: endpoints de gateway e endpoints de interface (usando o 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.

Endpoints de gateway para o Amazon S3	Endpoints de interface para o Amazon S3
Usar endereços IP públicos do Amazon S3	Usar endereços IP privados da VPC para acessar o Amazon S3
Usar os mesmos nomes DNS do Amazon S3	Exigir nomes DNS do Amazon S3 específicos do endpoint
Não permita o acesso pelo ambiente on-premises	Permitir acesso desde on-premises
Não permita o acesso por 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 [Considerações sobre o endpoint 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](#)
- [Endpoints globais herdados](#)
- [Endpoints S3 traço região](#)
- [Endpoints de pilha dupla do Amazon S3](#)
- Usar [CopyObject](#) ou [UploadPartCopy](#) entre buckets em diferentes Regiões da AWS
- Transport Layer Security (TLS) 1.1

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

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 DNS regionais incluem um ID de endpoint da VPC 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 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.

DNS privado

As opções de DNS privado para endpoints da VPC de interface simplificam o roteamento do tráfego do S3 pelos endpoints da VPC e ajudam você a aproveitar o caminho de rede de menor custo disponível para a aplicação. Você pode usar opções de DNS privadas para rotear o tráfego regional do S3 sem atualizar seus clientes do S3 para usar os nomes DNS específicos dos endpoints de interface ou gerenciar a infraestrutura de DNS. Com os nomes DNS privados ativados, as consultas ao DNS regionais do S3 são resolvidas para os endereços IP privados de AWS PrivateLink dos seguintes endpoints:

- Endpoints regionais de bucket (por exemplo, `s3.us-east-1.amazonaws.com`)
- Endpoints de controle (por exemplo, `s3-control.us-east-1.amazonaws.com`)
- Endpoints de pontos de acesso (por exemplo, `s3-accesspoint.us-east-1.amazonaws.com`)

Se você tiver um endpoint de gateway na VPC, poderá rotear automaticamente as solicitações na VPC pelo endpoint do gateway do S3 existente e as solicitações on-premises pelo endpoint da interface. Essa abordagem permite que você otimize os custos da rede usando endpoints de gateway, que não são cobrados, para o tráfego na VPC. As aplicações on-premises podem usar AWS PrivateLink com a ajuda do endpoint de resolvidor de entrada. A Amazon fornece um servidor de DNS à VPC, o Route 53 Resolver. Um endpoint do Resolver de entrada encaminha consultas de DNS da rede on-premises para o Route 53 Resolver.

Important

Para aproveitar o caminho de rede de menor custo ao usar Habilitar DNS privado somente para endpoints de entrada, um endpoint de gateway deve estar presente na VPC. A presença de um endpoint de gateway ajuda a garantir que o tráfego na VPC sempre seja roteado pela rede privada da AWS quando a opção Habilitar DNS privado somente para endpoints de entrada é selecionada. Você deve manter esse endpoint de gateway enquanto estiver com a opção Habilitar DNS privado somente para endpoints de entrada selecionada. Se você quiser excluir o endpoint de gateway, primeiro desmarque Habilitar DNS privado somente para endpoints de entrada.

Se você quiser atualizar um endpoint de interface existente para Habilitar DNS privado somente para endpoints de entrada, primeiro confirme se a VPC tem um endpoint de gateway do S3. Para obter mais informações sobre endpoints de gateway e gerenciamento de nomes DNS privados, consulte [Gateway de endpoints da VPC](#) e [Gerenciar nomes DNS](#), respectivamente, no Guia do AWS PrivateLink.

A opção Habilitar DNS privado somente para endpoints de entrada está disponível somente para serviços compatíveis com endpoints de gateway.

Para obter mais informações sobre a criação de um endpoint da VPC que usa Habilitar DNS privado somente para endpoints de entrada, consulte [Criar um endpoint de interface](#) no Guia do AWS PrivateLink.

Usar o console da VPC

No console, você tem duas opções: Habilitar nome DNS e Habilitar DNS privado somente para endpoints de entrada. Habilitar nome DNS é uma opção compatível com AWS PrivateLink. Ao usar a opção Habilitar nome de DNS, você pode usar a conectividade privada da Amazon com o Amazon

S3, enquanto faz solicitações aos nomes DNS de endpoints públicos padrão. Quando essa opção é ativada, os clientes podem aproveitar o caminho de rede de menor custo disponível para a aplicação.

Quando você habilita nomes DNS privados em um endpoint da VPC de interface existente ou novo para o Amazon S3, a opção Habilitar DNS privado somente para endpoints de entrada é selecionada por padrão. Se essa opção for selecionada, as aplicações usarão somente endpoints de interface para o tráfego on-premises. Esse tráfego na VPC usa automaticamente os endpoints de gateway de menor custo. Como alternativa, você pode desmarcar Habilitar DNS privado somente para endpoints de entrada para rotear todas as solicitações do S3 pelo endpoint de interface.

Usar o AWS CLI

Se você não especificar um valor para `PrivateDnsOnlyForInboundResolverEndpoint`, o padrão será `true`. No entanto, antes da VPC aplicar as configurações, ela realiza uma verificação para garantir que você tenha um endpoint de gateway presente na VPC. Se um endpoint de gateway estiver presente na VPC, a chamada será bem-sucedida. Caso contrário, você receberá a seguinte mensagem de erro:

Para definir `PrivateDnsOnlyForInboundResolverEndpoint` como verdadeiro, a VPC `vpce_id` deve ter um endpoint de gateway para o serviço.

Para um novo endpoint da VPC de interface

Use os atributos `private-dns-enabled` e `dns-options` para habilitar o DNS privado por meio da linha de comando. A opção `PrivateDnsOnlyForInboundResolverEndpoint` no atributo `dns-options` deve ser definida como `true`. Substitua os *user input placeholders* por suas próprias informações.

```
aws ec2 create-vpc-endpoint \  
--region us-east-1 \  
--service-name s3-service-name \  
--vpc-id client-vpc-id \  
--subnet-ids client-subnet-id \  
--vpc-endpoint-type Interface \  
--private-dns-enabled \  
--ip-address-type ip-address-type \  
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=true \  
--security-group-ids client-sg-id
```

Para um endpoint da VPC existente

Se você quiser usar o DNS privado para um endpoint da VPC existente, use o comando de exemplo a seguir e substitua *user input placeholders* por suas próprias informações.

```
aws ec2 modify-vpc-endpoint \  
--region us-east-1 \  
--vpc-endpoint-id client-vpc-id \  
--private-dns-enabled \  
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=false
```

Se você quiser atualizar um endpoint da VPC existente para habilitar o DNS privado somente para o resolvidor de entrada, use o exemplo a seguir e substitua os valores de amostra pelos seus próprios valores.

```
aws ec2 modify-vpc-endpoint \  
--region us-east-1 \  
--vpc-endpoint-id client-vpc-id \  
--private-dns-enabled \  
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=true
```

Acessar buckets, pontos de acesso e operações de API de controle do Amazon S3 de endpoints de interface do S3

Você pode usar a AWS CLI ou AWS SDKs para acessar buckets, pontos de acesso do S3 e operações de API de controle do Amazon 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`.

Details	Subnets	Security Groups	Policy	Notifications	Tags
Endpoint ID	vpce-0e25b8cdd720f900e	VPC ID	vpc-0c0ccb9d87b1734bd VPCStack VPC		
Status	available	Status message			
Creation time	January 8, 2021 at 1:30:11 AM UTC-8	Service name	com.amazonaws.us-east-1.s3		
Endpoint type	Interface	DNS names	*.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com (Z7HUB22UULQXV)		

Ao usar o nome DNS para acessar um recurso, substitua ***** pelo valor apropriado. Os valores apropriados a serem usados no lugar de ***** são os seguintes:

- bucket
- accesspoint
- control

Por exemplo, para acessar um bucket, use um nome DNS como este:

```
bucket.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com
```

Para obter exemplos de como usar nomes DNS para acessar buckets, pontos de acesso e operações da API de controle do Amazon S3, consulte as seções a seguir de [Exemplos do AWS CLI](#) e [Exemplos de AWS SDK](#).

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

Para acessar os buckets do S3, os pontos de acesso do S3 ou as operações de API de controle do Amazon S3 por meio de endpoints de interface do S3 em comandos AWS CLI, use os parâmetros `--region` e `--endpoint-url`.

Exemplo: usar um URL de endpoint para listar objetos no bucket

No exemplo a seguir, substitua o nome do bucket *my-bucket*, a região *us-east-1* e o nome de DNS do ID do endpoint da VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* por suas próprias informações.

```
aws s3 ls s3://my-bucket/ --region us-east-1 --endpoint-url
https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com
```

Exemplo: usar um URL de endpoint para listar objetos de um ponto de acesso

- Método 1: usar o nome do recurso da Amazon (ARN) do ponto de acesso com o endpoint do ponto de acesso

Substitua o ARN *us-east-1:123456789012:accesspoint/accesspointexamplename*, a região *us-east-1* e o ID do endpoint da VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* por suas próprias informações.

```
aws s3api list-objects-v2 --bucket arn:aws:s3:us-east-1:123456789012:accesspoint/
accesspointexamplename --region us-east-1 --endpoint-url
https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com
```

Se você não conseguir executar o comando com êxito, atualize a AWS CLI para a versão mais recente e tente novamente. Para obter mais informações sobre as instruções de atualização, consulte [Instalar ou atualizar a versão mais recente da AWS CLI](#) no Guia do usuário do AWS Command Line Interface.

- Método 2: usar o alias do ponto de acesso com o endpoint do bucket regional

No exemplo a seguir, substitua o alias do ponto de acesso *accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias*, a região *us-east-1* e o ID do endpoint da VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* por suas próprias informações.

```
aws s3api list-objects-v2 --
bucket accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias
--region us-east-1 --endpoint-url https://bucket.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com
```

- Método 3: usar o alias do ponto de acesso com o endpoint do ponto de acesso

Primeiro, para construir um endpoint S3 com o bucket incluído como parte do nome do host, defina o estilo de endereçamento como `virtual` para `aws s3api` para uso. Para obter mais informações sobre `AWS configure`, consulte [Arquivos de configuração e credencial](#) no Guia do usuário do AWS Command Line Interface.

```
aws configure set default.s3.addressing_style virtual
```

Depois, no exemplo a seguir, substitua o alias do ponto de acesso *accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias*, a região *us-east-1* e o ID do endpoint da VPC *vpce-1a2b3c4d-5e6f.s3.us-*

east-1.vpce.amazonaws.com por suas próprias informações. Consulte mais informações sobre alias de ponto de acesso em [Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3](#).

```
aws s3api list-objects-v2 --  
bucket accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias --  
region us-east-1 --endpoint-url https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-  
east-1.vpce.amazonaws.com
```

Exemplo: usar um URL de endpoint para listar trabalhos com uma operação de API de controle do S3

No exemplo a seguir, substitua a região *us-east-1*, o ID do endpoint da VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* e o ID da conta *12345678* por suas próprias informações.

```
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

Para acessar buckets do S3, pontos de acesso do S3 ou operações da API de controle do Amazon S3 por meio de endpoints de interface do S3 ao usar os AWS SDKs, atualize os SDKs para a versão mais recente. Depois, configure seus clientes para usar um URL de endpoint para acessar um bucket, ponto de acesso ou operações de API de controle do Amazon 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 da VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* por suas próprias informações.

```
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 da VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* por suas próprias informações.

```
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 Amazon S3

No exemplo a seguir, substitua a região *us-east-1* e o ID do endpoint da VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* por suas próprias informações.

```
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 suas próprias informações.

```
// 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 suas próprias informações.

```
// 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: usar um URL de endpoint para acessar uma operação de API de controle do Amazon S3

No exemplo a seguir, substitua o ID do endpoint da VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* por suas próprias informações.

```
// 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 suas próprias informações.

```
// 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 suas próprias informações.

```
// 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 Amazon 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 suas próprias informações.

```
// 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()
```

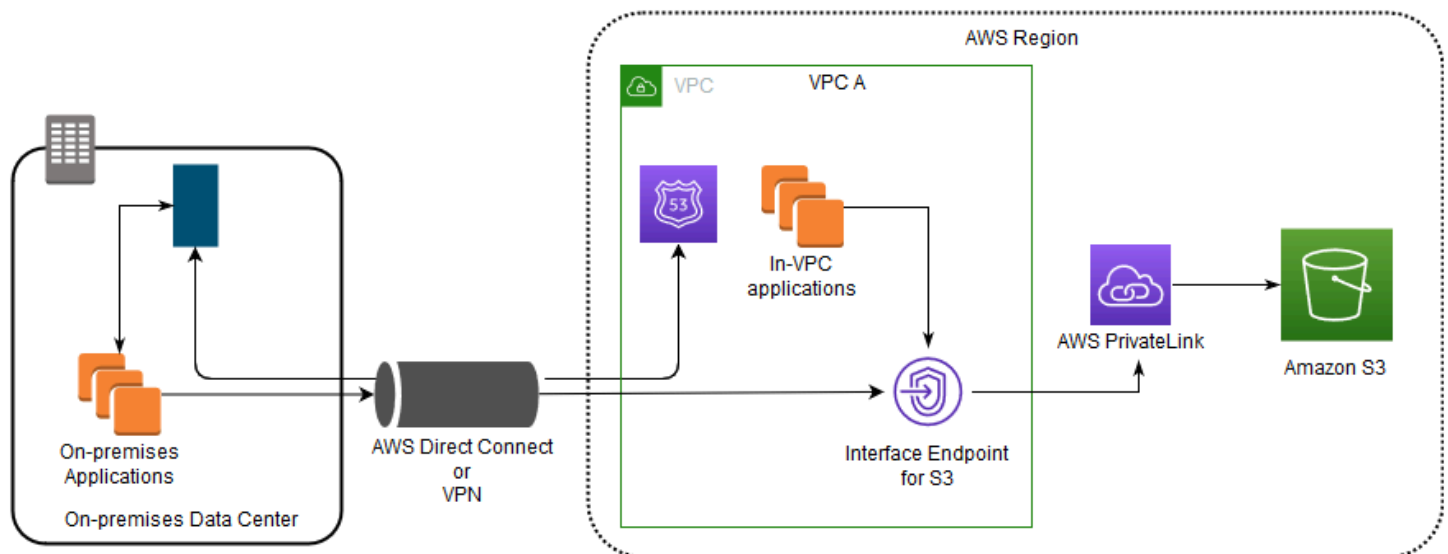
Atualizar uma configuração de DNS on-premises

Ao usar nomes de DNS específicos do endpoint para acessar os endpoints de interface do Amazon S3, você não precisa atualizar seu resolvidor 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.



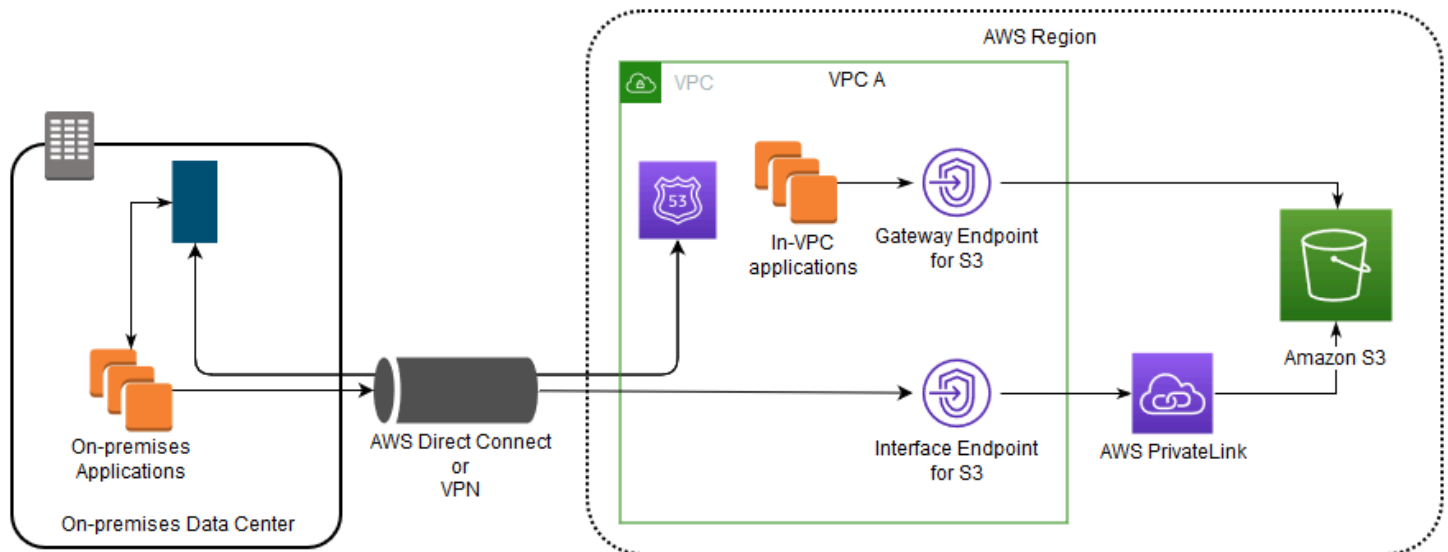
O diagrama ilustra o seguinte:

- 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 adotar essa abordagem, você permite que as aplicações na

VPC continuam acessando o Amazon S3 por meio do endpoint de gateway, que não é cobrado. Depois, apenas as aplicações on-premises usariam endpoints de interface para acessar o Amazon S3. Para acessar o Amazon 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 endpoint da VPC específico usando a condição `aws:sourceVpce` 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](#)
- [Exemplo: restringir o acesso a buckets em uma conta específica a partir de um endpoint da VPC](#)
- [Exemplo: restringir o acesso a um endpoint da VPC específico na política de bucket do S3](#)

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. Esse tipo de política será útil se houver outros Serviços da AWS em sua VPC que usem buckets. A política de bucket a seguir restringe o acesso somente a `DOC-EXAMPLE-BUCKET1`. Para usar essa política de endpoint, substitua `DOC-EXAMPLE-BUCKET1` pelo nome do 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 de endpoint que restrinja o acesso apenas aos buckets do S3 em uma Conta da AWS específica. Para impedir que os clientes dentro da VPC acessem buckets que você

não possui, use a declaração a seguir na política de endpoint. 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

Para especificar o ID da Conta da AWS do recurso que está sendo acessado, você pode usar a chave `aws:ResourceAccount` ou `s3:ResourceAccount` na política do IAM. No entanto, saiba que alguns 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 pelo endpoint da VPC **vpce-1a2b3c4d**. Essa política negará todo acesso ao bucket se o endpoint especificado não estiver sendo usado. 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 de bucket, substitua `DOC-EXAMPLE-BUCKET2` e `vpce-1a2b3c4d` pelo nome e endpoint do bucket.

Important

- Ao aplicar as políticas de bucket do Amazon S3 a seguir para restringir o acesso somente a certos endpoints da VPC, você pode bloquear o acesso ao bucket inadvertidamente. As políticas de bucket destinadas especificamente a limitar o acesso do bucket às conexões originadas do 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 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:::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).

Gerenciamento de identidade e acesso para o Amazon S3

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 (fazer login) e autorizado (ter permissões) para usar os recursos do Amazon S3. O IAM é um AWS service (Serviço da AWS) que pode ser usado sem custo adicional.

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Tópicos

- [Público](#)
- [Autenticando com identidades](#)
- [Gerenciamento do acesso usando políticas](#)
- [Como o Amazon S3 funciona com o IAM](#)
- [Políticas e permissões no Amazon S3](#)
- [Políticas de bucket para o Amazon S3](#)
- [Políticas baseadas em identidade do Amazon S3](#)
- [Demonstrações que usam políticas para gerenciar o acesso a recursos do Amazon S3](#)
- [Como o Amazon S3 autoriza uma solicitação](#)
- [Políticas gerenciadas da AWS para o Amazon S3](#)
- [Usar funções vinculadas a serviços para o Amazon S3 Storage Lens](#)
- [Solução de problemas de identidade e acesso do Amazon S3](#)

Público

A forma de usar o AWS Identity and Access Management (IAM) varia em função do trabalho realizado no Amazon S3.

Usuário do serviço: se você usar o serviço do Amazon S3 para fazer seu trabalho, o administrador fornecerá as credenciais e as permissões necessárias. À medida que usar mais recursos do Amazon S3 para fazer trabalhar, você poderá precisar de permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se você não puder acessar um recurso no Amazon S3, consulte [Solução de problemas de identidade e acesso do Amazon S3](#).

Administrador do serviço: se estiver a cargo dos recursos do Amazon S3 em sua empresa, você provavelmente terá acesso total ao Amazon S3. Cabe a você determinar quais funcionalidades e recursos do Amazon S3 os usuários do seu serviço devem acessar. Assim, você deve enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os Introdução ao IAM. Para saber mais sobre como a empresa pode usar o IAM com o Amazon S3, consulte [Como o Amazon S3 funciona com o IAM](#).

Administrador do IAM: se você for um administrador do IAM, talvez deseje saber detalhes sobre como pode escrever políticas para gerenciar o acesso ao Amazon S3. Consulte exemplos de políticas baseadas em identidade do Amazon S3 que podem ser usadas no IAM em [Políticas baseadas em identidade do Amazon S3](#).

Autenticando com identidades

A autenticação é a forma como você faz login na AWS usando suas credenciais de identidade. É necessário ser autenticado (fazer login na AWS) como o usuário-raiz da Usuário raiz da conta da AWS, como usuário do IAM ou assumindo um perfil do IAM.

Você pode fazer login na AWS como uma identidade federada usando credenciais fornecidas por uma fonte de identidades. AWS IAM Identity Center Aos usuários do IAM Identity Center, a autenticação única da empresa e as suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como uma identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Quando você acessa a AWS usando a federação, está indiretamente assumindo um perfil.

É possível fazer login no ou no portal de acesso da AWS Management Console dependendo do tipo de usuário que você é. AWS Para obter mais informações sobre como fazer login na AWS, consulte

[How to sign in to your \(Como fazer login na conta da\) Conta da AWS](#) no Início de Sessão da AWS User Guide (Guia do usuário do)..

Se você acessar a AWS programaticamente, a AWS fornecerá um kit de desenvolvimento de software (SDK) e uma interface da linha de comando (CLI) para você assinar criptograficamente as solicitações usando as suas credenciais. Se você não utilizar as ferramentas da AWS, deverá assinar as solicitações por conta própria. Para obter mais informações sobre como usar o método recomendado para assinar solicitações por conta própria, consulte [Assinar solicitações de API da AWS](#) no Guia do usuário do IAM.

Independentemente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, a AWS recomenda o uso da autenticação multifator (MFA) para aumentar a segurança de sua conta. Para saber mais, consulte [Autenticação multifator](#) no Guia do usuário do AWS IAM Identity Center. [Usar a autenticação multifator \(MFA\) na AWS](#) no Guia do usuário do IAM.

Usuário-raiz da Conta da AWS

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos os atributos e Serviços da AWS na conta. Essa identidade, denominada usuário-raiz da Conta da AWS, e é acessada por login com o endereço de email e a senha que você usou para criar a conta. 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 [Tarefas que exigem credenciais de usuário-raiz](#) no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que os usuários, inclusive os que precisam de acesso de administrador, usem a federação com um provedor de identidades para acessar Serviços da AWS usando credenciais temporárias.

Identidade federada é um usuário de seu diretório de usuários corporativos, um provedor de identidades da web AWS Directory Service, o , o diretório do Centro de Identidade ou qualquer usuário que acesse os Serviços da AWS usando credenciais fornecidas por meio de uma fonte de identidade. Quando as identidades federadas acessam Contas da AWS, elas assumem perfis que fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o .AWS IAM Identity Center. Você pode criar usuários e grupos no Centro de Identidade do IAM ou se conectar e sincronizar

com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todas as suas Contas da AWS e aplicações. Para obter mais informações sobre o Centro de Identidade do IAM, consulte [“What is IAM Identity Center?” \(O que é o Centro de Identidade do IAM?\)](#) no AWS IAM Identity Center Guia do usuário do .

Grupos e usuários do IAM

Um [usuário do IAM](#) é uma identidade dentro da Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos depender de credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, recomendamos alternar as chaves de acesso. Para obter mais informações, consulte [Altere as chaves de acesso regularmente para casos de uso que exijam credenciais](#) de longo prazo no Guia do usuário do IAM.

Um [grupo do IAM](#) é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdmins e atribuir a esse grupo permissões para administrar atributos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte [Quando criar um usuário do IAM \(em vez de um perfil\)](#) no Guia do usuário do IAM.

Perfis do IAM

um perfil do IAM https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html é uma identidade dentro da Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. É possível assumir temporariamente um perfil do IAM no AWS Management Console [alternando perfis](#). É possível assumir um perfil chamando uma operação de API da AWS CLI ou da AWS, ou usando um URL personalizado. Para obter mais informações sobre métodos para o uso de perfis, consulte [Usar perfis do IAM](#) no Guia do usuário do IAM.

Perfis do IAM com credenciais temporárias são úteis nas seguintes situações:

- **Acesso de usuário federado:** para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter mais informações sobre perfis para federação, consulte [Criar um perfil para um provedor de identidades de terceiros](#) no Guia do usuário do IAM. Se você usar o IAM Identity Center, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de permissões](#) no AWS IAM Identity Center Guia do usuário do .
- **Permissões temporárias para usuários do IAM:** um usuário ou um perfil do IAM pode assumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- **Acesso entre contas:** é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, alguns Serviços da AWS permitem que você anexe uma política diretamente a um atributo (em vez de usar um perfil como proxy). Para saber a diferença entre perfis e políticas baseadas em atributo para acesso entre contas, consulte [Como os perfis do IAM diferem das políticas baseadas em atributo](#) no Guia do usuário do IAM.
- **Acesso entre serviços:** alguns Serviços da AWS usam atributos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicações no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões da entidade principal de chamada, usando um perfil de serviço ou uma função vinculada ao serviço.
- **Encaminhamento de sessões de acesso (FAS):** qualquer pessoa que utilizar uma função ou usuário do IAM para realizar ações na AWS é considerada uma entidade principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O recurso FAS utiliza as permissões da entidade principal que chama um AWS service (Serviço da AWS), combinadas às permissões do AWS service (Serviço da AWS) solicitante, para realizar solicitações para serviços downstream. As solicitações de FAS só são feitas quando um serviço recebe uma solicitação que exige interações com outros Serviços da AWS ou com recursos para serem concluídas. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Encaminhar sessões de acesso](#).
- **Perfil de serviço:** um perfil de serviço é um perfil do IAM https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais

informações, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do usuário do IAM.

- Perfil vinculado a serviço: um perfil vinculado a serviço é um tipo de perfil de serviço vinculado a um AWS service (Serviço da AWS). O serviço pode assumir um perfil para executar uma ação em seu nome. Os perfis vinculados ao serviço aparecem em sua Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados ao serviço.
- Aplicações em execução no Amazon EC2: é possível usar um perfil do IAM para gerenciar credenciais temporárias para aplicações em execução em uma instância do EC2 e fazer solicitações da AWS CLI ou da AWS API. É preferível fazer isso e armazenar chaves de acesso na instância do EC2. Para atribuir um perfil da AWS a uma instância do EC2 e disponibilizá-la para todas as suas aplicações, crie um perfil de instância que esteja anexado a ela. Um perfil de instância contém a perfil e permite que os programas em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte [Usar um perfil do IAM para conceder permissões a aplicações em execução nas instâncias do Amazon EC2](#) no Guia do usuário do IAM.

Para saber se deseja usar as funções do IAM, consulte [Quando criar um perfil do IAM \(em vez de um usuário\)](#) no Guia do usuário do IAM.

Gerenciamento do acesso usando políticas

Você controla o acesso na AWS criando políticas e anexando-as a identidades ou atributos da AWS. Uma política é um objeto na AWS que, quando associado a uma identidade ou recurso, define suas permissões. A AWS avalia essas políticas quando uma entidade principal (usuário, usuário-raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada na AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte [Visão geral das políticas JSON](#) no Guia do usuário do IAM.

Os administradores podem usar as políticas da JSON AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e funções não têm permissões. Para conceder aos usuários permissão para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM a perfis, e os usuários podem assumir os perfis.

As políticas do IAM definem permissões para uma ação, independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de perfis do AWS Management Console, da AWS CLI ou da API da AWS.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Criar políticas do IAM](#) no Guia do usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas ainda mais como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas independentes que podem ser anexadas a vários usuários, grupos e perfis na Conta da AWS. As políticas gerenciadas incluem políticas gerenciadas pela AWS e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte [Escolher entre políticas gerenciadas e políticas em linha](#) no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificada pode executar nesse atributo e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. As entidades principais podem incluir contas, usuários, perfis, usuários federados ou Serviços da AWS.

Políticas baseadas em atributos são políticas em linha que estão localizadas nesse serviço. Não é possível usar as políticas gerenciadas da AWS do IAM em uma política baseada em atributos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3, o AWS WAF e a VPC são exemplos de serviços compatíveis com ACLs. Para saber mais sobre ACLs, consulte [Visão geral da lista de controle de acesso \(ACL\)](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

A AWS é compatível com tipos de política menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- **Limites de permissões:** um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (perfil ou usuário do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade e dos seus limites de permissões. As políticas baseadas em atributo que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte [Limites de permissões para identidades do IAM](#) no Guia do usuário do IAM.
- **Políticas de controle de serviço (SCPs):** SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (UO) no AWS Organizations. O AWS Organizations é um serviço para agrupar e gerenciar centralmente várias Contas da AWS pertencentes à empresa. Se você habilitar todos os atributos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP limita as permissões para entidades em contas-membro, incluindo cada `.Usuário raiz` da conta da AWS. Para obter mais informações sobre o Organizações e SCPs, consulte [Como os SCPs funcionam](#) no Guia do usuário do AWS Organizations.
- **Políticas de sessão:** são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em atributo. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte [Políticas de sessão](#) no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como a AWS determina se deve permitir uma solicitação quando há vários tipos de política envolvidos, consulte [Lógica da avaliação](#) de políticas no Guia do usuário do IAM.

Como o Amazon S3 funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao Amazon S3, saiba quais recursos do IAM estão disponíveis para uso com o Amazon S3.

Recursos do IAM que você pode usar com o Amazon S3

Atributo do IAM	Suporte ao Amazon S3
Políticas baseadas em identidade	Sim
Políticas baseadas em atributos	Sim
Ações de políticas	Sim
atributos de políticas	Sim
Chaves de condição de política (específicas do serviço)	Sim
ACLs	Sim
ABAC (tags em políticas)	Parcial
Credenciais temporárias	Sim
Sessões de acesso direto (FAS)	Sim
Perfis de serviço	Sim
Perfis vinculados ao serviço	Parcial

Consulte uma visão geral de como o Amazon S3 e outros serviços da AWS funcionam com a maioria dos recursos do IAM em [Serviços da AWS que funcionam com o IAM](#) no Guia do usuário do IAM.

Políticas baseadas em identidade do Amazon S3

É compatível com políticas baseadas em identidade	Sim
---	-----

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Criar políticas do IAM](#) no Guia do usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou atributos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Não é possível especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou função à qual ela está anexado. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte [Referência de elementos da política JSON do IAM](#) no Guia do Usuário do IAM.

Exemplos de políticas baseadas em identidade para o Amazon S3

Consulte exemplos de políticas baseadas em identidade do Amazon S3 em [Políticas baseadas em identidade do Amazon S3](#).

Políticas baseadas em recurso no Amazon S3

É compatível com políticas baseadas em atributos	Sim
--	-----

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificada pode executar nesse atributo e em que condições. Você deve [especificar uma entidade](#)

[principal](#) em uma política baseada em recursos. As entidades principais podem incluir contas, usuários, funções, usuários federados ou Serviços da AWS.

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em atributo. Adicionar uma entidade principal entre contas à política baseada em atributo é apenas metade da tarefa de estabelecimento da relação de confiança. Quando a entidade principal e o atributo estão em diferentes Contas da AWS, um administrador do IAM da conta confiável também deve conceder à entidade principal (usuário ou perfil) permissão para acessar o atributo. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em atributo conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, consulte [Como os perfis do IAM diferem de políticas baseadas em recursos](#) no Guia do usuário do IAM.

O serviço do Amazon S3 é compatível com políticas de bucket, políticas de pontos de acesso e concessões de acesso:

- Políticas de bucket são políticas baseadas em recursos que são anexadas a um bucket do Amazon S3. Uma política de bucket define quais entidades principais podem realizar ações no bucket.
- As políticas de ponto de acesso são políticas baseadas em recursos que são avaliadas em conjunto com a política de bucket correspondente.
- As concessões de acesso são um modelo simplificado para definir permissões de acesso aos dados no Amazon S3 por prefixo, bucket ou objeto. Consulte informações sobre a funcionalidade Concessões de Acesso do S3 em [Gerenciar o acesso com a funcionalidade Concessões de Acesso do S3](#).

Entidades principais para políticas de bucket

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"]}}
```

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](#) e [Exemplo 3: O proprietário do bucket concede permissões para objetos que não possui](#).

Note

Se uma identidade do IAM for excluída depois que você atualizar a política de bucket, a política de bucket mostrará um identificador exclusivo no elemento da entidade principal em vez de um ARN. Esses IDs exclusivos nunca são reutilizados; portanto você pode remover com segurança entidades principais com identificadores exclusivos de todas as suas instruções de política. Para obter mais informações sobre identificadores exclusivos, consulte [Identificadores do IAM](#) no Guia do usuário do IAM.

Conceder permissões anônimas

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.

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 usuários anônimos, esses dois métodos são equivalentes. 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.

Restringir permissões de recursos

Também é possível usar a política de recursos para restringir o acesso a recursos que, de outra forma, estariam disponíveis para entidades principais do IAM. Use uma instrução `Deny` para impedir o acesso.

O exemplo a seguir bloqueará o acesso se um protocolo de transporte seguro não for usado:

```
{ "Effect": "Deny",  
  "Principal": "*",  
  "Action": "s3:*",  
  "Resource": <bucket ARN>,  
  "Condition": {  
    "Boolean": { "aws:SecureTransport" : "false" }
```

```
}  
}
```

Usar "Principal": "*" para que essa restrição se aplique a todos é uma prática recomendada para essa política, em vez de tentar negar acesso somente a contas ou entidades principais específicas usando esse método.

Exigir acesso por URLs do CloudFront

Você pode exigir que os usuários acessem seu conteúdo do Amazon S3 somente usando URLs do CloudFront em vez de URLs do Amazon S3. Para fazer isso, crie um controle de acesso de origem (OAC) do CloudFront. Depois, altere as permissões em seus dados do S3. Na política de bucket, é possível definir o CloudFront como entidade principal da seguinte forma:

```
"Principal":{"Service":"cloudfront.amazonaws.com"}
```

Use um elemento Condition na política para permitir que o CloudFront acesse o bucket somente quando a solicitação for em nome da distribuição do CloudFront que contém a origem do S3.

```
    "Condition": {  
      "StringEquals": {  
        "AWS:SourceArn":  
"arn:aws:cloudfront::111122223333:distribution/CloudFront-distribution-ID"  
      }  
    }
```

Consulte mais informações sobre como solicitar acesso ao S3 por meio de URLs do CloudFront em [Restringir o acesso ao conteúdo de uma origem do Amazon Simple Storage Service](#) no Guia do desenvolvedor do Amazon CloudFront. Consulte mais informações sobre os benefícios de segurança e privacidade com o uso do Amazon CloudFront em [Configurar o acesso seguro e restringir o acesso ao conteúdo](#).

Exemplos de política baseada em recursos para o Amazon S3

- Consulte exemplos de política para buckets do Amazon S3 em [Políticas de bucket para o Amazon S3](#).
- Consulte exemplos de política para pontos de acesso em [Configurar políticas do IAM para uso de pontos de acesso](#).

Ações de políticas para o Amazon S3

Oferece suporte a ações de políticas	Sim
--------------------------------------	-----

Os administradores podem usar as políticas da JSON AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome que a operação de API da AWS associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Há também algumas operações que exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Veja a seguir diferentes tipos de relacionamento de mapeamento entre as operações de API do S3 e as ações de políticas necessárias.

- Mapeamento de um para um com o mesmo nome. Por exemplo, para usar a operação de API `PutBucketPolicy`, a ação de política `s3:PutBucketPolicy` é necessária.
- Mapeamento de um para um com nomes diferentes. Por exemplo, para usar a operação de API `ListObjectsV2`, a ação de política `s3:ListBucket` é necessária.
- Mapeamento de um para muitos. Por exemplo, para usar a operação de API `HeadObject`, a `s3:GetObject` é necessária. Além disso, quando você usa o Bloqueio de Objetos do S3 e deseja obter o status de retenção legal ou as configurações de retenção de um objeto, as ações de política `s3:GetObjectLegalHold` ou `s3:GetObjectRetention` correspondentes também são necessárias para que você possa usar a operação de API `HeadObject`.
- Mapeamento de muitos para um. Por exemplo, para usar as operações de API `ListObjectsV2` ou `HeadBucket`, a ação de política `s3:ListBucket` é necessária.

Consulte uma lista de ações do Amazon S3 para uso em políticas em [Actions defined by Amazon S3](#) na Referência de autorização do serviço. Consulte uma lista completa de operações de API do Amazon S3 em [Amazon S3 API Actions](#) na Referência de API do Amazon Simple Storage Service.

As ações da política no Amazon S3 usam o seguinte prefixo antes da ação:

```
s3
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [  
    "s3:action1",  
    "s3:action2"  
]
```

Operações de buckets

As operações de bucket são operações de API do S3 que operam no tipo de recurso de bucket. Por exemplo, `CreateBucket`, `ListObjectsV2` e `PutBucketPolicy`. As ações de política do S3 para operações de bucket exigem que o elemento `Resource` nas políticas de bucket ou nas políticas baseadas em identidade do IAM seja o identificador do nome do recurso da Amazon (ARN) do tipo de bucket do S3 no formato de exemplo a seguir.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
```

A política de bucket a seguir concede a permissão `s3:ListBucket` ao usuário *Akua* (com a conta *12345678901*) para realizar a operação de API [ListObjectsV2](#) e listar objetos em um bucket do S3.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "Allow Akua to list objects in the bucket",  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::12345678901:user/Akua"  
      },  
      "Action": [  
        "s3:ListBucket"  
      ],  
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"  
    }  
  ]  
}
```


Operações de bucket em políticas de ponto de acesso

As permissões concedidas em uma política de ponto de acesso entrarão em vigor somente se o bucket subjacente conceder as mesmas permissões. Ao usar os pontos de acesso do S3, você deve delegar o controle de acesso do bucket ao ponto de acesso ou adicionar as mesmas permissões nas políticas de ponto de acesso à política do bucket subjacente. Para ter mais informações, consulte [Configurar políticas do IAM para uso de pontos de acesso](#). Nas políticas de ponto de acesso, as ações de políticas do S3 para operações de bucket exigem que você use o ARN `accesspoint` para o elemento `Resource` no formato a seguir.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-ACCESS-POINT"
```

A política de ponto de acesso a seguir concede a permissão `s3:ListBucket` ao usuário *Akua* (com a conta *12345678901*) para realizar a operação de API [ListObjectsV2](#) por meio do ponto de acesso *DOC-EXAMPLE-ACCESS-POINT* do S3 para listar objetos no bucket associado ao ponto de acesso.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow Akua to list objects in the bucket through access point",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::12345678901:user/Akua"
      },
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-ACCESS-POINT"
    }
  ]
}
```

Note

Nem todas as operações de bucket são compatíveis com o ponto de acesso do S3. Para ter mais informações, consulte [Compatibilidade de ponto de acesso com operações do S3](#).

Operações com objetos

As operações de objeto são operações de API do S3 que atuam sobre o tipo de recurso do objeto. Por exemplo, `GetObject`, `PutObject` e `DeleteObject`. As ações de política do S3 para operações de objeto exigem que o elemento `Resource` nas políticas seja o ARN do objeto do S3 nos formatos de exemplo a seguir.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
```

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/prefix/*"
```

Note

O ARN do objeto deve conter uma barra após o nome do bucket, conforme mostrado nos exemplos anteriores.

A política de bucket a seguir concede a permissão `s3:PutObject` ao usuário *Akua* (com a conta *12345678901*) para realizar a operação de API [PutObject](#) e fazer upload de objetos em um bucket do S3.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow Akua to upload objects",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::12345678901:user/Akua"
      },
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    }
  ]
}
```

Operações de objetos em políticas de ponto de acesso

Ao usar os pontos de acesso do S3 para controlar o acesso às operações de objetos, você pode usar políticas de ponto de acesso. Quando você usa políticas de ponto de acesso, as ações de políticas do S3 para operações de objeto exigem que você use o ARN accesspoint para o elemento Resource no seguinte formato: `arn:aws:s3:region:account-id:accesspoint/access-point-name/object/resource`. Para operações de objeto que usam o ponto de acesso, você deve incluir o valor `/object/` após o ARN completo do ponto de acesso no elemento Resource. Aqui estão alguns exemplos.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-ACCESS-POINT/object/*"
```

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-ACCESS-POINT/object/prefix/*"
```

A política de ponto de acesso a seguir concede a permissão `s3:GetObject` ao usuário *Akua* (com a conta `12345678901`) para realizar a operação de API [GetObject](#) por meio do ponto de acesso `DOC-EXAMPLE-ACCESS-POINT` em todos os objetos no bucket associado ao ponto de acesso.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow Akua to get objects through access point",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::12345678901:user/Akua"
      },
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-ACCESS-POINT/object/*"
    }
  ]
}
```

Note

Nem todas as operações de objetos são compatíveis com o ponto de acesso do S3. Para ter mais informações, consulte [Compatibilidade de ponto de acesso com operações do S3](#).

Operações de ponto de acesso

As operações de ponto de acesso são operações de API do S3 que operam no tipo de recurso de accesspoint. Por exemplo, `CreateAccessPoint`, `DeleteAccessPoint` e `GetAccessPointPolicy`. As ações de política do S3 para operações de ponto de acesso só podem ser usadas em políticas baseadas em identidade do IAM, não em políticas de bucket ou políticas de ponto de acesso. As operações de pontos de acesso exigem que o elemento `Resource` seja o ARN `accesspoint` no formato de exemplo a seguir.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-ACCESS-POINT"
```

A política baseada em identidade do IAM a seguir concede a permissão `s3:GetAccessPointPolicy` para realizar a operação de API [GetAccessPointPolicy](#) no ponto de acesso `DOC-EXAMPLE-ACCESS-POINT` do S3.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Grant permission to retrieve the access point policy of access
point DOC-EXAMPLE-ACCESS-POINT",
      "Effect": "Allow",
      "Action": [
        "s3:GetAccessPointPolicy"
      ],
      "Resource": "arn:aws:s3:*:123456789012:access point/DOC-EXAMPLE-ACCESS-
POINT"
    }
  ]
}
```

Se você usa pontos de acesso, consulte [Operações de bucket em políticas de ponto de acesso](#) para controlar o acesso às operações de bucket e [Operações de objetos em políticas de ponto de acesso](#)

para controlar o acesso às operações de objetos. Consulte mais informações sobre como configurar políticas de ponto de acesso em [Configurar políticas do IAM para uso de pontos de acesso](#).

Operações de ponto de acesso do Object Lambda

Com o Amazon S3 Object Lambda, você pode adicionar seu próprio código às solicitações GET, LIST e HEAD do Amazon S3 para modificar e processar dados, conforme eles são retornados para uma aplicação. É possível fazer solicitações por um ponto de acesso do Object Lambda, que funciona da mesma maneira que fazer solicitações por outros pontos de acesso. Para ter mais informações, consulte [Transformar objetos com o S3 Object Lambda](#).

Consulte mais informações sobre como configurar políticas de operações de ponto de acesso do Object Lambda em [Configurar políticas do IAM para pontos de acesso do Object Lambda](#).

Operações de ponto de acesso multirregionais

Um ponto de acesso multirregional fornece um endpoint global que as aplicações podem usar para atender a solicitações de buckets do S3 localizados em várias Região da AWS. É possível usar um ponto de acesso multirregional para criar aplicações de várias regiões com a mesma arquitetura usada em uma única região, depois executar essas aplicações em qualquer lugar do mundo. Para ter mais informações, consulte [Pontos de acesso multirregionais no Amazon S3](#).

Consulte mais informações sobre como configurar políticas de operações de ponto de acesso multirregionais em [Exemplos de políticas de pontos de acesso multirregionais](#).

Operações de trabalho em lote

(Operações em Lote) As operações de trabalho são operações de API do S3 que operam no tipo de recurso de trabalho. Por exemplo, `DescribeJob` e `CreateJob`. As ações de política do S3 para operações de trabalho só podem ser usadas em políticas baseadas em identidade do IAM, não em políticas de buckets. Além disso, as operações de trabalho exigem que o elemento `Resource` nas políticas baseadas em identidade do IAM seja o ARN `job` no formato de exemplo a seguir.

```
"Resource": "arn:aws:s3:*:123456789012:job/*"
```

A política baseada em identidade do IAM a seguir concede a permissão `s3:DescribeJob` para realizar a operação de API [DescribeJob](#) no trabalho do Operações em Lote do S3 **DOC-EXAMPLE-JOB**.

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "Allow describing the Batch operation job DOC-EXAMPLE-JOB",
    "Effect": "Allow",
    "Action": [
      "s3:DescribeJob"
    ],
    "Resource": "arn:aws:s3:*:123456789012:job/DOC-EXAMPLE-JOB"
  }
]
}

```

Operações de configuração da Lente de Armazenamento do S3

Consulte mais informações sobre como definir as operações de configuração do Lente de Armazenamento do S3 em [Permissões do Amazon S3 Storage Lens](#).

Operações de conta

As operações da conta são operações de API do S3 que operam no nível da conta. Por exemplo, `GetPublicAccessBlock` (para conta). A conta não é um tipo de recurso definido pelo Amazon S3. As ações de política do S3 para operações de conta só podem ser usadas em políticas baseadas em identidade do IAM, não em políticas de buckets. Além disso, as operações da conta exigem que o elemento `Resource` nas políticas baseadas em identidade do IAM seja `"*"`.

A política baseada em identidade do IAM a seguir concede a permissão `s3:GetAccountPublicAccessBlock` para realizar a operação de API [GetPublicAccessBlock](#) no nível da conta e recuperar as configurações do Bloqueio de Acesso Público no nível da conta.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow retrieving the account-level Public Access Block settings",
      "Effect": "Allow",
      "Action": [
        "s3:GetAccountPublicAccessBlock"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}

```

```
    }  
  ]  
}
```

Exemplos de políticas para o Amazon S3

- Consulte exemplos de políticas baseadas em identidade do Amazon S3 em [Políticas baseadas em identidade do Amazon S3](#).
- Consulte exemplos de políticas baseadas em recursos do Amazon S3 em [Políticas de bucket para o Amazon S3](#) e [Configurar políticas do IAM para uso de pontos de acesso](#).

Recursos de políticas para o Amazon S3

Oferece suporte a atributos de políticas	Sim
--	-----

Os administradores podem usar as políticas da JSON AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Resource` de política JSON especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Como prática recomendada, especifique um recurso usando [Nome do recurso da Amazon \(ARN\)](#). Isso pode ser feito para ações que oferecem suporte a um tipo de atributo específico, conhecido como permissões em nível de atributo.

Para ações não compatíveis com permissões no nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Algumas ações de API do Amazon S3 são compatíveis com vários recursos. Por exemplo, `s3:GetObject` acessa `EXAMPLE-RESOURCE-1` e `EXAMPLE-RESOURCE-2`; portanto, uma entidade principal deve ter permissões para acessar os dois recursos. Para especificar vários recursos em uma única instrução, separe os ARNs com vírgulas.

```
"Resource": [  
  "EXAMPLE-RESOURCE-1",
```

```
"EXAMPLE-RESOURCE-2"
```

Os recursos no Amazon S3 são buckets, objetos, pontos de acesso ou trabalhos. Em uma política, use o nome do recurso da Amazon (ARN) do bucket, do objeto, do ponto de acesso ou do trabalho para identificar o recurso.

Consulte uma lista completa dos tipos de recurso do Amazon S3 e os respectivos ARNs em [Resources defined by Amazon S3](#) na Referência de autorização do serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte [Actions defined by Amazon S3](#).

Curingas para ARNs de recursos

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 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 política para ARNs de recursos

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 de usuário da pessoa que faz a solicitação.

Exemplos de políticas para o Amazon S3

- Consulte exemplos de políticas baseadas em identidade do Amazon S3 em [Políticas baseadas em identidade do Amazon S3](#).
- Consulte exemplos de políticas baseadas em recursos do Amazon S3 em [Políticas de bucket para o Amazon S3](#) e [Configurar políticas do IAM para uso de pontos de acesso](#).

Chaves de condição de política para o Amazon S3

Compatível com chaves de condição de política específicas do serviço Sim

Os administradores podem usar as políticas da JSON AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Condition` (ou `Condition` bloco de) permite que você especifique condições nas quais uma instrução está em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usam [agentes de condição](#), como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica AND. Se você especificar vários valores para uma única chave de condição, a AWS avaliará a condição usando uma operação lógica OR. Todas as condições devem ser atendidas para que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar as condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um atributo somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte [Elementos de política do IAM: variáveis e tags](#) no Guia do usuário do IAM.

A AWS é compatível com chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as AWS chaves de condição globais da , consulte [AWSChaves de contexto de condição globais da](#) no Guia do usuário do IAM.

Cada chave de condição do Amazon S3 é associada ao 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. Por exemplo, a chave de condição `s3:VersionId` usada para conceder permissão condicional à permissão `s3:GetObjectVersion` define o comportamento do parâmetro de consulta `versionId` que você define em uma solicitação GET Object.

Consulte uma lista de chaves de condição do Amazon S3 em [Condition keys for Amazon S3](#) na Referência de autorização do serviço. Para saber com quais ações e recursos é possível usar uma chave de condição, consulte [Actions defined by Amazon S3](#).

Exemplo: restringir uploads aos 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 A deseja restringir Dave, um usuário na Conta A, para que ele só possa carregar objetos no bucket armazenado 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"
          ]
        }
      }
    }
  ]
}
```

```
    }  
  }  
} ]  
}
```

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.

Exemplos de políticas para o Amazon S3

- Consulte exemplos de políticas baseadas em identidade do Amazon S3 em [Políticas baseadas em identidade do Amazon S3](#).
- Consulte exemplos de políticas baseadas em recursos do Amazon S3 em [Políticas de bucket para o Amazon S3](#) e [Configurar políticas do IAM para uso de pontos de acesso](#).

ACLs no Amazon S3

Oferece suporte a ACLs

Sim

No Amazon S3, as listas de controle de acesso (ACLs) controlam quais Contas da AWS têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

Important

A maioria dos casos de uso modernos no Amazon S3 não exige mais o uso de ACLs.

Consulte informações sobre como usar ACLs para controlar o acesso no Amazon S3 em [Gerenciar o acesso com ACLs](#).

ABAC com o Amazon S3

Oferece suporte a ABAC (tags em políticas) Parcial

O controle de acesso baseado em recurso (ABAC) é uma estratégia de autorização que define permissões com base em recursos. Na AWS, esses atributos são chamados de tags. É possível anexar tags a entidades do IAM (usuários ou perfis) e a muitos atributos da AWS. A marcação de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do atributo que ela está tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no [elemento de condição](#) de uma política usando as chaves de condição `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial.

Para obter mais informações sobre o ABAC, consulte [O que é ABAC?](#) no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte [Usar controle de acesso baseado em atributos \(ABAC\)](#) no Guia do usuário do IAM.

Consulte exemplos de políticas baseadas em identidade para limitar o acesso aos trabalhos de Operações em Lote do S3 com base em tags em [Controlar permissões para o recurso Operações em lote do S3 usando tags de trabalho](#).

ABAC e tags de objetos

Nas políticas de ABAC, os objetos usam tags `s3:` em vez de tags `aws:`. Para controlar o acesso a objetos com base em tags, forneça informações sobre a tag no [elemento de condição](#) de uma política usando as seguintes tags:

- `s3:ExistingObjectTag/tag-key`
- `s3:s3:RequestObjectTagKeys`

- `s3:RequestObjectTag/tag-key`

Consulte informações sobre como usar tags de objeto para controlar o acesso, incluindo exemplos de políticas de permissão, em [Marcação e políticas de controle de acesso](#).

Usar credenciais temporárias com o Amazon S3

Oferece suporte a credenciais temporárias Sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte [Serviços da AWS que funcionam com o IAM](#) no Guia do usuário do IAM.

Você está usando credenciais temporárias se faz login no AWS Management Console usando qualquer método, exceto um nome de usuário e uma senha. Por exemplo, quando você acessa a AWS usando o link de autenticação única (SSO) da empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar perfis, consulte [Alternar para um perfil \(console\)](#) no Guia do usuário do IAM.

Você pode criar credenciais temporárias manualmente usando a AWS CLI ou a API da AWS. Em seguida, você pode usar essas credenciais temporárias para acessar a AWS. A AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para mais informações, consulte [Credenciais de segurança temporárias no IAM](#).

Sessões de acesso direto para o Amazon S3

Suporte para o recurso Encaminhamento de sessões de acesso (FAS) Sim

Quando você usa um usuário do IAM ou um perfil para executar ações na AWS, você é considerado uma entidade principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O recurso FAS utiliza as permissões da entidade principal que chama um AWS service (Serviço da AWS), combinadas às permissões do AWS service (Serviço da AWS) solicitante, para realizar solicitações para serviços downstream. As solicitações de FAS só são feitas quando um serviço recebe uma solicitação que exige interações com outros Serviços da AWS ou

com recursos para serem concluídas. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Encaminhar sessões de acesso](#).

- O FAS é usado pelo Amazon S3 para fazer chamadas ao AWS KMS para descriptografar um objeto quando se usa a SSE-KMS para criptografá-lo. Para ter mais informações, consulte [Usar criptografia do lado do servidor com o AWS KMS \(SSE-KMS\)](#).
- A funcionalidade Concessões de Acesso do S3 também usa FAS. Depois de criar uma concessão de acesso aos seus dados do S3 para uma identidade específica, o beneficiário solicita uma credencial temporária da funcionalidade Concessões de Acesso do S3. A funcionalidade Concessões de Acesso do S3 obtém uma credencial temporária para o solicitante pelo AWS STS e vende a credencial a ele. Para ter mais informações, consulte [Solicitar acesso aos dados do Amazon S3 por meio da funcionalidade Concessões de Acesso do S3](#).

Perfis de serviço para o Amazon S3

Oferece suporte a perfis de serviço	Sim
-------------------------------------	-----

Um perfil de serviço é um [perfil do IAM](#) que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do usuário do IAM.

Warning

A alteração das permissões de um perfil de serviço pode interromper a funcionalidade do Amazon S3. Edite perfis de serviço somente quando o Amazon S3 instruir a fazer isso.

Perfis vinculados ao serviço para o Amazon S3

Oferece suporte a perfis vinculados ao serviço	Parcial
--	---------

Um perfil vinculado ao serviço é um tipo de perfil de serviço vinculado a um AWS service (Serviço da AWS). O serviço pode assumir o perfil de executar uma ação em seu nome. Os perfis vinculados ao

serviço aparecem em sua Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados ao serviço.

O Amazon S3 é compatível com perfis vinculados a serviço da Lente de Armazenamento do Amazon S3. Consulte detalhes de como criar ou gerenciar perfis vinculados ao serviço do Amazon S3 em [Usar funções vinculadas a serviços para o Amazon S3 Storage Lens](#).

Serviço do Amazon S3 como entidade principal

Nome do serviço na política	Recurso do S3	Mais informações
s3.amazonaws.com	Replicação do S3	Configuração da replicação
s3.amazonaws.com	Notificações de eventos do S3	Notificações de eventos do Amazon S3
s3.amazonaws.com	Inventário S3	Inventário do Amazon S3
access-grants.s3.amazonaws.com	Concessões de Acesso do S3	Registrar um local
batchoperations.s3.amazonaws.com	Operações em lote do S3	Conceder permissões para operações em lote do Amazon S3
logging.s3.amazonaws.com	Registro em log de acesso ao servidor do S3	Habilitar o log de acesso ao servidor do Amazon S3
storage-lens.s3.amazonaws.com	Lente de Armazenamento do S3	Exibição de métricas de lente de armazenamento do Amazon S3 usando uma exportação de dados

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.

Consulte uma lista completa de ações, recursos e condições do Amazon S3 em [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço.

No sentido mais básico, uma política contém os seguintes elementos:

- **Recurso:** o bucket, o objeto, o ponto de acesso ou o trabalho do Amazon S3 ao qual a política se aplica. Use o nome do recurso da Amazon (ARN) do bucket, do objeto, do ponto de acesso ou do trabalho para identificar o recurso.

Um exemplo de operações em nível de bucket:

- "Resource": "arn:aws:s3:::*bucket_name*".

Exemplos de operação em nível de objeto:

- "Resource": "arn:aws:s3:::*bucket_name*/*" para todos os objetos no bucket.

- "Resource": "arn:aws:s3:::*bucket_name/prefix*/*" para objetos com um prefixo específico no bucket.

Para ter mais informações, consulte [Recursos de políticas para o Amazon S3](#).

- **Ações:** 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 de políticas para o Amazon S3](#). 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** — 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 [Entidades principais para políticas de bucket](#).

- **Condição:** 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 políticas de bucket usando chaves de condição](#).

A política de bucket de exemplo a seguir mostra os elementos efeito, principal, ação e recurso. A política permite que Akua, um usuário na conta *ID da 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/Akua"
      },
      "Action": [
        "s3:GetObject",
        "s3:GetBucketLocation",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::awsexamplebucket1/*",
        "arn:aws:s3:::awsexamplebucket1"
      ]
    }
  ]
}
```

Consulte informações completas sobre a linguagem de políticas em [Políticas e permissões no IAM](#) e em [Referência de política JSON do IAM](#) no Guia do usuário do IAM.

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. No entanto, uma conta que

recebe permissões de outra conta não pode delegar permissão entre contas para outra Conta da AWS.

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](#).

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](#).

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 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"](#).

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 o 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](#). Para obter mais informações sobre ACLs, consulte [Visão geral da lista de controle de acesso \(ACL\)](#).

Important

Recomendamos que você não use credenciais de usuário raiz da Conta da AWS para fazer solicitações autenticadas. Em vez disso, crie um perfil do IAM e conceda acesso total a ele. Usuários com esse perfil são conhecidos como usuários administradores. Em vez de usar credenciais de usuário raiz da Conta da AWS, você pode usar credenciais atribuídas ao perfil administrador para interagir com a AWS e realizar tarefas, como criar um bucket, criar usuários e conceder permissões a eles. Consulte mais informações em [Credenciais de segurança da AWS](#) no Guia do usuário do IAM e em [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Políticas de bucket para o Amazon S3

Uma política de bucket é uma política baseada em recursos que você pode usar para conceder permissões de acesso ao bucket do Amazon S3 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 pertencentes a outras Contas da AWS.

A Propriedade de objeto do S3 é uma configuração em nível de bucket do Amazon S3 que você pode usar para controlar a propriedade de objetos carregados no bucket. Você também pode usar a Propriedade de objeto para desabilitar ou habilitar listas de controle de acesso (ACLs). Por padrão, a Propriedade de objeto está definida com a configuração Aplicada pelo proprietário do bucket e todas as ACLs estão desabilitadas. O proprietário do bucket possui todos os objetos do bucket e gerencia o acesso aos dados exclusivamente usando políticas.

As políticas de bucket usam uma linguagem de política do AWS Identity and Access Management (IAM) 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. Esses elementos incluem o solicitante, ações do S3, recursos e aspectos ou condições da solicitação (como o endereço IP usado para fazer a solicitação).

Por exemplo, você pode criar uma política de bucket que faça o seguinte:

- Concede a outras contas permissões entre contas para fazer upload de objetos em seu bucket do S3
- Garante que você, o proprietário do bucket, tenha controle total dos objetos enviados

Para ter mais informações, consulte [Exemplos de políticas de bucket do Amazon S3](#).

Os tópicos nesta seção fornecem exemplos e mostram como adicionar uma política de bucket no console do S3. Consulte mais informações sobre políticas baseadas em identidade em [Políticas baseadas em identidade do Amazon S3](#). Para obter informações sobre a linguagem da política de bucket, consulte [Políticas e permissões no Amazon S3](#).

Tópicos

- [Adicionar uma política de bucket usando o console do Amazon S3](#)
- [Controlar o acesso a partir de VPC endpoints com políticas de bucket](#)
- [Exemplos de políticas de bucket do Amazon S3](#)
- [Exemplos de políticas de bucket usando chaves de condição](#)

Adicionar uma política de bucket usando o console do Amazon S3

Use o [Gerador de Políticas da AWS](#) e 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 [Gerenciamento de identidade e acesso para o Amazon S3](#).

Solucione 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 do 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 obter orientação sobre como solucionar erros com uma política, consulte [Solucionar erros de acesso negado \(403 proibido\) no Amazon 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 Buckets.
3. 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.
4. Escolha a aba Permissões.
5. Em Bucket policy (Política de bucket), escolha Edit (Editar). A página Edit bucket policy (Editar política de bucket) é exibida.
6. Na página Edit bucket policy (Editar política de bucket), faça uma das seguintes ações:
 - Para ver exemplos de políticas de bucket no Guia do usuário do Amazon S3, escolha Policy examples (Exemplos de políticas).
 - Para gerar uma política automaticamente ou editar o JSON na seção Policy (Política), escolha Policy generator (Gerador de políticas).


Se você escolher Policy generator (Gerador de políticas), o Gerador de Políticas da AWS será aberto em uma nova janela.

- a. Na página Gerador de Políticas da AWS, em Selecionar tipo de política, escolha Política de bucket do S3.
- b. Adicione uma instrução inserindo as informações nos campos fornecidos e escolha Add Statement (Adicionar instrução). Repita esta etapa para todas as instruções que deseja adicionar. Para ter 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 Editar política de bucket exibe o ARN do bucket (nome do recurso da Amazon) do bucket atual acima do campo de texto 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.
7. Na caixa Policy (Política), edite a política existente ou cole a política de bucket do Gerador de Políticas da AWS. Solucione avisos de segurança, erros, avisos gerais e sugestões antes de salvar a política.

 Note

As políticas de bucket são limitadas a 20 KB.

8. (Opcional) Escolha Preview external access (Pré-visualizar o acesso externo) na canto inferior direito para pré-visualizar a forma como sua nova política afetará 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, escolha Go to Access Analyzer (Acessar o Access Analyzer) para [criar um analisador de contas](#) no IAM Access Analyzer. Para obter mais informações, consulte [Visualizar acesso](#) no Guia do usuário do IAM.
9. Escolha Save changes (Salvar alterações), o que levará você de volta à guia Permissions (Permissões).

Controlar o acesso a partir de VPC endpoints com políticas de bucket

É possível usar políticas de bucket do Amazon S3 para controlar o acesso a buckets por endpoints específicos da nuvem privada virtual (VPC) ou por VPCs específicas. Esta seção contém exemplos de políticas de bucket que você pode usar para controlar o acesso de endpoints da VPC ao bucket do Amazon S3. 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 da VPC permite criar uma conexão privada entre a VPC e outro AWS service (Serviço da AWS). Essa conexão privada não exige acesso pela internet, por meio de uma conexão de rede privada virtual (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. Consulte informações importantes sobre o uso de endpoints da VPC com o Amazon S3 em [Gateway endpoints](#) e em [Gateway endpoints for 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. Consulte informações sobre esse tipo de controle de acesso em [Controlling access to VPC endpoints using endpoint policies](#) 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](#)
- [Restringir o acesso a uma VPC específica](#)

⚠ Important

Ao aplicar políticas de bucket do Amazon S3 para os endpoints da VPC descritos nesta seção, talvez você bloqueie o acesso ao bucket de maneira não intencional. 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. Consulte informações sobre como corrigir esse problema em [Como corrigir uma política de bucket quando ela tem o ID da VPC ou do endpoint da VPC errado?](#) no Centro de Conhecimento da AWS Support.

Restringir o acesso a um VPC endpoint específico

O seguinte é um exemplo de uma política de bucket do Amazon S3 que restringe o acesso a um bucket específico, `awsexamplebucket1`, somente no VPC endpoint com o ID `vpce-1a2b3c4d`. Se o endpoint especificado não for usado, essa política negará todo acesso ao bucket. A condição `aws:SourceVpce` especifica o endpoint. A condição `aws:SourceVpce` não exige um nome do recurso da Amazon (ARN) para o recurso do endpoint da VPC, somente o ID correspondente. Para obter mais informações sobre o uso de condições em uma política, consulte [Exemplos de políticas de bucket usando chaves de condição](#).

⚠ 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 endpoint da VPC 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:SourceVpce`. 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. Veja a seguir um exemplo de política que nega acesso a `awsexamplebucket1` e seus objetos para qualquer pessoa de fora da VPC `vpc-111bbb22`. Se a VPC especificada não for usada, a política negará todo acesso ao bucket. Essa instrução não concede acesso ao bucket. Para conceder acesso, é necessário adicionar uma instrução `Allow` separada. A chave de condição `vpc-111bbb22` não exige um ARN para o recurso da VPC, somente o ID correspondente.

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 do Amazon S3

Com as políticas de bucket do Amazon S3, você pode proteger o acesso a objetos em seus buckets, para que somente usuários com as permissões apropriadas possam acessá-los. Você pode até mesmo impedir que usuários autenticados sem as permissões apropriadas acessem seus recursos do Amazon S3.

Esta seção apresenta exemplos de casos de uso típicos de políticas de bucket. Essas políticas de exemplo usam *DOC-EXAMPLE-BUCKET* como valor do recurso. Para testar essas políticas, substitua *user input placeholders* por suas informações (como o nome do seu bucket).

Para conceder ou negar permissões para um conjunto de objetos, você pode usar caracteres curinga (*) em nomes de recursos da Amazon (ARNs) e outros valores. Por exemplo, é possível controlar o acesso a grupos de objetos que começam com um [prefixo](#) ou terminam com uma extensão específica, como `.html`.

Para obter mais informações sobre a linguagem da política do AWS Identity and Access Management (IAM), consulte [Políticas e permissões no Amazon S3](#).

Note

Ao testar as permissões usando o console do Amazon S3, você deve conceder as permissões adicionais que o console exige: `s3:ListAllMyBuckets`, `s3:GetBucketLocation` e `s3:ListBucket`. Para acessar um exemplo de demonstração que concede permissões aos usuários e testa essas permissões usando o console, consulte [Controlar o acesso a um bucket com políticas de usuário](#).

Recursos adicionais para criar políticas de bucket incluem:

- Consulte uma lista de ações de políticas do IAM, recursos e chaves de condição que você pode usar ao criar uma política de bucket em [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço.
- Para obter orientação sobre como criar uma política do S3, consulte [Adicionar uma política de bucket usando o console do Amazon S3](#).
- Para solucionar erros com uma política, consulte [Solucionar erros de acesso negado \(403 proibido\) no Amazon S3](#).


Tópicos

- [Conceder permissão somente leitura a um usuário anônimo público](#)
- [Exigir criptografia](#)
- [Gerenciar buckets usando ACLs predefinidas](#)
- [Gerenciar o acesso a objetos com marcação de objetos](#)
- [Gerenciar o acesso a objetos usando chaves de condição globais](#)
- [Gerenciar o acesso com base em endereços IP específicos](#)
- [Gerenciar o acesso com base em solicitações HTTP ou HTTPS](#)
- [Gerenciar o acesso de usuários a pastas específicas](#)
- [Gerenciar o acesso para logs de acesso](#)
- [Gerenciar o acesso para uma OAI do Amazon CloudFront](#)
- [Gerenciar o acesso para a Lente de Armazenamento do Amazon S3](#)
- [Gerenciar permissões para o Inventário do S3, análises do S3 e relatórios do Inventário do S3](#)
- [Exigir MFA](#)
- [Impedir que os usuários excluam objetos](#)

Conceder permissão somente leitura a um usuário anônimo público


É possível usar as configurações de política para conceder acesso a usuários anônimos públicos, o que será útil se você estiver configurando o bucket como um site estático. Isso exige a desabilitação do Bloqueio de Acesso Público ao bucket. Consulte mais informações sobre como fazer isso e a política exigida em [Configuração de permissões para acesso ao site](#). Para saber como configurar políticas mais restritivas para o mesmo propósito, consulte [How can I grant public read access to some objects in my Amazon S3 bucket?](#) no Centro de Conhecimento da AWS.

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](#) 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 bloqueie 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](#) 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 bloqueie todo o acesso público aos seus 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)).

Exigir criptografia

Exigir SSE-KMS para todos os objetos gravados em um bucket

O exemplo de política a seguir exige que cada objeto gravado no bucket seja criptografado com criptografia do lado do servidor usando chaves do AWS Key Management Service (AWS KMS) (SSE-KMS). Se o objeto não estiver criptografado com a SSE-KMS, a solicitação será negada.

```
{
  "Version": "2012-10-17",
  "Id": "PutObjPolicy",
  "Statement": [{
    "Sid": "DenyObjectsThatAreNotSSEKMS",
    "Principal": "*",
    "Effect": "Deny",
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
    "Condition": {
      "Null": {
        "s3:x-amz-server-side-encryption-aws-kms-key-id": "true"
      }
    }
  }]
}
```

Exigir SSE-KMS com uma AWS KMS key específica para todos os objetos gravados em um bucket

O exemplo de política a seguir impede que objetos sejam gravados no bucket quando não estão criptografados com SSE-KMS usando um ID de chave do KMS específico. Mesmo que os objetos sejam criptografados com a SSE-KMS usando um cabeçalho por solicitação ou criptografia padrão de bucket, os objetos não serão gravados no bucket se não tiverem sido criptografados com a chave do KMS especificada. Substitua o ARN da chave do KMS usado neste exemplo pelo ARN da sua chave do KMS.

```
{
  "Version": "2012-10-17",
  "Id": "PutObjPolicy",
  "Statement": [{
    "Sid": "DenyObjectsThatAreNotSSEKMSWithSpecificKey",
    "Principal": "*",
    "Effect": "Deny",
    "Action": "s3:PutObject",
```

```

"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
"Condition": {
  "ArnNotEqualsIfExists": {
    "s3:x-amz-server-side-encryption-aws-kms-key-id": "arn:aws:kms:us-
east-2:111122223333:key/01234567-89ab-cdef-0123-456789abcdef"
  }
}
}]
}

```

Gerenciar buckets usando ACLs predefinidas

Conceder permissões a várias contas para fazer upload de objetos ou definir ACLs de objetos para acesso público

O exemplo de política a seguir concede as permissões `s3:PutObject` e `s3:PutObjectAcl` a várias Contas da AWS. Além disso, o exemplo de política exige que as solicitações para essas operações incluam a lista de controle de acesso (ACL) `public-read` predefinida. Para obter mais informações, consulte [Ações de políticas para o Amazon S3](#) e [Chaves de condição de política para o Amazon S3](#).

Warning

A ACL predefinida `public-read` permite que qualquer pessoa no mundo veja os objetos em seu bucket. 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](#). Se você quiser habilitar configurações de bloqueio de acesso público para hospedagem de sites estáticos, consulte [Tutorial: Configurar um site estático no Amazon S3](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AddPublicReadCannedAcl",
      "Effect": "Allow",
      "Principal": {

```

```

        "AWS": [
            "arn:aws:iam::111122223333:root",
            "arn:aws:iam::444455556666:root"
        ],
        "Action": [
            "s3:PutObject",
            "s3:PutObjectAcl"
        ],
        "Resource": "arn:aws:s3::DOC-EXAMPLE-BUCKET/*",
        "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": [
                    "public-read"
                ]
            }
        }
    ]
}

```

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 faça upload de objetos no bucket e, ao mesmo tempo, garantir controle total sobre os objetos carregados. Essa política concede a uma Conta da AWS específica (**111122223333**) a capacidade de carregar objetos se essa conta incluir a ACL pré-configurada `bucket-owner-full-control` no carregamento. A condição `StringEquals` na política especifica a chave de condição `s3:x-amz-acl` para expressar o requisito de ACL predefinida. Para ter mais informações, consulte [Chaves de condição de política para o Amazon S3](#).

```

{
  "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"}
  }
}
]
}

```

Gerenciar o acesso a objetos com marcação de objetos

Permitir que um usuário leia somente os objetos que têm a chave e o valor de uma etiqueta específica

A política de permissões a seguir limita a capacidade de leitura dos usuários para que leiam somente objetos que tenham a chave e o valor da tag `environment: production`. Essa política usa a chave de condição `s3:ExistingObjectTag` para especificar a chave e o valor da etiqueta.

```

{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Principal":{
        "AWS":"arn:aws:iam::111122223333:role/JohnDoe"
      },
      "Effect":"Allow",
      "Action":[
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource":"arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
      "Condition":{
        "StringEquals":{
          "s3:ExistingObjectTag/environment":"production"
        }
      }
    }
  ]
}

```

Restringir quais chaves de etiqueta de objeto os usuários podem adicionar

O exemplo de política a seguir concede ao usuário permissão para executar a ação `s3:PutObjectTagging`, que permite que o usuário adicione etiquetas a um objeto existente. A condição usa a chave de condição `s3:RequestObjectTagKeys` para especificar as chaves de

etiqueta permitidas, como `Owner` ou `CreationDate`. Para obter mais informações, consulte [Criar uma condição que testa vários valores de chave](#) no Guia do usuário do IAM.

A política garante que cada chave de etiqueta especificada na solicitação seja uma chave de etiqueta autorizada. O qualificador `ForAnyValue` na condição garante que pelo menos uma das chaves especificadas esteja presente na solicitação.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:role/JohnDoe"
        ]
      },
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ],
      "Condition": {
        "ForAnyValue:StringEquals": {
          "s3:RequestObjectTagKeys": [
            "Owner",
            "CreationDate"
          ]
        }
      }
    }
  ]
}
```

Exigir a chave e o valor de uma etiqueta específica ao permitir que os usuários adicionem etiquetas de objeto

O exemplo de política a seguir concede ao usuário permissão para executar a ação `s3:PutObjectTagging`, que permite que o usuário adicione etiquetas a um objeto existente. A condição requer que o usuário inclua uma chave de etiqueta específica (como *Project*) com o valor definido como *X*.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```

{"Principal":{"AWS":[
  "arn:aws:iam::111122223333:user/JohnDoe"
]},
"Effect": "Allow",
"Action": [
  "s3:PutObjectTagging"
],
"Resource": [
  "arn:aws:s3::DOC-EXAMPLE-BUCKET/*"
],
"Condition": {"StringEquals": {"s3:RequestObjectTag/Project": "X"}}
}
]
}

```

Permitir que um usuário adicione somente objetos com a chave e o valor de uma etiqueta específica

O exemplo de política a seguir concede ao usuário permissão para realizar a ação `s3:PutObject` para que possa adicionar objetos a um bucket. No entanto, a instrução `Condition` restringe as chaves e os valores de etiqueta permitidos nos objetos enviados. Neste exemplo, o usuário só pode adicionar objetos que tenham a chave de etiqueta específica (*Department*) com o valor definido como *Finance* no bucket.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Principal": {
      "AWS": [
        "arn:aws:iam::111122223333:user/JohnDoe"
      ]
    },
    "Effect": "Allow",
    "Action": [
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3::DOC-EXAMPLE-BUCKET/*"
    ],
    "Condition": {
      "StringEquals": {

```

```

        "s3:RequestObjectTag/Department": "Finance"
    }
}
}]
}

```

Gerenciar o acesso a objetos usando chaves de condição globais

As [chaves de condição globais](#) são chaves de condição com um prefixo `aws`. Os Serviços da AWS podem oferecer suporte a chaves de condição globais ou chaves específicas do serviço que incluem seu prefixo de serviço. É possível usar o elemento `Condition` de uma política JSON para comparar chaves em uma solicitação com os valores de chave especificados em sua política.

Restringir o acesso somente às entregas de logs de acesso ao servidor do Amazon S3

No exemplo de política de bucket a seguir, a chave de condição global [aws:SourceArn](#) é usada para comparar o [nome do recurso da Amazon \(ARN\)](#), fazendo uma solicitação de serviço a serviço com o ARN especificado na política. A chave de condição global `aws:SourceArn` é usada para impedir que o serviço Amazon S3 seja usado como um [representante confuso](#) durante transações entre os serviços. Somente o serviço Amazon S3 tem permissão para adicionar objetos ao bucket do Amazon S3.

Este exemplo de política de bucket concede permissões de `s3:PutObject` somente à entidade principal do serviço de registro em log (`logging.s3.amazonaws.com`).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutObjectS3ServerAccessLogsPolicy",
      "Principal": {
        "Service": "logging.s3.amazonaws.com"
      },
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET-logs/*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111111111111"
        },
        "ArnLike": {

```

```

        "aws:SourceArn": "arn:aws:s3:::EXAMPLE-SOURCE-BUCKET"
    }
}
},
{
    "Sid": "RestrictToS3ServerAccessLogs",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET-Logs/*",
    "Condition": {
        "ForAllValues:StringNotEquals": {
            "aws:PrincipalServiceNamesList": "logging.s3.amazonaws.com"
        }
    }
}
]
}

```

Permitir o acesso somente à sua organização

Se você quiser exigir que todas as [entidades principais do IAM](#) que acessam um recurso sejam de uma Conta da AWS em sua organização (incluindo a conta de gerenciamento do AWS Organizations), poderá usar a chave de condição global `aws:PrincipalOrgID`.

Para conceder ou restringir esse tipo de acesso, defina a condição `aws:PrincipalOrgID` e defina o valor como o [ID da sua organização](#) na política de bucket. O ID da organização é usado para controlar o acesso ao bucket. Quando você usa a condição `aws:PrincipalOrgID`, as permissões da política de bucket também são aplicadas a todas as novas contas adicionadas à organização.

Aqui está um exemplo de política de bucket baseada em recursos que você pode usar para conceder a entidades principais do IAM específicas em sua organização acesso direto ao bucket. Ao adicionar a chave de condição global `aws:PrincipalOrgID` à sua política de bucket, agora é necessário que a conta da entidade principal esteja em sua organização para obter acesso ao recurso. Mesmo que você acidentalmente especifique uma conta incorreta ao conceder acesso, a [chave de condição global `aws:PrincipalOrgID`](#) atuará como uma proteção adicional. Quando uma chave global é usada em uma política, ela impede que todas as entidades principais fora da organização especificada acessem o bucket do S3. Somente entidades principais de contas na organização listada podem obter acesso ao recurso.

```
{
```

```
"Version": "2012-10-17",
"Statement": [{
  "Sid": "AllowGetObject",
  "Principal": {
    "AWS": "*"
  },
  "Effect": "Allow",
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
  "Condition": {
    "StringEquals": {
      "aws:PrincipalOrgID": ["o-aa111bb222"]
    }
  }
}]
}
```

Gerenciar o acesso com base em endereços IP específicos

Restringir o acesso a endereços IP específicos

O exemplo a seguir impede que todos os usuários executem operações do Amazon S3 em objetos nos buckets especificados, a menos que a solicitação tenha origem no intervalo de endereços IP especificado.

Note

Ao restringir o acesso a um endereço IP específico, especifique quais endpoints da VPC, endereços IP de origem VPC ou endereços IP externos podem acessar o bucket do S3. Caso contrário, você poderá perder o acesso ao bucket se a sua política negar aos usuários a execução de qualquer operação do S3 em objetos no bucket sem as permissões adequadas já em vigor.

A instrução `Condition` dessa política identifica `192.0.2.0/24` como o intervalo de endereços IP IPv4 (Internet Protocol versão 4) 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. A chave de condição `aws:SourceIp` só pode ser usada para intervalos de endereços IP públicos. Para obter mais informações sobre essas chaves de condição, consulte [Chaves de condição de política para o Amazon S3](#). 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 `192.0.2.0/24` 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": "192.0.2.0/24"
        }
      }
    }
  ]
}
```

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. Isso ajudará a garantir que as políticas continuem funcionando à medida que você faz a transição para 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 permite

acesso aos endereços IP **192.0.2.1** e **2001:DB8:1234:5678::1** e nega acesso aos endereços **203.0.113.1** e **2001:DB8:1234:5678:ABCD::1**.

A chave de condição `aws:SourceIp` só pode ser usada para intervalos de endereços IP públicos. 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 `2001: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": [
            "192.0.2.0/24",
            "2001:DB8:1234:5678::/64"
          ]
        },
        "NotIpAddress": {
          "aws:SourceIp": [
            "203.0.113.0/24",
            "2001:DB8:1234:5678:ABCD::/80"
          ]
        }
      }
    }
  ]
}
```



```
}
  }
}
]
```

Gerenciar o acesso com base em solicitações HTTP ou HTTPS

Restringir o acesso somente a solicitações HTTPS

Se você quiser evitar que possíveis invasores manipulem o tráfego de rede, poderá usar HTTPS (TLS) para permitir somente conexões criptografadas e, ao mesmo tempo, restringir o acesso de solicitações HTTP ao seu bucket. Para determinar se a solicitação é HTTP ou HTTPS, use a chave de condição global [aws:SecureTransport](#) em sua política de bucket do S3. A chave de condição `aws:SecureTransport` verifica se uma solicitação foi enviada usando HTTP.

Se uma solicitação retornar `true`, isso significa que ela foi enviada por HTTPS. Se a solicitação retornar `false`, isso significa que ela foi enviada por HTTP. Depois, você pode permitir ou negar acesso ao bucket com base no esquema de solicitação desejado.

No exemplo a seguir, a política de bucket nega explicitamente as solicitações HTTP.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "RestrictToTLSRequestsOnly",
    "Action": "s3:*",
    "Effect": "Deny",
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    ],
    "Condition": {
      "Bool": {
        "aws:SecureTransport": "false"
      }
    },
    "Principal": "*"
  }]
}
```

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 em seu bucket chamado *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 de que a solicitação GET deve se originar de páginas específicas da Web. A política a seguir restringe as solicitações usando a condição `StringLike` com a chave de condição `aws:Referer`.

```
{
  "Version":"2012-10-17",
  "Id":"HTTP referer policy example",
  "Statement":[
    {
      "Sid":"Allow only GET requests originating from www.example.com and
example.com.",
      "Effect":"Allow",
      "Principal":"*",
      "Action":["s3:GetObject","s3:GetObjectVersion"],
      "Resource":"arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
      "Condition":{"
        "StringLike":{"aws:Referer":["http://www.example.com/*","http://example.com/
*"]}}
    }
  ]
}
```

Certifique-se de que os navegadores que você usa incluam 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 HTTP 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.

Gerenciar o acesso de usuários a pastas específicas

Conceder acesso a pastas específicas para os usuários

Suponha que você esteja tentando conceder a uma pasta específica para os usuários. Se o usuário do IAM e o bucket do S3 pertencerem à mesma Conta da AWS, você poderá usar uma política do IAM para conceder acesso a uma pasta de bucket específica para os usuários. Com essa abordagem, você não precisa atualizar sua política de bucket para conceder acesso. Você pode adicionar a política do IAM a um perfil do IAM para o qual vários usuários podem mudar.

Se a identidade do IAM e o bucket do S3 pertencerem a Contas da AWS diferentes, você deverá conceder acesso entre contas tanto na política do IAM quanto na política de bucket. Para obter mais informações sobre como conceder acesso entre contas, consulte [Proprietário do bucket concedendo permissões de bucket entre contas](#).

O exemplo de política de bucket a seguir concede a *JohnDoe* acesso total ao console somente na pasta dele (`home/JohnDoe/`). Ao criar uma pasta `home` e conceder as permissões apropriadas aos seus usuários, você pode fazer com que vários usuários compartilhem um único bucket. Essa política consiste em três instruções `Allow`:

- *AllowRootAndHomeListingOfCompanyBucket*: permite que o usuário (*JohnDoe*) liste objetos no nível raiz do bucket `DOC-EXAMPLE-BUCKET` e na pasta `home`. Essa instrução também permite que o usuário pesquise o prefixo `home/` usando o console.
- *AllowListingOfUserFolder*: permite que o usuário (*JohnDoe*) liste todos os objetos na pasta `home/JohnDoe/` e em quaisquer subpastas.
- *AllowAllS3ActionsInUserFolder*: permite que o usuário execute todas as ações do Amazon S3 ao conceder as permissões `Read`, `Write` e `Delete`. As permissões são limitadas à pasta inicial do proprietário do bucket.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```

{
  "Sid": "AllowRootAndHomeListingOfCompanyBucket",
  "Principal": {
    "AWS": [
      "arn:aws:iam::111122223333:user/JohnDoe"
    ]
  },
  "Effect": "Allow",
  "Action": ["s3:ListBucket"],
  "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET"],
  "Condition": {
    "StringEquals": {
      "s3:prefix": ["", "home/", "home/JohnDoe"],
      "s3:delimiter": ["/"]
    }
  }
},
{
  "Sid": "AllowListingOfUserFolder",
  "Principal": {
    "AWS": [
      "arn:aws:iam::111122223333:user/JohnDoe"
    ]
  },
  "Action": ["s3:ListBucket"],
  "Effect": "Allow",
  "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET"],
  "Condition": {
    "StringLike": {
      "s3:prefix": ["home/JohnDoe/*"]
    }
  }
},
{
  "Sid": "AllowAllS3ActionsInUserFolder",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::111122223333:user/JohnDoe"
    ]
  },
  "Action": ["s3:*"],
  "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET/home/JohnDoe/*"]
}

```

```
    ]
  }
```

Gerenciar o acesso para logs de acesso

Conceder acesso ao Application Load Balancer para habilitar logs de acesso

Ao habilitar os logs de acesso para o Application Load Balancer, você deve especificar o nome do bucket do S3 em que o balanceador de carga [armazenará os logs](#). O bucket deve ter uma [política anexada](#) que conceda ao Elastic Load Balancing permissão para gravar no bucket.

No exemplo a seguir, a política de bucket concede ao Elastic Load Balancing (ELB) permissão para gravar os logs de acesso no bucket:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": "arn:aws:iam::elb-account-id:root"
      },
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/prefix/AWSLogs/111122223333/*"
    }
  ]
}
```

Note

Substitua *elb-account-id* pelo ID da Conta da AWS do Elastic Load Balancing para sua Região da AWS. Para ver a lista de regiões do Elastic Load Balancing, consulte [Anexar uma política ao seu bucket do S3](#) no Guia do usuário do Elastic Load Balancing.

Se sua Região da AWS não aparecer na lista de regiões compatíveis do Elastic Load Balancing, use a política a seguir, que concede permissões ao serviço de entrega de logs especificado.

```
{
  "Version": "2012-10-17",
```

```
"Statement": [  
  {  
    "Principal": {  
      "Service": "logdelivery.elasticloadbalancing.amazonaws.com"  
    },  
    "Effect": "Allow",  
    "Action": "s3:PutObject",  
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/prefix/AWSLogs/111122223333/*"  
  }  
]
```

Depois, configure os [logs de acesso do Elastic Load Balancing](#) habilitando-os. Você pode [verificar as permissões de bucket](#) criando um arquivo de teste.

Gerenciar o acesso para uma OAI do Amazon CloudFront

Conceder permissão para uma OAI do Amazon CloudFront

O exemplo de política de bucket a seguir concede uma permissão de identidade do acesso de origem (OAI) do CloudFront para obter (ler) todos os objetos em seu bucket do 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 do 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 como usar políticas de bucket do S3 para conceder acesso a uma OAI do CloudFront, consulte [Migrar da identidade do acesso de origem \(OAI\) para o controle de acesso à origem \(OAC\)](#) 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 do acesso de origem) no console do CloudFront ou use [ListCloudFrontOriginAccessIdentities](#) na API do CloudFront.
- Substitua `DOC-EXAMPLE-BUCKET` pelo nome do seu bucket.

```
{  
  "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/*"
  }
]
```

Gerenciar o acesso para a Lente de Armazenamento do Amazon S3

Conceder permissões para a Lente de Armazenamento do Amazon S3

A Lente de Armazenamento do S3 agrega suas métricas e exibe as informações na seção Account snapshot (Snapshot da conta) na página Buckets do console do Amazon S3. A Lente de Armazenamento do S3 também oferece um painel interativo 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. O painel tem opções de detalhamento para gerar e visualizar insights no nível da organização, conta, Região da AWS, classe de armazenamento, bucket, prefixo ou grupo de Lente de Armazenamento. Você também pode enviar uma exportação diária de métricas em CSV ou no formato Parquet para um bucket do S3.

A Lente de Armazenamento do S3 pode exportar as métricas de uso de armazenamento agregadas 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. Ao configurar a exportação de métricas do S3 Storage Lens, você deve ter uma política para o bucket de destino. Para ter mais informações, consulte [Avaliação de sua atividade de armazenamento e uso com a Lente de Armazenamento do Amazon S3](#).

O exemplo de política de bucket a seguir concede ao Amazon S3 permissão para gravar objetos (solicitações PUT) 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:region-code:111122223333:storage-lens/storage-lens-dashboard-configuration-id"
        }
      }
    }
  ]
}

```

Ao configurar uma exportação de métricas da organização da Lente de Armazenamento do S3, use a modificação a seguir na instrução Resource da política de bucket anterior.

```

"Resource": "arn:aws:s3:::destination-bucket/destination-prefix/StorageLens/your-organization-id/*",

```

Gerenciar permissões para o Inventário do S3, análises do S3 e relatórios do Inventário do S3

Conceder permissões para o Inventário do S3 e análises do S3

O Inventário do S3 cria listas dos objetos em um bucket do, e a exportação da análise da classe de armazenamento do 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 ou arquivo de exportação da análise é gravado é chamado de bucket de destino. Ao configurar a exportação de um inventário ou uma análise, você deve criar uma política de bucket para o bucket de destino. Para obter mais informações, consulte [Inventário do Amazon S3](#) e [Análise do Amazon S3 – Análise de classe de armazenamento](#).

O exemplo da política de bucket a seguir concede permissão ao Amazon S3 para gravar objetos (solicitações PUT) da conta do bucket de origem para o bucket de destino. Use uma política de bucket como essa no bucket de destino ao configurar o Inventário do S3 e a exportação da análise do S3.

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

Controlar a criação da configuração de relatórios do Inventário S3

[Inventário do Amazon S3](#) cria listas dos objetos em um bucket do S3 e dos metadados de cada objeto. Com a permissão `s3:PutInventoryConfiguration`, o usuário tem autorização para criar uma configuração de inventário que inclui todos os campos de metadados de objetos que estão disponíveis por padrão e especificar o bucket de destino para armazenar o inventário. Um usuário com acesso de leitura aos objetos no bucket de destino pode acessar todos os campos de metadados do objeto que estão disponíveis no relatório de inventário. Para obter mais informações sobre os campos de metadados que estão disponíveis no S3 Inventory, consulte [Lista do Amazon S3 Inventory](#).

Para impedir que um usuário configure um relatório do Inventário S3, remova a permissão `s3:PutInventoryConfiguration` do usuário.

Alguns campos de metadados de objetos nas configurações de relatório do Inventário S3 são opcionais, o que significa que estão disponíveis por padrão, mas podem ser restritos quando você concede a permissão `s3:PutInventoryConfiguration` a um usuário. É possível controlar se os usuários podem incluir esses campos de metadados opcionais em seus relatórios usando a chave de condição `s3:InventoryAccessibleOptionalFields`. Consulte uma lista dos campos de metadados opcionais disponíveis no Inventário S3 em [OptionalFields](#) na Referência de API do Amazon Simple Storage Service.

Para conceder permissão ao usuário para criar uma configuração de inventário com campos de metadados opcionais específicos, use a chave de condição `s3:InventoryAccessibleOptionalFields` para refinar as condições na política de bucket.

O exemplo de política a seguir concede a um usuário (*Ana*) permissão para criar uma configuração de inventário condicionalmente. A condição `ForAllValues:StringEquals` na política usa a chave de condição `s3:InventoryAccessibleOptionalFields` para especificar os dois campos de metadados opcionais permitidos, a saber `Size` e `StorageClass`. Então, quando *Ana* está criando uma configuração de inventário, os únicos campos de metadados opcionais que ela pode incluir são `Size` e `StorageClass`.

```
{
  "Id": "InventoryConfigPolicy",
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowInventoryCreationConditionally",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:user/Ana"
    },
    "Action":
      "s3:PutInventoryConfiguration",
    "Resource":
      "arn:aws:s3::DOC-EXAMPLE-SOURCE-BUCKET",
    "Condition": {
      "ForAllValues:StringEquals": {
        "s3:InventoryAccessibleOptionalFields": [
          "Size",
          "StorageClass"
        ]
      }
    }
  ]
}
```

```

    }
  }
}
]
}

```

Para impedir que um usuário configure um relatório do Inventário S3 que inclua campos de metadados opcionais específicos, adicione uma instrução Deny explícita à política do bucket de origem. O exemplo de política de bucket a seguir impede que a usuária *Ana* crie uma configuração de inventário no bucket de origem *DOC-EXAMPLE-SOURCE-BUCKET* que inclua os campos de metadados opcionais `ObjectAccessControlList` ou `ObjectOwner`. A usuária *Ana* ainda pode criar uma configuração de inventário com outros campos de metadados opcionais.

```

{
  "Id": "InventoryConfigSomeFields",
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowInventoryCreation",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:user/Ana"
    },
    "Action": "s3:PutInventoryConfiguration",
    "Resource":
      "arn:aws:s3::DOC-EXAMPLE-SOURCE-BUCKET",
  },
  {
    "Sid": "DenyCertainInventoryFieldCreation",
    "Effect": "Deny",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:user/Ana"
    },
    "Action": "s3:PutInventoryConfiguration",
    "Resource":
      "arn:aws:s3::DOC-EXAMPLE-SOURCE-BUCKET",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "s3:InventoryAccessibleOptionalFields": [
          "ObjectOwner",
          "ObjectAccessControlList"
        ]
      }
    }
  }
]
}

```

```
    }  
  }  
}  
]  
}
```

Note

O uso da chave de condição `s3:InventoryAccessibleOptionalFields` nas políticas de bucket não afeta a entrega de relatórios de inventário com base nas configurações de inventário existentes.

Important

Recomendamos usar `ForAllValues` com efeito `Allow` ou `ForAnyValue` com efeito `Deny`, conforme mostrado nos exemplos anteriores.

Não use `ForAllValues` com efeito `Deny` nem `ForAnyValue` com efeito `Allow`, pois essas combinações podem ser excessivamente restritivas e bloquear a exclusão da configuração do inventário.

Para saber mais sobre os operadores do conjunto de condições `ForAllValues` e `ForAnyValue`, consulte [Multivalued context keys](#) no Guia do usuário do IAM.

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. MFA 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.

Para impor o requisito de MFA, use a chave de condição `aws:MultiFactorAuthAge` em uma política de bucket. 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 de condição `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 a seguir.

O exemplo de política negará qualquer operação do 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:*",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/taxdocuments/*",
      "Condition": { "Null": { "aws:MultiFactorAuthAge": true } }
    }
  ]
}
```

A condição `Null` no bloco `Condition` será avaliada como `true` se o valor da chave de condição `aws:MultiFactorAuthAge` for nulo, o que indica que as credenciais de segurança temporárias da solicitação foram criadas sem um dispositivo de MFA.

A política de bucket a seguir é uma extensão da política de bucket anterior. A política de bucket a seguir inclui duas instruções 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 validade da chave `aws:MultiFactorAuthAge`. A duração que você especifica com a chave `aws:MultiFactorAuthAge` não depende 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:*",
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/taxdocuments/*",
    "Condition": {"NumericGreaterThan": {"aws:MultiFactorAuthAge": 3600 }}
  },
  {
    "Sid": "",
    "Effect": "Allow",
    "Principal": "*",
    "Action": ["s3:GetObject"],
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
  }
]
}

```

Impedir que os usuários excluam objetos

Por padrão, os usuários não têm nenhuma permissão. No entanto, ao criar políticas, você pode conceder permissões que não pretendia conceder. Para evitar essas brechas de permissão, você pode elaborar uma política de acesso mais estrita adicionando uma negação explícita.

Para impedir explicitamente que usuários ou contas excluam objetos, é necessário adicionar as seguintes ações a uma política de bucket: as permissões `s3:DeleteObject`, `s3:DeleteObjectVersion` e `s3:PutLifecycleConfiguration`. Todas as três ações são necessárias porque é possível excluir objetos chamando explicitamente a API DELETE Object ou configurando seu ciclo de vida (consulte [Gerenciando seu ciclo de vida de armazenamento](#)) de modo que o Amazon S3 possa remover os objetos quando o respectivo ciclo de vida expirar.

No exemplo de política a seguir, você nega explicitamente as permissões de DELETE Object ao usuário Dave. Uma negação explícita sempre se sobrepõe a qualquer outra permissão concedida.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Dave"
      },
      "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/*"
  ]
}
]
```

Exemplos de políticas de bucket usando chaves de condição

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 de política para o Amazon S3](#). Consulte uma lista completa de ações, chaves de condição e recursos do Amazon S3 que você pode especificar em políticas em [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço.

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. Consulte uma lista completa de ações, chaves de condição e

recursos do Amazon S3 que você pode especificar em políticas em [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço.

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 \(ACL\)](#).

Tópicos

- [Exemplo 1: conceder a permissão `s3:PutObject` que exige os objetos armazenados usando criptografia do lado do servidor](#)
- [Exemplo 2: conceder a permissão `s3:PutObject` para copiar objetos com uma restrição na origem da cópia](#)
- [Exemplo 3: conceder acesso a uma versão específica de um objeto](#)
- [Exemplo 4: conceder permissões com base em tags de objetos](#)
- [Exemplo 5: restringir o acesso pelo ID da Conta da AWS do proprietário do bucket](#)
- [Exemplo 6: exigir uma versão mínima do TLS](#)

Exemplo 1: conceder a permissão `s3:PutObject` que exige os objetos armazenados usando criptografia do 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 AccountBadmin
```

Exemplo 2: conceder a 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/*"
    },
    {
      "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](#).

```

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 3: conceder 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 4: 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](#).

Exemplo 5: 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 (VPC) que restrinjam o acesso de usuários, perfis ou aplicações aos buckets do Amazon S3 pertencentes a um ID de Conta da AWS específico. 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
- [Limitar o acesso a buckets do Amazon S3 pertencentes a Contas da AWS específicas](#) no blog de armazenamento da AWS

Exemplo 6: 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",
      "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 permissão s3:GetObject com uma condição em um endereço IP](#)
- [Exemplo 2: obter uma lista de objetos em um bucket com um prefixo específico](#)
- [Exemplo 3: definir o número máximo de chaves](#)

Exemplo 1: conceder a permissão s3:GetObject com uma condição em um endereço IP

Você poderá conceder 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.*), 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": {
          "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:SourceVpce` 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](#).

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. Consulte uma lista completa de chaves de condição e tipos de recurso compatíveis com o Amazon S3 em [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço.

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 prefixos de nome de chave. O console do Amazon S3 usa prefixos 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 prefixos e delimitadores para filtrar permissões de acesso, consulte [Controlar o acesso a um bucket com políticas de usuário](#).

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 prefixos, não objetos em pastas. No entanto, na API do Amazon S3, se você organizar suas chaves de objeto usando tais prefixos, 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](#).

```
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.

Políticas baseadas em identidade do Amazon S3

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do Amazon S3. Eles também não podem executar tarefas usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) ou a API AWS. Para conceder aos usuários permissão para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis, e os usuários podem assumir os perfis.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documento de política JSON, consulte [Criação de políticas do IAM](#) no Guia do Usuário do IAM.

Consulte detalhes sobre ações e tipos de recurso definidos pelo Amazon S3, incluindo o formato dos ARNs para cada tipo de recurso, em [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço.

Tópicos

- [Melhores práticas de política](#)
- [Exemplos de políticas baseadas em identidade para o Amazon S3](#)
- [Controlar o acesso a um bucket com políticas de usuário](#)

Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do Amazon S3 em sua conta. Essas ações podem incorrer em custos para a Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas gerenciadas pela AWS e avance para as permissões de privilégio mínimo: para começar a conceder permissões a seus usuários e workloads, use as políticas gerenciadas pela AWS que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis na sua Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo cliente AWS específicas para seus casos de uso. Para obter mais informações, consulte [Políticas gerenciadas pela AWS](#) ou [Políticas gerenciadas pela AWS para perfis de trabalho](#) no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também

- conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte [Políticas e permissões no IAM](#) no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e atributos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso a ações de serviço, se elas forem usadas por meio de um AWS service (Serviço da AWS) específico, como o AWS CloudFormation. Para obter mais informações, consulte [Elementos de política JSON do IAM: condições](#) no Manual do usuário do IAM.
 - Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações acionáveis para ajudar você a criar políticas seguras e funcionais. Para obter mais informações, consulte [Validação de políticas do IAM Access Analyzer](#) no Guia do usuário do IAM.
 - Exigir autenticação multifator (MFA): se houver um cenário que exija usuários do IAM ou um usuário raiz em sua Conta da AWS, ative a MFA para obter segurança adicional. Para exigir a MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte [Configuração de acesso à API protegido por MFA](#) no Guia do usuário do IAM.

Para mais informações sobre as práticas recomendadas do IAM, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Exemplos de políticas baseadas em identidade para o Amazon S3

Esta seção mostra diversos exemplos de políticas baseadas em identidade do AWS Identity and Access Management (IAM) para controle de acesso ao Amazon S3. Consulte exemplos de políticas de bucket (políticas baseadas em recursos) em [Políticas de bucket para o Amazon S3](#). Para obter informações sobre a linguagem de políticas do IAM, consulte [Políticas e permissões no Amazon S3](#).

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](#).

Tópicos

- [Permitir que um usuário do IAM acesse um dos seus buckets](#)
- [Permitir que cada usuário do IAM acesse uma pasta em um bucket](#)
- [Permitir que um grupo tenha uma pasta compartilhada no Amazon S3](#)
- [Permitir que todos os seus usuários leiam objetos em uma parte de um bucket](#)
- [Permitir que um parceiro solte arquivos em uma parte específica de um bucket](#)
- [Restringir o acesso a buckets do Amazon S3 em uma Conta da AWS específica](#)
- [Restringir o acesso a buckets do Amazon S3 em sua unidade organizacional](#)
- [Restringir o acesso a buckets do Amazon S3 em sua unidade organizacional](#)
- [Conceder permissão para recuperar a configuração PublicAccessBlock para uma Conta da AWS](#)
- [Restringir a criação de buckets a uma região](#)

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](#).

```
{
  "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. Você pode criar pastas com nomes correspondentes aos seus nomes de usuário.

```
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](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowGroupToSeeBucketListInTheConsole",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:GetBucketLocation"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3::*"
    },
    {
      "Sid": "AllowRootLevelListingOfTheBucket",
      "Action": "s3:ListBucket",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::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:user-id}`. 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:user-id}/*"
    }
  ]
}

```

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. Também é desejável limitar o acesso de todos os usuários à própria pasta, mas não é possível criar pastas antes que alguém baixe sua aplicação e comece a jogar, pois você 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 permitem que o aplicativo acesse seu bucket e execute operações como criação de pastas específicas de usuário e upload de dados. Para obter mais informações sobre federação de identidades da Web, consulte [Sobre federação de identidades da Web](#) no Guia do usuário do IAM.

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 os 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

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-exempleou* 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 o 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.

Conceder permissão para recuperar a configuração `PublicAccessBlock` para uma Conta da AWS

O exemplo de política baseada em identidade a seguir concede a permissão `s3:GetAccountPublicAccessBlock` a um usuário. Para todas essas permissões, defina o valor `Resource` como `"*"`. Consulte mais informações sobre os ARNs de recursos em [Recursos de políticas para o Amazon S3](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Action": [
        "s3:GetAccountPublicAccessBlock"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```


Restringir a criação de buckets a uma região

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 conferir a 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, adicione uma instrução de negação explícita à política acima.

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 ele receba. A política a seguir 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. Anote 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](#).

```
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"}
```

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. Depois, 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](#)
- [Resumo da demonstração](#)
- [Preparar para a demonstração](#)
- [Etapa 1: Criar um bucket](#)
- [Etapa 2: criar usuários do IAM e um grupo](#)
- [Etapa 3: verificar se os usuários do IAM não têm nenhuma permissão](#)
- [Etapa 4: Conceder permissões no nível do grupo](#)
- [Etapa 5: conceder permissões específicas do usuário do IAM Alice](#)
- [Etapa 6: conceder permissões específicas do usuário do IAM Bob](#)
- [Etapa 7: Proteger a pasta Private](#)
- [Etapa 8: Limpar](#)
- [Recursos relacionados](#)

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 visualização dessas pastas e subpastas lógicas no bucket.

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 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 prefixos 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. O Amazon S3 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. O console exibe os prefixos 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. Na demonstração, 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 conceder gradualmente permissões como proprietário da Conta da AWS e testar permissões como um usuário do IAM, você precisará entrar e sair, sempre usando credenciais diferentes. Você pode fazer o teste em apenas um navegador, mas o processo é mais rápido se você usar dois navegadores diferentes. Use um navegador para se conectar ao AWS Management Console com as 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 a ele e carrega uma ou duas amostras de documentos 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. Criará um bucket.

Para obter instruções detalhadas, consulte [Criação de um bucket](#).

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](#) 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](#).

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. Para obter instruções passo a passo, consulte [Criar um usuário do IAM na sua Conta da AWS](#) no Guia do usuário do IAM.

Crie também um grupo administrativo chamado Consultants. Depois, adicione os dois usuários ao grupo. Para obter instruções passo a passo, consulte [Criação de grupos de usuários do IAM](#).

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.

Consulte instruções detalhadas de como criar um usuário do IAM em [Criar um usuário do IAM na sua Conta da AWS](#) no Guia do usuário do IAM. Ao criar usuários seguindo este passo a passo, selecione Acesso ao AWS Management Console e desmarque [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](#)), 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 mensagem do console que informa 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 Políticas (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 Service do Amazon S3, que retorna uma lista de todos os buckets pertencentes ao 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`.

Consulte instruções detalhadas de como anexar uma política gerenciada em [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](#)), 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, examine o fluxo mais detalhadamente.

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 `prefix` 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.

Consulte instruções detalhadas em [Edição de 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](#)), 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 Alice possa listar o conteúdo da pasta Development, você deve aplicar uma política à usuária 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 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. Selecione o 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](#)), 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 Uses (Usuários).
 - b. Selecione o 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"],
    "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](#)), 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 incluam 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](#).

- 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":["s3:GetObject", "s3:PutObject"],
      "Effect":"Allow",
      "Resource":["arn:aws:s3:::companybucket/Development/*"]
    },
    {
      "Sid": "ExplicitlyDenyAnyRequestsForAllOtherFoldersExceptDevelopment",
      "Action": ["s3:ListBucket"],
      "Effect": "Deny",
      "Resource": ["arn:aws:s3:::companybucket"],
      "Condition":{"StringNotLike": {"s3:prefix":["Development/*", ""] },
                  "Null"           : {"s3:prefix":false }
      }
    }
  ]
}

```

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](#).

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 acesse o conteúdo da pasta Private. Como garantir que você não vai conceder acidentalmente uma permissão ao usuário para a pasta Private? 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.

Consulte instruções em [Edição de políticas gerenciadas pelo cliente](#) no Guia do usuário do IAM. Ao seguir as instruções, não se esqueça de seguir as orientações para aplicar alterações em todas as entidades principais às quais a política está anexada.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid":
"AllowGroupToSeeBucketListAndAlsoAllowGetBucketLocationRequiredForListBucket",
      "Action": ["s3:ListAllMyBuckets", "s3:GetBucketLocation"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::*"]
    },
    {
      "Sid": "AllowRootLevelListingOfCompanyBucket",
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::companybucket"],
      "Condition":{
        "StringEquals":{"s3:prefix":[""]}
      }
    },
    {
      "Sid": "RequireFolderStyleList",
      "Action": ["s3:ListBucket"],
      "Effect": "Deny",
      "Resource": ["arn:aws:s3:::*"],
      "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. Consulte instruções detalhadas em [Exclusão de um usuário do IAM](#) no Guia do usuário do IAM.

Para garantir que você não seja mais cobrado pelo armazenamento, exclua também os objetos e o bucket que criou neste exercício.

Recursos relacionados

- [Como gerenciar políticas do IAM](#) no Guia do usuário do IAM

Demonstrações que usam políticas para gerenciar o acesso a 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 (AWS CLI) quanto o AWS Tools for Windows PowerShell.

- [Exemplo 1: Proprietário do bucket que concede permissões do bucket aos usuários](#)

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](#)

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. Quando o proprietário do bucket escreve uma política de bucket concedendo acesso aos objetos, ela 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](#)

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 permissão entre contas a objetos que não possui](#)

Após o recebimento das 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 permitida (consulte [Delegação de permissão](#)). 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 de linha de comando, a AWS CLI e o AWS Tools for Windows PowerShell para que não seja necessário escrever nenhum código. Para testar permissões, você deve configurar uma dessas ferramentas. Para ter mais informações, consulte [Configurar as ferramentas para as demonstrações](#).

Além disso, ao criar recursos, esses exemplos não usam as credenciais de usuário-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 as credenciais do usuário raiz da sua Conta da AWS para fazer solicitações. Em vez disso, crie um perfil ou usuário do IAM, conceda acesso total a esse perfil ou usuário, depois use as credenciais desse perfil ou usuário para fazer solicitações. Essa entidade é conhecida como perfil ou usuário administrador. Para obter mais informações, acesse [Credenciais de Usuário raiz da conta da AWS e identidades do IAM](#) na Referência geral da AWS e [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.

Para fazer login no AWS Management Console usando as credenciais de usuário, é necessário 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

Os exemplos introdutórios (consulte [Demonstrações que usam políticas para gerenciar o acesso a recursos do Amazon S3](#)) usam o AWS Management Console para criar recursos e conceder permissões. Para testar permissões, os exemplos usam ferramentas de linha de comando, a AWS Command Line Interface (AWS CLI) e o AWS Tools for Windows PowerShell para que não seja necessário escrever nenhum código. 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:

[Install or update to the latest version of the AWS Command Line Interface](#)

[Get started with the AWS Command Line Interface](#)

2. Defina o perfil padrão.

As credenciais de usuário são armazenadas no arquivo de configuração da AWS CLI. Crie um perfil padrão no arquivo de configuração usando as credenciais da sua Conta da AWS. Consulte instruções sobre como encontrar e editar o arquivo de configuração da AWS CLI em [Configuration and credential file settings](#).

```
[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 `help`.

```
aws help
```

- Para obter uma lista dos buckets na conta configurada, use o comando `aws s3 ls`.

```
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. Esses perfis têm os nomes de `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 o perfil padrão.

```
$ export AWS_DEFAULT_PROFILE=AccountAdmin
```

Como configurar o AWS Tools for Windows PowerShell

1. Faça download e configure a AWS Tools for Windows PowerShell. Consulte instruções em [Installing the AWS Tools for Windows PowerShell](#) no Guia do usuário do AWS Tools for Windows PowerShell.

Note

Para carregar o módulo do AWS Tools for Windows PowerShell, é necessário habilitar a execução de script do PowerShell. Consulte mais informações em [Enable Script Execution](#) no Guia do usuário do AWS Tools for Windows PowerShell.

2. Para essas demonstrações, 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.

- Para recuperar uma lista de comandos disponíveis que podem ser usados para operações do Amazon S3, execute o comando `Get-Command`.

```
Get-Command -module awspowershell -noun s3* -StoredCredentials string
```

- Para recuperar uma lista de objetos em um bucket, execute o comando `Get-S3Object`.

```
Get-S3Object -BucketName bucketname -StoredCredentials string
```

Consulte uma lista de comandos em [AWS Tools for PowerShell Cmdlet Reference](#).

Agora está tudo pronto para você testar as demonstrações. Siga os links fornecidos no início de cada seção.

Exemplo 1: Proprietário do bucket que concede permissões do bucket aos usuários

⚠ Important

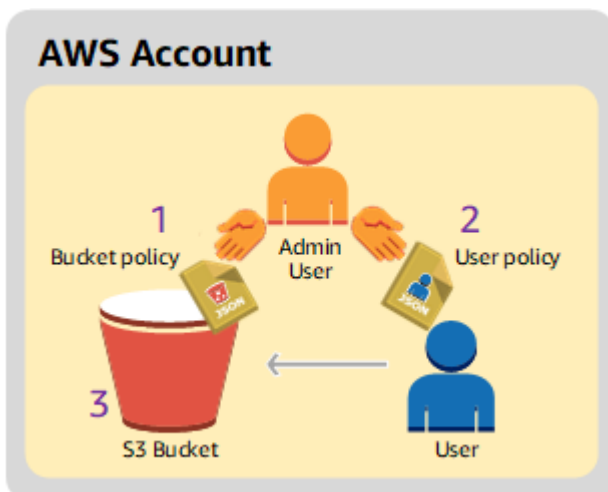
Conceder permissões a perfis do IAM é uma prática mais adequada do que conceder permissões a usuários individuais. Consulte mais informações sobre como conceder permissões a perfis do IAM em [Noções básicas de permissões entre contas e do uso de perfis do IAM](#).

Tópicos

- [Preparar para a demonstração](#)
- [Etapa 1: criar recursos na Conta A e conceder permissões](#)
- [Etapa 2: Testar permissões](#)

Nessa demonstração, uma Conta da AWS possui um bucket e a conta inclui um usuário do IAM. Por padrão, o usuário não tem permissões. 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 do usuário raiz da conta, você criará um usuário administrador (consulte [Sobre o uso de um usuário administrador para criar recursos e conceder permissões](#)). Fazemos referência à Conta da AWS e ao usuário administrador conforme indicado na tabela a seguir.

ID da conta	Conta referida como	Usuário administrador na conta
<i>1111-1111-1111</i>	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 (AWS CLI), e o AWS Tools for Windows PowerShell. Portanto, você não precisa escrever nenhum código.

Preparar 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 da AWS, se necessário. Nós nos referimos a essa conta como conta A.
 - i. Acesse <https://aws.amazon.com/s3> e selecione Criar 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. 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.

Consulte instruções em [Criar um usuário do IAM na sua Conta da AWS](#) no Guia do usuário do IAM.
 - ii. Conceda privilégios de administrador a AccountAdmin anexando uma política de usuário com pleno acesso.

Consulte instruções em [Gerenciar políticas do IAM](#) no Guia do usuário do IAM.
 - iii. Anote o URL de login de usuário do IAM de AccountAdmin. Você precisará usar esse URL para fazer login no AWS Management Console. Consulte mais informações sobre onde encontrar o URL de login em [Sign in to the AWS Management Console as an IAM user](#) no Guia do usuário do IAM. Anote o URL de cada uma das contas.
2. Configure a AWS CLI ou o AWS Tools for Windows PowerShell. Salve as credenciais de usuário do administrador da seguinte maneira:
 - 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 da sessão como AccountAdmin.

Para obter instruções, consulte [Configurar as ferramentas para as demonstrações](#).

Etapa 1: criar recursos 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 execute o seguinte:

1. Crie recursos de um bucket e um usuário do IAM
 - a. No console do Amazon S3, crie um bucket. Anote a Região da AWS na qual você criou o bucket. Para obter instruções, consulte [Criação de um bucket](#).
 - b. No [console do IAM](#), faça o seguinte:
 - i. Crie um usuário chamado Dave.

Para obter instruções passo a passo, consulte [Criação de usuários do IAM \(console\)](#) no Guia do usuário do IAM.

- ii. Anote as credenciais de UserDave.
- iii. Anote o nome do recurso da Amazon (ARN) do usuário Dave. No [console do IAM](#), selecione o usuário para que a guia Resumo fornece o ARN do usuário.

2. Conceder 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 e permissões no Amazon S3](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
      },
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucket"
      ]
    }
  ]
}
```



```

    "Resource": [
      "arn:aws:s3:::awsexamplebucket1"
    ],
  },
  {
    "Sid": "statement2",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
    },
    "Action": [
      "s3:GetObject"
    ],
    "Resource": [
      "arn:aws:s3:::awsexamplebucket1/*"
    ]
  }
]
}

```

- b. Crie uma política inline para o usuário Dave usando as seguintes políticas. A política concede a Dave a permissão `s3:PutObject`. Você precisa atualizar a política, fornecendo o nome de seu bucket.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PermissionForObjectOperations",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::awsexamplebucket1/*"
      ]
    }
  ]
}

```

Consulte instruções em [Gerenciamento de políticas do IAM](#) 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 as permissões usando a AWS CLI

1. Atualize o arquivo de configuração da AWS CLI adicionando o perfil `UserDaveAccountA` a seguir. Para ter mais informações, consulte [Configurar as ferramentas para as demonstrações](#).

```
[profile UserDaveAccountA]
aws_access_key_id = access-key
aws_secret_access_key = secret-access-key
region = us-east-1
```

2. Verifique se Dave pode executar as operações conforme concedido na política de usuário. 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 estiver na raiz da unidade C: de um computador Windows, você deve especificar `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 as permissões usando o AWS Tools for Windows PowerShell

1. Armazene as credenciais de Dave como `AccountADave`. Depois, use essas credenciais para PUT e GET um objeto.

```
set-awscredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas
AccountADave
```

2. 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

Important

Conceder permissões a perfis do IAM é uma prática melhor do que conceder permissões a usuários individuais. Para aprender a fazer isso, consulte [Noções básicas de permissões entre contas e do uso de perfis do IAM](#).

Tópicos

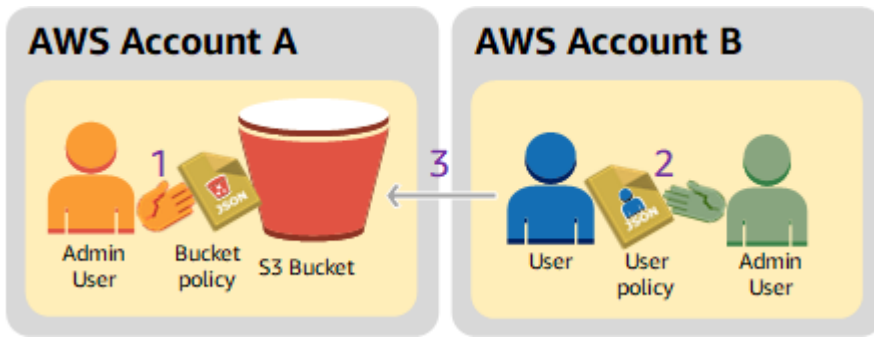
- [Preparar para a demonstração](#)
- [Etapa 1: Fazer as tarefas da Conta A](#)
- [Etapa 2: Fazer as tarefas da Conta B](#)
- [Etapa 3: \(opcional\) tentar negação explícita](#)
- [Etapa 4: limpar](#)

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. No entanto, o usuário ainda precisará ter permissão da conta principal, 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 principal, 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 (consulte [Sobre o uso de um usuário administrador para criar recursos e conceder permissões](#)), não usamos as credenciais do usuário-raiz nessa demonstração. 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
<i>1111-1111-1111</i>	Conta A	AccountAdmin
<i>2222-2222-2222</i>	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.

Preparar 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.
 - b. Usando as credenciais da Conta A, faça login no [console do IAM](#) para criar o usuário administrador:
 - i. Crie um usuário **AccountAdmin** e anote as credenciais de segurança. Para obter instruções, consulte [Criar um usuário do IAM na sua Conta da AWS](#) no Guia do usuário do IAM.
 - ii. Conceda privilégios de administrador a 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 o URL de login de usuário do IAM indicado no 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 **AccountBadmín**.
2. Configure a AWS Command Line Interface (AWS CLI) ou o AWS Tools for Windows PowerShell. Salve as credenciais de usuário do administrador da seguinte maneira:
 - Se estiver usando a AWS CLI, crie dois perfis, AccountAdmin e AccountBadmín, no arquivo de configuração.
 - Se estiver usando o AWS Tools for Windows PowerShell, armazene as credenciais da sessão como AccountAdmin e AccountBadmín.

Para obter instruções, consulte [Configurar as ferramentas para as demonstrações](#).

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 ter mais informações, consulte [Configurar as ferramentas para as demonstrações](#).

- a. Adicione perfis no arquivo de credenciais da AWS CLI para cada um dos usuários administradores, AccountAdmin e AccountBadmín, nas duas contas.

```
[AccountAdmin]
aws_access_key_id = access-key-ID
aws_secret_access_key = secret-access-key
region = us-east-1

[AccountBadmín]
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, execute o comando a seguir.

```
set-awscredentials -AccessKey AcctA-access-key-ID -SecretKey AcctA-secret-access-key -storeas AccountAdmin
set-awscredentials -AccessKey AcctB-access-key-ID -SecretKey AcctB-secret-access-key -storeas AccountBadmín
```

Etapa 1: Fazer as tarefas da Conta A

Etapa 1.1: fazer login no AWS Management Console

Usando o URL de login do usuário do IAM da Conta A, primeiro faça login no AWS Management Console como o usuário AccountAdmin. Esse usuário criará um bucket e anexará uma política a ele.

Etapa 1.2: Criar um bucket

1. No console do Amazon S3, crie um bucket. Esse exercício supõe que o bucket é criado na Região da AWS Leste dos EUA (Norte da Virgínia) e que o nome dele é *DOC-EXAMPLE-BUCKET*.

Para obter instruções, consulte [Criação de um bucket](#).

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](#).

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` à Conta B, presumindo que você ainda está usando as credenciais do usuário `AccountAdmin` no console.

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](#).

```
{
  "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 (ou seja, do usuário administrador) pode executar as operações.

- Fazer a verificação usando a AWS CLI

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET --profile AccountBadmin
aws s3api get-bucket-lifecycle-configuration --bucket DOC-EXAMPLE-BUCKET --
profile AccountBadmin
```

- Fazer a verificação usando o 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 o 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 chamado **Dave**.

Para obter instruções, consulte [Criação de usuários do IAM \(console\)](#) 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.

Presume-se que você fez login 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"  
      ]  
    }  
  ]  
}
```

```
}
```

Consulte instruções em [Gerenciar políticas do IAM](#) 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 as permissões 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](#)

```
[profile UserDave]
aws_access_key_id = access-key
aws_secret_access_key = secret-access-key
region = us-east-1
```

2. No prompt de comando, insira o comando da AWS CLI a seguir para verificar se agora Dave consegue obter uma lista de objetos do `UserDave` pertencente à Conta A. Observe que o comando especifica o perfil *DOC-EXAMPLE-BUCKET*.

```
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 de `get-bucket-lifecycle` a seguir), o Amazon S3 retornará uma negação da permissão.

```
aws s3api get-bucket-lifecycle-configuration --bucket DOC-EXAMPLE-BUCKET --profile
UserDave
```

Testar as permissões 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 `get-s3bucketlifecycleconfiguration` a seguir), o Amazon S3 retornará uma negação da permissão.

```
get-s3bucketlifecycleconfiguration -BucketName DOC-EXAMPLE-BUCKET -  
StoredCredentials AccountBDave
```

Etapa 3: (opcional) tentar negação explícita

Você pode ter permissões concedidas usando uma lista de controle de acesso (ACL), uma política de bucket ou uma política de usuário. Entretanto, se houver uma negação explícita definida por uma política de bucket ou por uma política de usuário, ela terá precedência sobre qualquer outra permissão. Para testar, atualize a política de bucket e negue explicitamente a permissão `s3:ListBucket` para a Conta B. A política também concede a permissão `s3:ListBucket`. No entanto, 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, o acesso será negado.

- Usando a AWS CLI, execute o seguinte comando:

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET --profile AccountBadmin
```

- Usando o AWS Tools for Windows PowerShell, execute o seguinte comando:

```
get-s3object -BucketName DOC-EXAMPLE-BUCKET -StoredCredentials AccountBDave
```

Etapa 4: limpar

1. Depois de terminar os testes, você pode fazer o seguinte para concluir 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 AccountAdmin.
2. Faça login no [console do IAM](#) usando as credenciais da Conta B. Exclua o usuário AccountBadmin. Consulte instruções detalhadas em [Exclusão de um usuário do IAM](#) no Guia do usuário do IAM.

Exemplo 3: O proprietário do bucket concede permissões para objetos que não possui

Important

Conceder permissões a perfis do IAM é uma prática melhor do que conceder permissões a usuários individuais. Para aprender a fazer isso, consulte [Noções básicas de permissões entre contas e do uso de perfis do IAM](#).

Tópicos

- [Etapa 0: Preparar-se para a demonstração](#)
- [Etapa 1: Realizar as tarefas da Conta A](#)
- [Etapa 2: Realizar as tarefas da Conta B](#)
- [Etapa 3: Testar permissões](#)
- [Etapa 4: limpar](#)

O cenário desse exemplo é que o proprietário do bucket deseja conceder permissão para acesso a objetos, mas o proprietário do bucket não possui todos os objetos 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 dos objetos no bucket, primeiro o proprietário dos objetos deverá conceder permissão ao proprietário do bucket usando uma lista de controle de acesso (ACL) de objetos. Depois, o proprietário do bucket poderá conceder permissões a um objeto que ele não possui. Para ter mais informações, consulte [Propriedade de bucket e objeto do Amazon S3](#).

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,

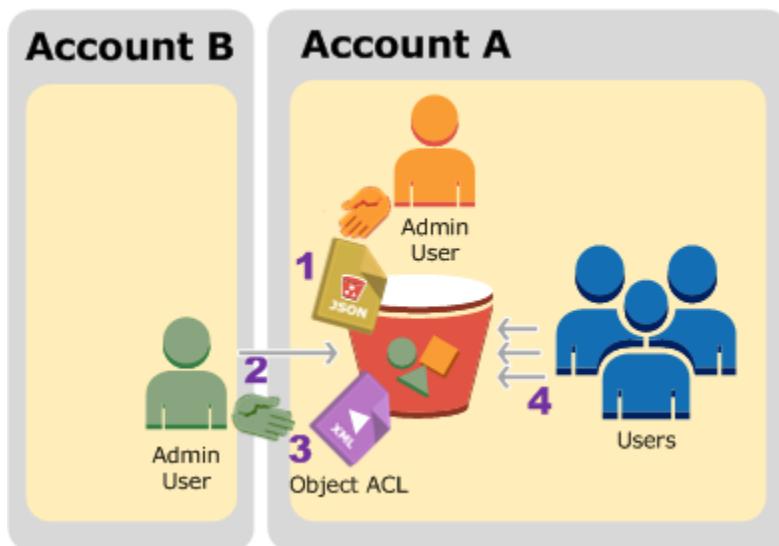
inclusive de objetos gravados por outra Conta da AWS. Essa abordagem resolve o problema de objetos que não pertencem ao proprietário do bucket. Em seguida, é possível delegar permissão aos usuários em sua própria conta ou para outras Contas da AWS.

Note

A Propriedade de objetos do S3 é uma configuração no nível do bucket do Amazon S3 que você pode usar para controlar a propriedade de objetos carregados no bucket e para desabilitar ou habilitar as ACLs. Por padrão, a Propriedade de Objetos está definida com a configuração Imposto pelo proprietário do bucket e todas as ACLs estão desabilitadas. Quando as ACLs são desabilitadas, o proprietário do bucket possui todos os objetos do bucket e gerencia o acesso a eles exclusivamente usando políticas de gerenciamento de acesso.

A maioria dos casos de uso modernos no Amazon S3 não exige mais o uso de ACLs. Recomendamos manter as ACLs desabilitadas, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com as ACLs desabilitadas, é possível usar políticas para controlar o acesso a todos os objetos no bucket, independentemente de quem carregou os objetos para o bucket. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

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. Veja a seguir 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 do usuário raiz da conta, conforme as diretrizes recomendadas do IAM. Para ter mais informações, consulte [Sobre o uso de um usuário administrador para criar recursos e conceder permissões](#). 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
<i>1111-1111-1111</i>	Conta A	AccountAdmin
<i>2222-2222-2222</i>	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.
 - 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 do usuário. Para obter mais informações sobre como adicionar usuários, consulte [Criar um usuário do IAM na sua Conta da AWS](#) no Guia do usuário do IAM.
 - Conceda permissões de administrador a AccountAdmin anexando uma política de usuário com pleno acesso. Consulte instruções em [Gerenciar políticas do IAM](#) no Guia do usuário do IAM.
 - No [console do IAM](#), anote o URL de login de usuário do IAM indicado no Painel. 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](#).

Etapa 1: Realizar 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 do usuário do IAM da Conta A, faça login no AWS Management Console como o 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. Esse exercício supõe que o bucket foi criado na Região da AWS Leste dos EUA (Norte da Virgínia) e que o nome dele é **DOC-EXAMPLE-BUCKET1**.

Para obter instruções, consulte [Criação de um bucket](#).

2. No [console do IAM](#), crie um usuário chamado **Dave**.

Para obter instruções passo a passo, consulte [Criação de usuários do IAM \(console\)](#) no Guia do usuário do IAM.

3. Anote as credenciais do usuário 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](#). 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: Realizar 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 pode fazer o seguinte:

- Carregar 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.

Usando a AWS CLI

1. Usando o comando `put-object` da AWS CLI, carregue 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, especifique `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 [Find the canonical user ID for your Conta da AWS](#) no Guia de referência de gerenciamento de contas da AWS.

```
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. Carregue um objeto usando o comando `Write-S3Object`.

```
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.

Usando a 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](#).

```
[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 -stores 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: limpar

1. Depois de terminar os testes, você pode fazer o seguinte para concluir 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 AccountAdmin. Consulte instruções detalhadas em [Exclusão de um usuário do IAM](#) no Guia do usuário do IAM.
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 permissão entre contas a objetos que não possui

Tópicos

- [Noções básicas de permissões entre contas e do uso de perfis do IAM](#)
- [Etapa 0: Preparar-se para a demonstração](#)
- [Etapa 1: Realizar as tarefas da Conta A](#)
- [Etapa 2: Realizar as tarefas da Conta B](#)

- [Etapa 3: realizar as tarefas da Conta C](#)
- [Etapa 4: limpar](#)
- [Recursos relacionados](#)

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. Essa abordagem resolve o problema de 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. Para que o proprietário do bucket possa conceder permissões em objetos que não possui, o proprietário do objeto deve primeiro conceder permissão ao proprietário do bucket. O proprietário do objeto é a Conta da AWS que criou os objetos. Depois, o proprietário do bucket pode delegar essas permissões.
- O proprietário do bucket pode delegar permissões aos usuários em sua própria conta (consulte [Exemplo 3: O proprietário do bucket concede permissões para objetos que não possui](#)). No entanto, a conta do proprietário do bucket não pode delegar permissões a outras Contas da AWS porque a delegação entre contas não é permitida.

Nesse cenário, o proprietário do bucket pode criar um perfil do AWS Identity and Access Management (IAM) com permissão para acessar objetos e, depois, conceder permissão a outra Conta da AWS para assumir o perfil temporariamente, permitindo que ela acesse objetos no bucket.

Note

A Propriedade de objetos do S3 é uma configuração no nível do bucket do Amazon S3 que você pode usar para controlar a propriedade de objetos carregados no bucket e para desabilitar ou habilitar as ACLs. Por padrão, a Propriedade de Objetos está definida com

a configuração Imposto pelo proprietário do bucket e todas as ACLs estão desabilitadas. Quando as ACLs são desabilitadas, o proprietário do bucket possui todos os objetos do bucket e gerencia o acesso a eles exclusivamente usando políticas de gerenciamento de acesso.

A maioria dos casos de uso modernos no Amazon S3 não exige mais o uso de ACLs. Recomendamos manter as ACLs desabilitadas, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com as ACLs desabilitadas, é possível usar políticas para controlar o acesso a todos os objetos no bucket, independentemente de quem carregou os objetos para o bucket. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

Noções básicas de permissões entre contas e do uso de perfis 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 um perfil do IAM para delegar temporariamente o acesso entre contas a objetos aos usuários em uma outra Conta da AWS, a Conta C. Cada perfil do IAM que você cria tem estas duas políticas anexadas a ela:

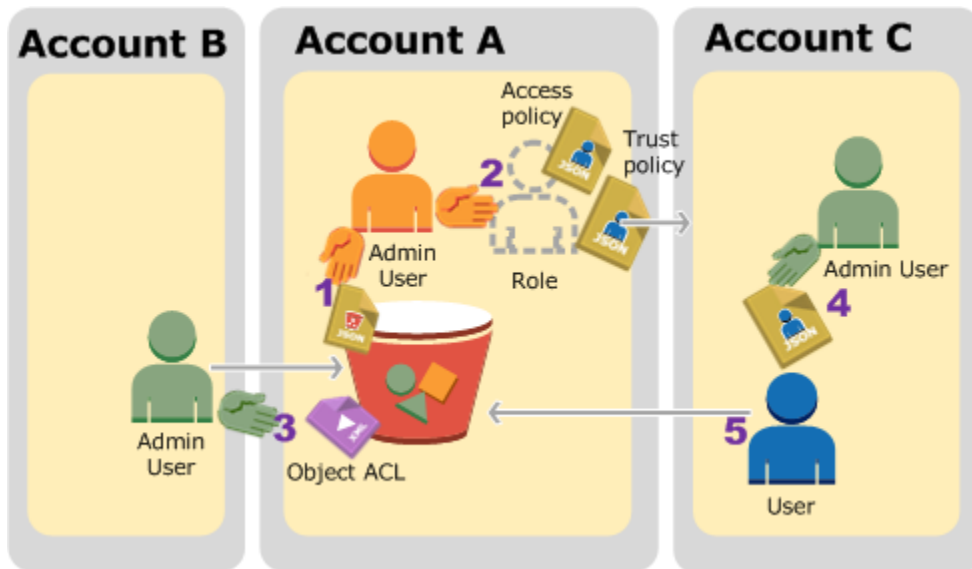
- Uma política de confiança que identifica outra Conta da AWS que pode assumir a função.
- Uma política de acesso que define quais permissões - por exemplo, `s3:GetObject` - são permitidas quando alguém assume a função. Para obter uma lista de permissões que você pode especificar em uma política, consulte [Ações de políticas para o Amazon S3](#).

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 funções do IAM, consulte [Perfis do IAM](#) no Guia do usuário do IAM.

Veja a seguir 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 (consulte [Sobre o uso de um usuário administrador para criar recursos e conceder permissões](#)), não usamos as credenciais do Usuário raiz da conta da AWS nesta demonstração. 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
<i>1111-1111-1111</i>	Conta A	AccountAdmin
<i>2222-2222-2222</i>	Conta B	AccountBAdmin
<i>3333-3333-3333</i>	Conta C	AccountCAdmin

Etapa 0: Preparar-se para a demonstração

Note

É recomendado abrir um editor de texto e anotar 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 (ARNs) dos perfis e usuários do IAM.


1. Você precisa ter três Contas da AWS, e cada uma delas deve ter um usuário administrador, conforme exibido na tabela da 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.
 - 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 [Criar um usuário do IAM na sua Conta da AWS](#) no Guia do usuário do IAM.
 - Conceda privilégios de administrador a AccountAdmin anexando uma política de usuário com pleno acesso. Consulte instruções em [Gerenciar políticas do IAM](#) no Guia do usuário do IAM.
 - No console do IAM, anote o URL de login de usuário do IAM indicado no Painel. Os usuários nessa conta devem usar esse URL para fazer login no AWS Management

Console. Consulte mais informações em [Sign in to the AWS Management Console as an IAM user](#) 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 Acesso à sua Conta da AWS, a coluna Conta inclui um identificador longo, como `c1daexampleaaf850ea79cf0430f33d72579fd1611c97f7ded193374c0b163b6`. Esse é seu ID de usuário canônico.
3. Ao criar uma política do bucket, você precisará das seguintes informações. Anote estes 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.

- ARN do usuário para o administrador da Conta B: é possível encontrar o ARN do usuário no [console do IAM](#). Selecione o usuário e veja o respectivo ARN na guia Resumo.

Na política do bucket, você concede permissão a AccountBadmIn para carregar objetos e especifica o usuário com o 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. Salve as credenciais de usuário do administrador da seguinte maneira:
 - Se estiver usando a AWS CLI, crie os perfis `AccountAdmin` e `AccountAdmin` no arquivo de configuração.
 - Se estiver usando o AWS Tools for Windows PowerShell, armazene as credenciais da sessão como `AccountAdmin` e `AccountAdmin`.

Para obter instruções, consulte [Configurar as ferramentas para as demonstrações](#).

Etapa 1: Realizar as tarefas da Conta A

Neste exemplo, a Conta A é o proprietário do bucket. Assim, o usuário `AccountAdmin` na Conta A realizará estas ações:

- Criará um bucket.
- Anexará uma política de bucket que conceda ao administrador da Conta B uma permissão para carregar objetos.
- Criará um perfil do IAM que conceda permissão à Conta C para assumir o perfil para que seja possível acessar os objetos no bucket.

Etapa 1.1: fazer login no AWS Management Console

Usando o URL de login do usuário do IAM da Conta A, primeiro faça login no AWS Management Console como o 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 *DOC-EXAMPLE-BUCKET1*.

Para obter instruções, consulte [Criação de um bucket](#).

2. Anexe a política de bucket a seguir. A política concede ao administrador da Conta B uma permissão condicional para carregar objetos.

Atualize a política fornecendo seus próprios valores para *DOC-EXAMPLE-BUCKET1*, *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::DOC-EXAMPLE-BUCKET1/*"
    },
    {
      "Sid": "112",
      "Effect": "Deny",
      "Principal": {
        "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3::DOC-EXAMPLE-BUCKET1/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-grant-full-control": "id=CanonicalUserId-of-AWSaccountA-BucketOwner"
        }
      }
    }
  ]
}
```

Etapa 1.3: criar um perfil do IAM para conceder acesso entre contas na à Conta C na Conta A

No [console do IAM](#), crie um perfil do IAM (**examplerole**) que conceda à Conta C permissão para assumir o perfil do IAM. Você deve manter a conexão como administrador da Conta A porque o perfil deve ser criado 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, escolha Políticas e Criar política.
- b. Próximo a Create Your Own Policy, escolha Select.
- 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 ao perfil `s3:GetObject`; assim, quando o usuário da Conta C assume o perfil, ele só pode executar a operação `s3:GetObject`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*"
    }
  ]
}
```

- e. Escolha Criar política.

As novas políticas aparecem na lista de políticas gerenciadas.

2. No painel de navegação à esquerda, escolha Perfis e Criar novo perfil.
3. Em Selecionar tipo de função, selecione Função para acesso entre contas e o botão Selecionar ao lado de Fornecer acesso entre as Contas da AWS que você possui.
4. Insira o ID de conta da Conta C.

Para esta demonstração, não é necessário exigir que os usuários tenham a autenticação multifator (MFA) para assumir o perfil; portanto, deixe essa opção desmarcada.

5. Escolha Próxima etapa para definir as permissões que serão associadas à função.
6. Marque a caixa de seleção ao lado da política `access-accountA-bucket` que você criou e escolha 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 precisam usar essa função. Os usuários que usam o link vão diretamente para a página Trocar perfil com os campos “ID da conta” e “Nome do perfil” 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 do perfil e selecione Próxima etapa.
8. Depois de revisar o perfil, escolha Criar perfil.

A função `examplerole` é exibida na lista de funções.

9. Selecione o nome de perfil `examplerole`.
10. Selecione a guia Trust Relationships (Relacionamentos de confiança).
11. Clique em Mostrar documento de política e verifique se a política de confiança mostrada corresponde à indicada 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, consulte [AssumeRole](#) na Referência de APIs 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 do recurso da Amazon (ARN) do perfil `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: Realizar as tarefas da Conta B

O exemplo de bucket 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, faça upload de um objeto em ***DOC-EXAMPLE-BUCKET1***.

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET1 --key HappyFace.jpg --  
body HappyFace.jpg --grant-full-control id="canonicalUserId-ofTheBucketOwner" --  
profile AccountBadmin
```

Observe o seguinte:

- O parâmetro `--Profile` especifica o perfil `AccountBadmin`; portanto, 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 estiver na unidade C: de um computador Windows, você especifica `c:\HappyFace.jpg`.

Etapa 3: realizar as tarefas da Conta C

Nas etapas anteriores, a Conta A já criou um perfil, `examplerole`, estabelecendo a confiança com a Conta C. Esse perfil permite que os usuários na Conta C acessem a Conta A. Nessa etapa, o administrador da Conta C cria um usuário (Dave) e delega para ele a permissão `sts:AssumeRole`, recebida da Conta A. Essa abordagem permite que Dave assuma o `examplerole` e tenha acesso temporário à Conta A. A política de acesso que a Conta A anexou ao perfil limita o que Dave pode fazer ao acessar a Conta A (especificamente, obter objetos no ***DOC-EXAMPLE-BUCKET1***).

Etapa 3.1: criar um usuário na Conta C e delegar permissão para assumir `examplerole`

1. Usando o URL de login de usuário do IAM para a Conta C, primeiro faça login no AWS Management Console como o usuário **`AccountAdmin`**.
2. No [console do IAM](#), crie um usuário chamado Dave.

Para obter instruções passo a passo, 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 `examplerole`.
4. Crie uma política em linha para o usuário do IAM Dave para delegar a ele a permissão `sts:AssumeRole` no perfil `examplerole` da conta A.
 - a. No painel de navegação à esquerda, escolha Uses (Usuários).
 - b. Selecione o 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. Escolha Custom Policy e depois Select.
 - 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 deve atualizar a política fornecendo o *AccountA-ID*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["sts:AssumeRole"],
      "Resource": "arn:aws:iam::AccountA-ID:role/examplerole"
    }
  ]
}
```

- h. Selecione 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 o perfil (examplerole) e acessar objetos

Agora Dave pode acessar objetos no bucket de propriedade da Conta A, desta forma:

- Dave primeiro assume a `examplerole` usando suas próprias credenciais. Isso retornará credenciais temporárias.
 - Usando as credenciais temporárias, Dave então acessará os objetos no bucket da Conta A.
1. No prompt de comando, execute o comando `assume-role` da AWS CLI usando o perfil `AccountCDave`.

É necessário atualizar o valor do ARN no comando fornecendo o *AccountA-ID* em que `examplerole` foi definido.

```
aws sts assume-role --role-arn arn:aws:iam::AccountA-ID:role/examplerole --profile AccountCDave --role-session-name test
```

Em resposta, o AWS Security Token Service (AWS 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 DOC-EXAMPLE-BUCKET1 --key HappyFace.jpg SaveFileAs.jpg --profile TempCred
```

Como a política de acesso anexada a `examplerole` permite as ações, o Amazon S3 processa a solicitação. Você pode tentar qualquer outra ação em qualquer outro objeto no bucket.

Se tentar qualquer outra ação, como `get-object-acl`, você terá a permissão negada porque o perfil não tem permissão para realizar essa ação.

```
aws s3api get-object-acl --bucket DOC-EXAMPLE-BUCKET1 --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 *DOC-EXAMPLE-BUCKET1*. O aplicativo pode obter credenciais de segurança temporárias, e a Conta C pode delegar permissão de aplicativo para assumir `exampleRole`.

Etapa 4: limpar

1. Depois de terminar os testes, você pode fazer o seguinte para concluir 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 `exampleRole` que você criou na Conta A. Consulte instruções detalhadas em [Exclusão de um usuário do IAM](#) no Guia do usuário do IAM.
 - No [console do IAM](#), remova o usuário `AccountAdmin`.
2. Faça login no [console do IAM](#) usando as credenciais da Conta B. Exclua o usuário `AccountAdmin`.
3. Faça login no [console do IAM](#) usando as credenciais da Conta C. Exclua `AccountAdmin` e o usuário `Dave`.

Recursos relacionados

Consulte mais informações relacionadas a essa demonstração nos seguintes recursos do Guia do usuário do IAM:

- [Criação de uma função para delegar permissões a um usuário do IAM](#)
- [Tutorial: delegar acesso entre Contas da AWS usando perfis do IAM](#)
- [Gerenciamento de políticas do IAM](#)

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, lista de controle de acesso (ACL) de bucket e ACL de objeto) para decidir se autoriza a solicitação.

Note

Se a verificação de permissão do Amazon S3 não conseguir encontrar permissões válidas, será retornado um erro de permissão negada “Acesso negado” (403 Proibido). Consulte mais informações em [Solucionar erros de acesso negado \(403 proibido\) no Amazon S3](#).

Para determinar se o solicitante tem permissão para executar a operação específica, o Amazon S3 faz o seguinte, na ordem que se apresenta, quando recebe uma solicitação:

1. Converte todas as políticas de acesso relevantes (política de usuário, política de bucket e ACLs) no 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 a conta principal também possuir o recurso na solicitação (bucket ou 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 as credenciais do usuário 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 ou 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 uma 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 listas de controle de acesso (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 de usuários do IAM, políticas de bucket do S3, políticas de endpoint da nuvem privada virtual (VPC) e políticas de controle de serviço (SCPs) do AWS Organizations. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

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](#)
- [Como o Amazon S3 autoriza uma solicitação para uma operação de objeto](#)

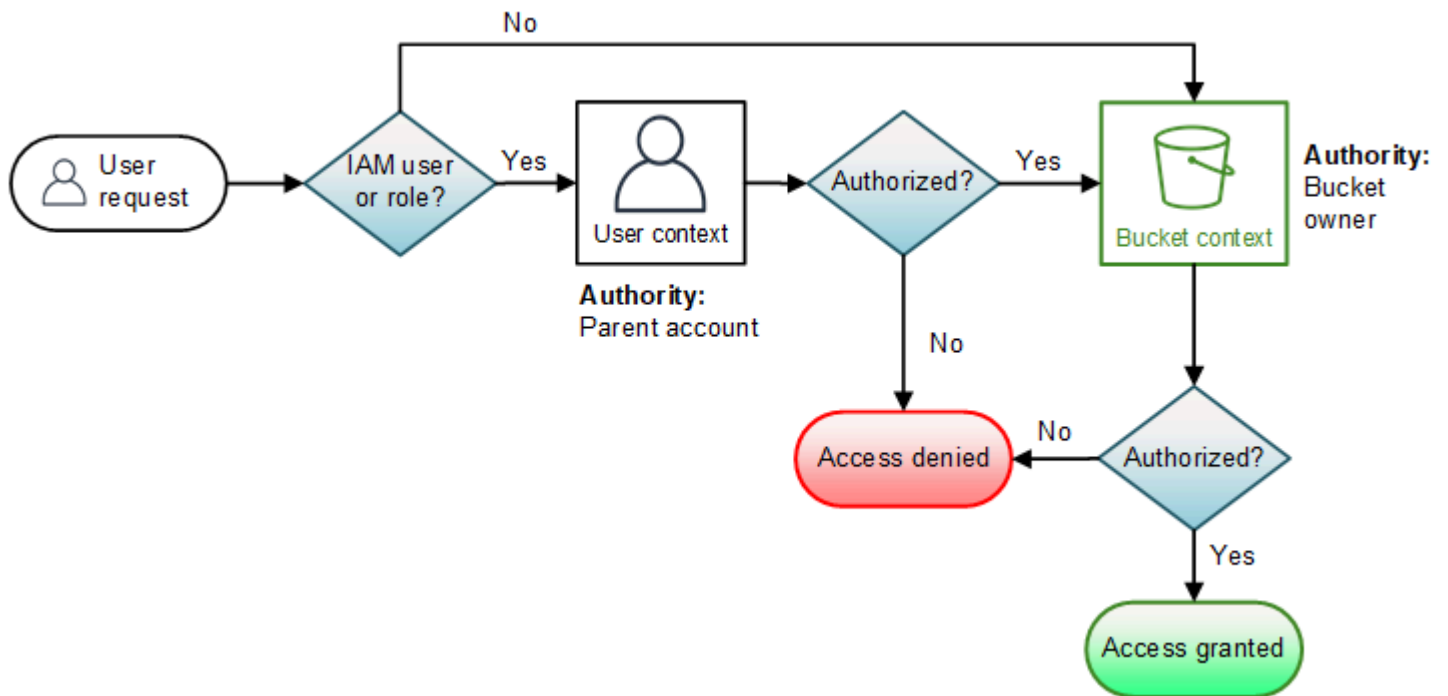
Como o Amazon S3 autoriza uma solicitação para uma operação de bucket

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 se a solicitação vier de uma entidade 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 do usuário ou do 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](#).
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. Se a Conta da AWS que é proprietária do bucket também for a conta principal de uma entidade principal do IAM, ela poderá configurar permissões de bucket em uma política de usuário.

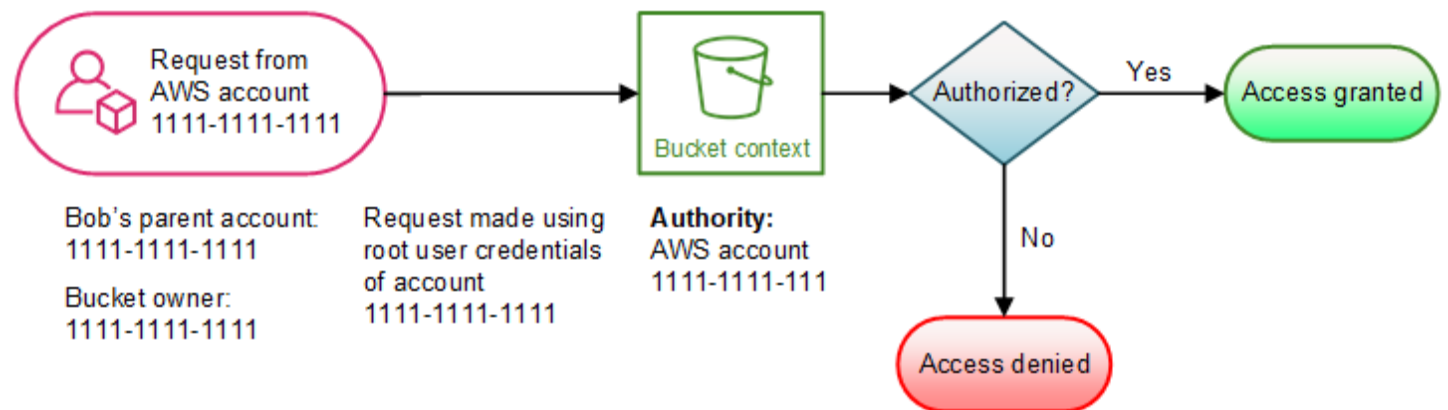
O seguinte é uma ilustração gráfica da avaliação baseada em contexto para a operação de bucket.



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.

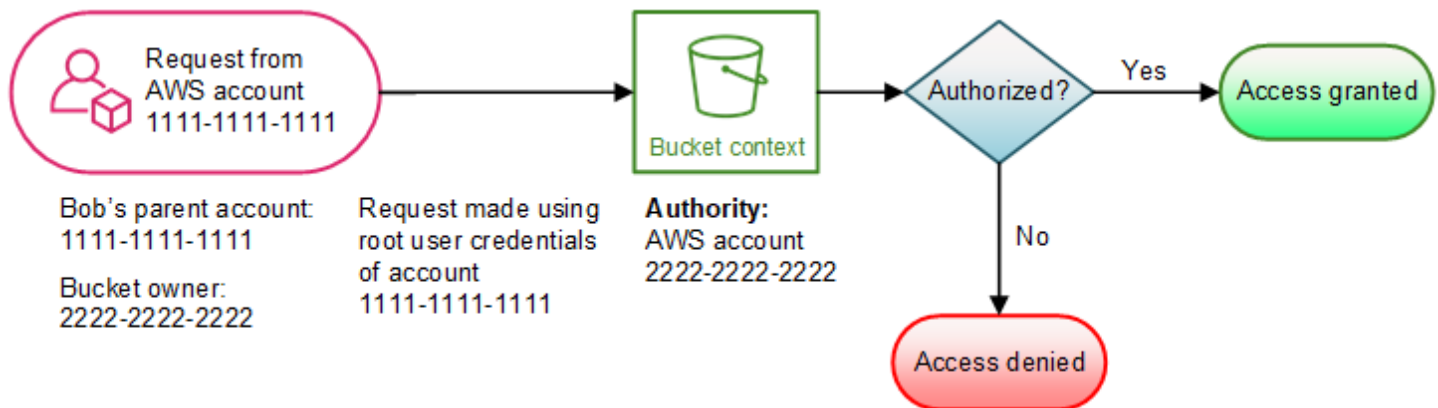


O Amazon S3 executa a avaliação de contexto da seguinte forma:

1. Como a solicitação é feita usando as credenciais do usuário 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 do usuário 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.

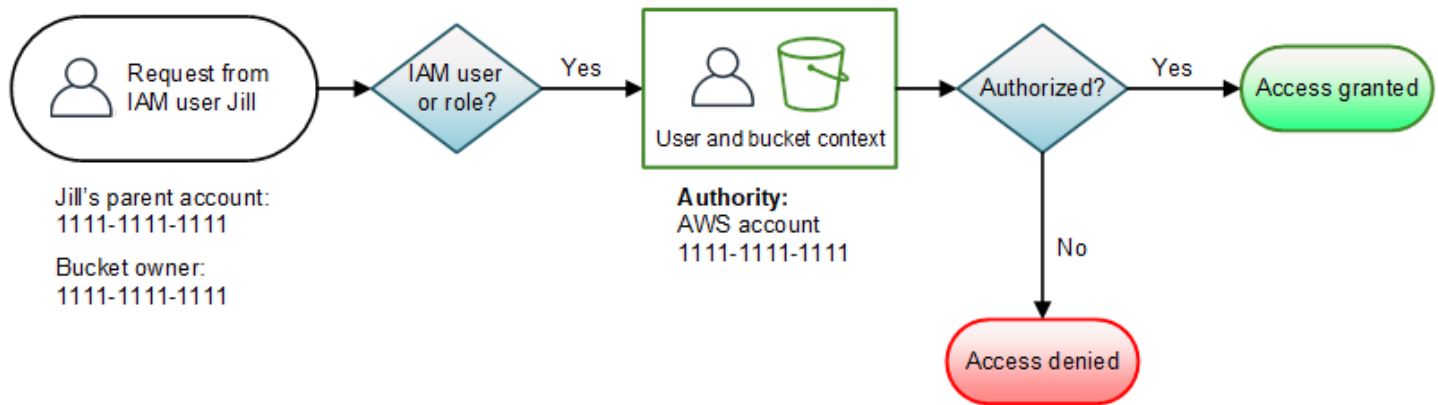


Nesse exemplo, o Amazon S3 avalia o contexto da seguinte forma:

1. Como a solicitação é feita usando as credenciais do usuário 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:

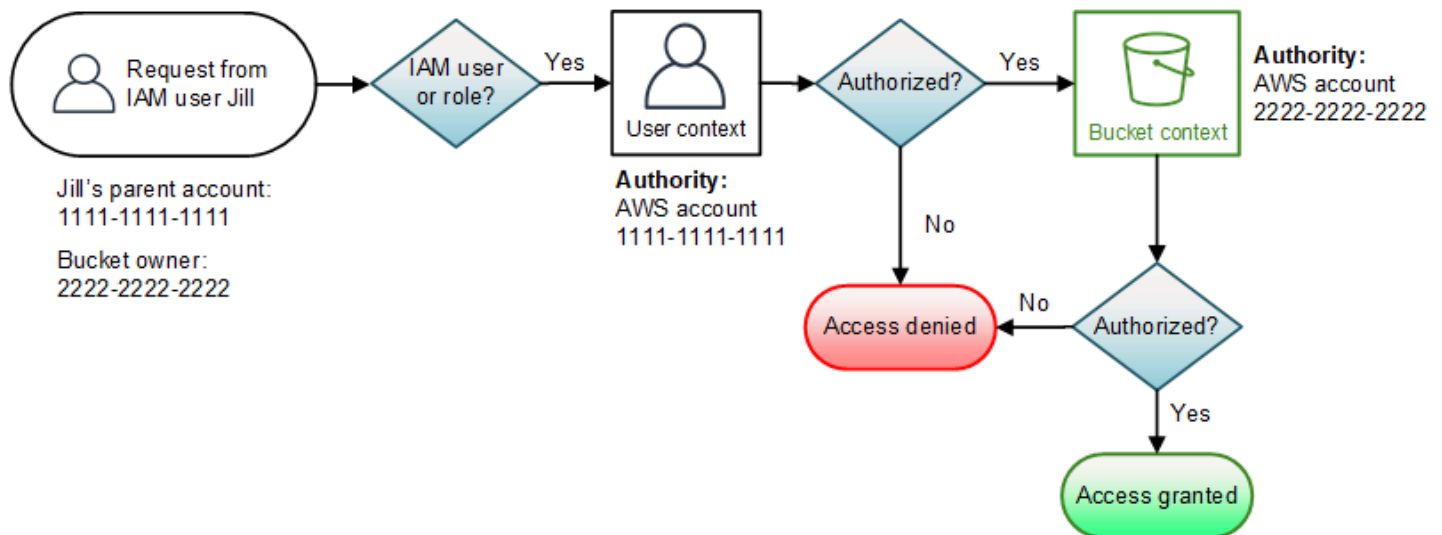
1. Como a solicitação é de uma entidade principal do IAM, no contexto de usuário, o Amazon S3 avalia todas as políticas que pertencem à Conta da AWS pai para determinar se Jill tem permissão para executar a operação.

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 e a ACL do bucket no mesmo contexto, já que pertencem à mesma conta.

2. Como o Amazon S3 avaliou a política do bucket e a ACL do bucket como parte do contexto de usuário, ele não avalia o contexto de bucket.

Exemplo 4: operação de bucket solicitada por uma 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:


1. 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. Se Jill não tiver permissão, a solicitação será negada.
2. 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, ele avalia um subconjunto de políticas em três contextos específicos (contexto do usuário, do bucket e do 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 a conta

principal também possuir o recurso na solicitação (bucket ou 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, 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.

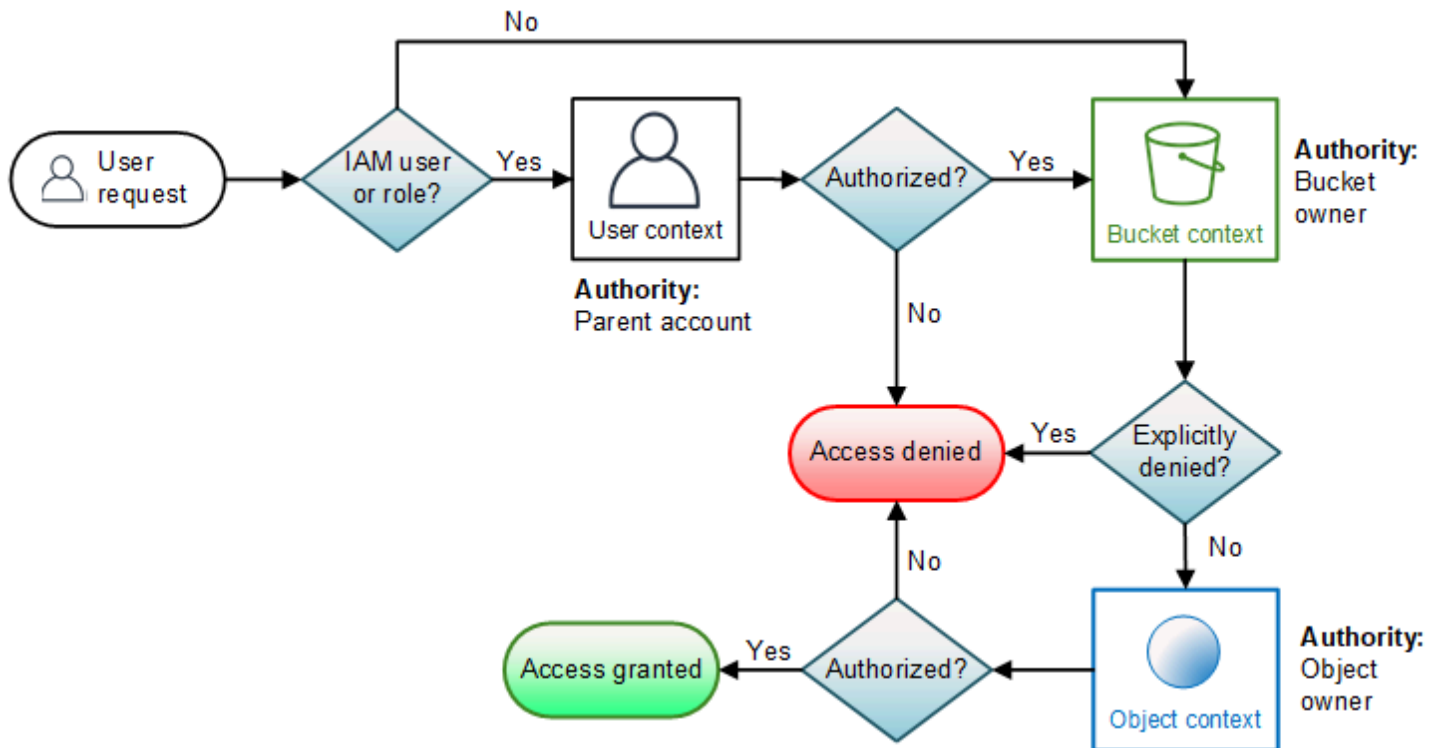
 Note

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 [Demonstrações que usam políticas para gerenciar o acesso a recursos do Amazon S3](#).

Como proprietário do bucket, se você quiser ter propriedade de todos os objetos do bucket e usar políticas de bucket ou políticas baseadas no IAM para gerenciar o acesso a esses objetos, use a configuração aplicada pelo proprietário do bucket em Propriedade do objeto. 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

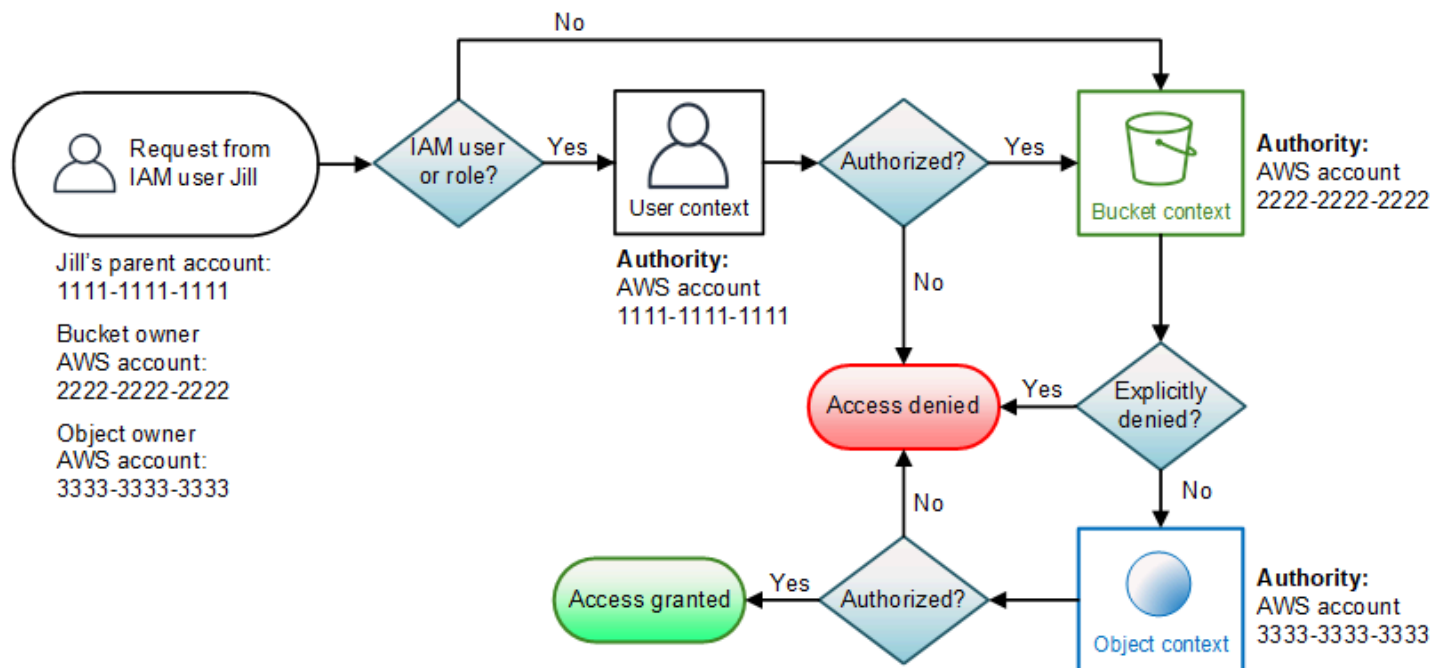
mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

O seguinte é uma ilustração gráfica da avaliação baseada em contexto para uma operação de objeto.



Exemplo de uma 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, GetObject) 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 gerenciadas da AWS para o Amazon S3

Uma política gerenciada pela AWS é uma política independente criada e administrada pela AWS. As políticas gerenciadas pela AWS são criadas para fornecer permissões a vários casos de uso comuns a fim de que você possa começar a atribuir permissões a usuários, grupos e perfis.

Lembre-se de que as políticas gerenciadas pela AWS podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque estão disponíveis para todos os clientes da AWS

usarem. Recomendamos que você reduza ainda mais as permissões definindo [políticas gerenciadas pelo cliente da](#) específicas para seus casos de uso.

Você não pode alterar as permissões definidas em políticas gerenciadas pela AWS. Se a AWS atualiza as permissões definidas em uma política gerenciada pela AWS, a atualização afeta todas as identidades de entidades principais (usuários, grupos e perfis) às quais a política está vinculada. É mais provável que a AWS atualize uma política gerenciada pela AWS quando um novo AWS service (Serviço da AWS) é lançado ou novas operações de API são disponibilizadas para os serviços existentes.

Para obter mais informações, consulte [AWSPolíticas gerenciadas pela](#) no Manual do usuário do IAM.

AWSPolítica gerenciada da : 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.

AWSPolítica gerenciada da : 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.

AWSPolítica gerenciada da : AmazonS3ObjectLambdaExecutionRolePolicy

Fornecer 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 de descrição a AmazonS3ReadOnlyAccess .	O Amazon S3 adicionou permissões s3:Describe* a AmazonS3ReadOnlyAccess .	11 de agosto de 2023
O Amazon S3 adicionou permissões do Lambda para objetos do S3 a AmazonS3FullAccess e AmazonS3ReadOnlyAccess	O Amazon S3 atualizou as políticas AmazonS3FullAccess e AmazonS3ReadOnlyAccess para incluir do Lambda para objetos do S3.	27 de setembro de 2021
O Amazon S3 adicionou AmazonS3ObjectLambdaExecutionRolePolicy	O Amazon S3 adicionou uma nova política gerenciada pela AWS chamada AmazonS3ObjectLambdaExecutionRolePolicy 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

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. A Lente de Armazenamento do S3 cria um perfil vinculado ao serviço (SLR) 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. Os perfis vinculados ao serviço são predefinidos pela Lente de Armazenamento do S3 e incluem todas as permissões que o serviço exige para chamar outros Serviços 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 de permissões 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:ListAWSServiceAccessForOrganization
```

```
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

A Lente de Armazenamento do S3 permite 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:ListAWSServiceAccessForOrganization",
      "organizations:ListDelegatedAdministrators"
    ],
    "Resource": [
      "*"
    ]
  }
}
```

Editar uma função vinculada a serviços para o Amazon S3 Storage Lens

A Lente de Armazenamento do S3 não permite que você edite o perfil vinculado ao serviço `AWSServiceRoleForS3StorageLens`. Depois que você criar um perfil vinculado ao serviço, não poderá alterar o nome do perfil, pois várias entidades podem fazer referência ao perfil. No entanto, você poderá editar a descrição do perfil 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 excluí-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 o `AWSServiceRoleForS3StorageLens`, você deve excluir todas as configurações da Lente de Armazenamento do S3 no nível da organização presentes em todas as Regiões da AWS usando o gerenciamento do AWS Organizations ou as contas de administrador delegadas.

Os recursos são configurações do S3 Storage Lens no nível da organização. Use a Lente de Armazenamento do S3 para limpar os recursos e, depois, use o [console do IAM](#), a CLI, a API REST ou o SDK da AWS para excluir o perfil.

Na API REST, na AWS CLI e nos SDKs, as configurações da Lente de Armazenamento do S3 podem ser descobertas usando `ListStorageLensConfigurations` em todas as regiões onde sua organização tiver criado configurações da Lente de Armazenamento do S3. 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.

Como excluir recursos da Lente de Armazenamento do Amazon S3 usados pelo SLR `AWSServiceRoleForS3StorageLens`

1. Para obter uma lista de configurações de nível de organização, é necessário usar `ListStorageLensConfigurations` em todas as regiões em que você tem configurações da Lente de Armazenamento do S3. Essa lista também pode ser encontrada no console do Amazon S3.
2. Exclua essas configurações dos endpoints regionais apropriados invocando a chamada de API `DeleteStorageLensConfiguration` ou usando o console do Amazon S3.

Como excluir manualmente a função vinculada a serviço usando o IAM

Depois de excluir as configurações, exclua o SLR `AWSServiceRoleForS3StorageLens` no [console do IAM](#), invocando a API `DeleteServiceLinkedRole` do IAM ou usando a AWS CLI ou o SDK da AWS. 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](#).

Solução de problemas de identidade e acesso do Amazon S3

Use as informações a seguir para ajudar a diagnosticar e corrigir problemas comuns que você possa encontrar ao trabalhar com o Amazon S3 e o IAM.

Tópicos

- [Recebi um erro de acesso negado](#)
- [Não tenho autorização para executar uma ação no Amazon S3](#)
- [Não estou autorizado a executar iam:PassRole](#)
- [Quero permitir que as pessoas fora da minha Conta da AWS acessem meus recursos do Amazon S3](#)

Recebi um erro de acesso negado

Verifique se não há uma instrução Deny explícita contra o solicitante ao qual você está tentando conceder permissões, seja na política de bucket ou na política baseada em identidade.

Consulte informações detalhadas sobre a solução de problemas de erros de acesso negado em [Solucionar erros de acesso negado \(403 proibido\) no Amazon S3](#).

Não tenho autorização para executar uma ação no Amazon S3

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM mateojackson tenta usar o console para visualizar detalhes sobre um atributo *my-example-widget* fictício, mas não tem as permissões `s3:GetWidget` fictícias.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
s3:GetWidget on resource: my-example-widget
```

Nesse caso, a política do usuário mateojackson deve ser atualizada para permitir o acesso ao recurso *my-example-widget* usando a ação `s3:GetWidget`.

Se você precisar de ajuda, entre em contato com seu administrador da AWS. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a executar iam:PassRole

Se receber uma mensagem de erro informando que você não tem autorização para executar a ação `iam:PassRole`, atualize suas políticas para permitir a transmissão de um perfil ao Amazon S3.

Alguns Serviços da AWS permitem que você passe um perfil existente para o serviço, em vez de criar um novo perfil de serviço ou perfil vinculado ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta usar o console para executar uma ação no Amazon S3. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se você precisar de ajuda, entre em contato com seu administrador da AWS. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que as pessoas fora da minha Conta da AWS acessem meus recursos do Amazon S3

Você pode criar uma função que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços compatíveis com políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o Amazon S3 é compatível com esses recursos, consulte [Como o Amazon S3 funciona com o IAM](#).
- Para saber como conceder acesso a seus recursos em todas as Contas da AWS pertencentes a você, consulte [Fornecimento de acesso a um usuário do IAM em outra Conta da AWS pertencente a você](#) no Guia de usuário do IAM.

- Para saber como conceder acesso a seus atributos para terceiros Contas da AWS, consulte [Fornecimento de acesso a Contas da AWS pertencentes a terceiros](#) no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte [Conceder acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do usuário do IAM.
- Para saber a diferença entre usar perfis e políticas baseadas em recursos para acesso entre contas, consulte [Como os perfis do IAM diferem de políticas baseadas em recursos](#) no Guia do usuário do IAM.

Gerenciar o acesso com a funcionalidade Concessões de Acesso do S3

Para aderir ao princípio do privilégio mínimo, defina o acesso granular aos dados do Amazon S3 com base em aplicações, personas, grupos ou unidades organizacionais. Você pode usar várias abordagens para obter acesso granular aos dados no Amazon S3, dependendo da escala e da complexidade dos padrões de acesso.

A abordagem mais simples para gerenciar o acesso a números pequenos a médios de conjuntos de dados no Amazon S3 por entidades principais do AWS Identity and Access Management (IAM) é definir [políticas de permissões do IAM](#) e [políticas de buckets do S3](#). Essa estratégia funcionará se as políticas necessárias se encaixarem nos limites de tamanho das políticas de bucket do S3 (20 KB) e das políticas do IAM (5 KB), bem como no [número de entidades principais do IAM permitidas por conta](#).

À medida que o número de conjuntos de dados e casos de uso aumenta, talvez você precise de mais espaço para políticas. Uma abordagem que oferece muito mais espaço para declarações de política é usar os [pontos de acesso do S3](#) como endpoints adicionais para buckets do S3, porque cada ponto de acesso pode ter sua própria política. Você pode definir padrões de controle de acesso bastante granulares, já que é possível ter milhares de pontos de acesso por Região da AWS por conta, com uma política de até 20 KB para cada ponto de acesso. Embora os pontos de acesso do S3 aumentem a quantidade de espaço disponível para políticas, é necessário um mecanismo para que os clientes descubram o ponto de acesso certo para o conjunto de dados certo.

Uma terceira abordagem é implementar um padrão de [agente de sessão do IAM](#), no qual você implementa a lógica de decisão de acesso e gera dinamicamente credenciais de sessão do IAM de curto prazo para cada sessão de acesso. Embora a abordagem de agente de sessão do IAM ofereça

suporte a padrões de permissões arbitrariamente dinâmicos e escale de forma eficaz, você deve criar a lógica do padrão de acesso.

Em vez de usar essas abordagens, você pode usar a funcionalidade Concessões de Acesso do S3 para gerenciar o acesso aos dados do Amazon S3. A funcionalidade Concessões de Acesso do S3 fornece um modelo simplificado para definir permissões de acesso aos dados no Amazon S3 por prefixo, bucket ou objeto. Além disso, você pode usar a funcionalidade Concessões de Acesso do S3 para conceder acesso às entidades principais do IAM e diretamente aos usuários ou grupos do diretório corporativo.

Em geral, você define permissões para os dados no Amazon S3 mapeando usuários e grupos para conjuntos de dados. Você pode usar a funcionalidade Concessões de Acesso do S3 para definir mapeamentos de acesso direto dos prefixos do S3 para usuários e perfis em buckets e objetos do Amazon S3. Com o esquema de acesso simplificado da funcionalidade Concessões de Acesso do S3, você pode conceder acesso somente leitura, somente gravação ou leitura e gravação com base em prefixo do S3 tanto para entidades principais do IAM quanto diretamente para usuários ou grupos de um diretório corporativo. Com esses recursos da funcionalidade Concessões de Acesso do S3, as aplicações podem solicitar dados do Amazon S3 em nome do usuário atualmente autenticado da aplicação.

Quando você integra a funcionalidade Concessões de Acesso do S3 ao recurso de [propagação de identidades confiáveis](#) do AWS IAM Identity Center, as aplicações podem fazer solicitações aos Serviços da AWS (incluindo a funcionalidade Concessões de Acesso do S3) diretamente em nome de um usuário autenticado do diretório corporativo. As aplicações não precisam mais primeiro mapear o usuário para uma entidade principal do IAM. Além disso, como as identidades dos usuários finais são propagadas até o Amazon S3, a auditoria de qual usuário acessou qual objeto do S3 é simplificada. Você não precisa mais reconstruir o relacionamento entre diferentes usuários e sessões do IAM. Quando você usa a funcionalidade Concessões de Acesso do S3 com a propagação de identidades confiáveis do Centro de Identidade do IAM, cada evento de dados do [AWS CloudTrail](#) para o Amazon S3 contém uma referência direta ao usuário final em cujo nome os dados foram acessados.

Para obter mais informações sobre a funcionalidade Concessões de Acesso do S3, consulte os tópicos a seguir.

Tópicos

- [Conceitos da funcionalidade Concessões de Acesso do S3](#)
- [Concessões de Acesso do S3 e identidades de diretórios corporativos](#)

- [Conceitos básicos da funcionalidade Concessões de Acesso do S3](#)
- [Criar uma instância da funcionalidade Concessões de Acesso do S3](#)
- [Registrar um local](#)
- [Criar concessões](#)
- [Solicitar acesso aos dados do Amazon S3 por meio da funcionalidade Concessões de Acesso do S3](#)
- [Acessar dados do S3 por meio de uma concessão de acesso](#)
- [Acesso entre contas da funcionalidade Concessões de Acesso do S3](#)
- [Usar etiquetas da AWS com a funcionalidade Concessões de Acesso do S3](#)
- [Limitações da funcionalidade Concessões de Acesso do S3](#)
- [Integrações da funcionalidade Concessões de Acesso do S3](#)

Conceitos da funcionalidade Concessões de Acesso do S3

A funcionalidade Concessões de Acesso do S3 apresenta os seguintes conceitos para seu esquema de acesso simplificado:

Instâncias da funcionalidade Concessões de Acesso do S3

Uma instância da funcionalidade Concessões de Acesso do S3 é um contêiner lógico para concessões individuais que definem quem tem qual nível de acesso a quais dados do Amazon S3. Você pode ter uma instância da funcionalidade Concessões de Acesso do S3 por Região da AWS por Conta da AWS. Use essa instância da funcionalidade Concessões de Acesso do S3 para controlar o acesso a todos os buckets na mesma conta e Região da AWS. Se quiser usar a funcionalidade Concessões de Acesso do S3 para conceder acesso às identidades de usuários e grupos em seu diretório corporativo, você também deverá associar a instância da funcionalidade Concessões de Acesso do S3 a uma instância do Centro de Identidade do AWS Identity and Access Management (IAM).

Locais

Um local define a quais dados a instância da funcionalidade Concessões de Acesso do S3 pode conceder acesso. A funcionalidade Concessões de Acesso do S3 funciona fornecendo credenciais do IAM com acesso limitado a um prefixo, bucket ou objeto específico do S3. Associe um local da funcionalidade Concessões de Acesso do S3 a um perfil do IAM, a partir do qual essas sessões temporárias são criadas. A configuração de local mais comum é um único local

em `s3://` para toda a instância da funcionalidade Concessões de Acesso do S3, que pode abranger o acesso a todos os buckets do S3 na conta e na Região da AWS. Você também pode criar vários locais em uma instância da funcionalidade Concessões de Acesso do S3. Por exemplo, você pode registrar um bucket como um local `s3://DOC-EXAMPLE-BUCKET1` para concessões que deseja limitar a esse bucket e também pode registrar o local padrão `s3://`.

Concessões

Para restringir o escopo de acesso em um local, crie concessões individuais. Uma concessão individual em uma instância da funcionalidade Concessões de Acesso do S3 permite que uma entidade específica (uma entidade principal do IAM ou um usuário ou grupo em um diretório corporativo) acesse um prefixo, bucket ou objeto do Amazon S3. Para cada concessão, você pode definir um escopo diferente (prefixo, bucket ou objeto) e um nível de acesso (READ, WRITE ou READWRITE). Por exemplo, você pode ter uma concessão que permita acesso READ a `s3://DOC-EXAMPLE-BUCKET1/projects/items/*` para determinado grupo de diretórios corporativos, `01234567-89ab-cdef-0123-456789abcdef`. Essa concessão dá aos usuários desse grupo acesso READ a todos os objetos que tenham um nome de chave com o prefixo `projects/items/` no bucket chamado `DOC-EXAMPLE-BUCKET1`.

Credenciais temporárias da funcionalidade Concessões de Acesso do S3

Uma aplicação pode solicitar credenciais de acesso just-in-time chamando uma nova operação de API do S3, [GetDataAccess](#), para solicitar acesso a um único objeto, prefixo ou bucket com um nível de permissão READ, WRITE ou READWRITE. A instância da funcionalidade Concessões de Acesso do S3 avalia a solicitação `GetDataAccess` em relação às concessões que ela tem. Se houver uma concessão correspondente, a funcionalidade Concessões de Acesso do S3 assumirá o perfil do IAM associado ao local da concessão correspondente. Depois, a funcionalidade Concessões de Acesso do S3 definirá o escopo das permissões da sessão do IAM de acordo com o bucket, prefixo ou objeto do S3 especificado pelo escopo da concessão. O tempo de expiração das credenciais temporárias de acesso é padronizado em 1 hora, mas você pode configurá-lo para qualquer valor entre 15 minutos a 36 horas.

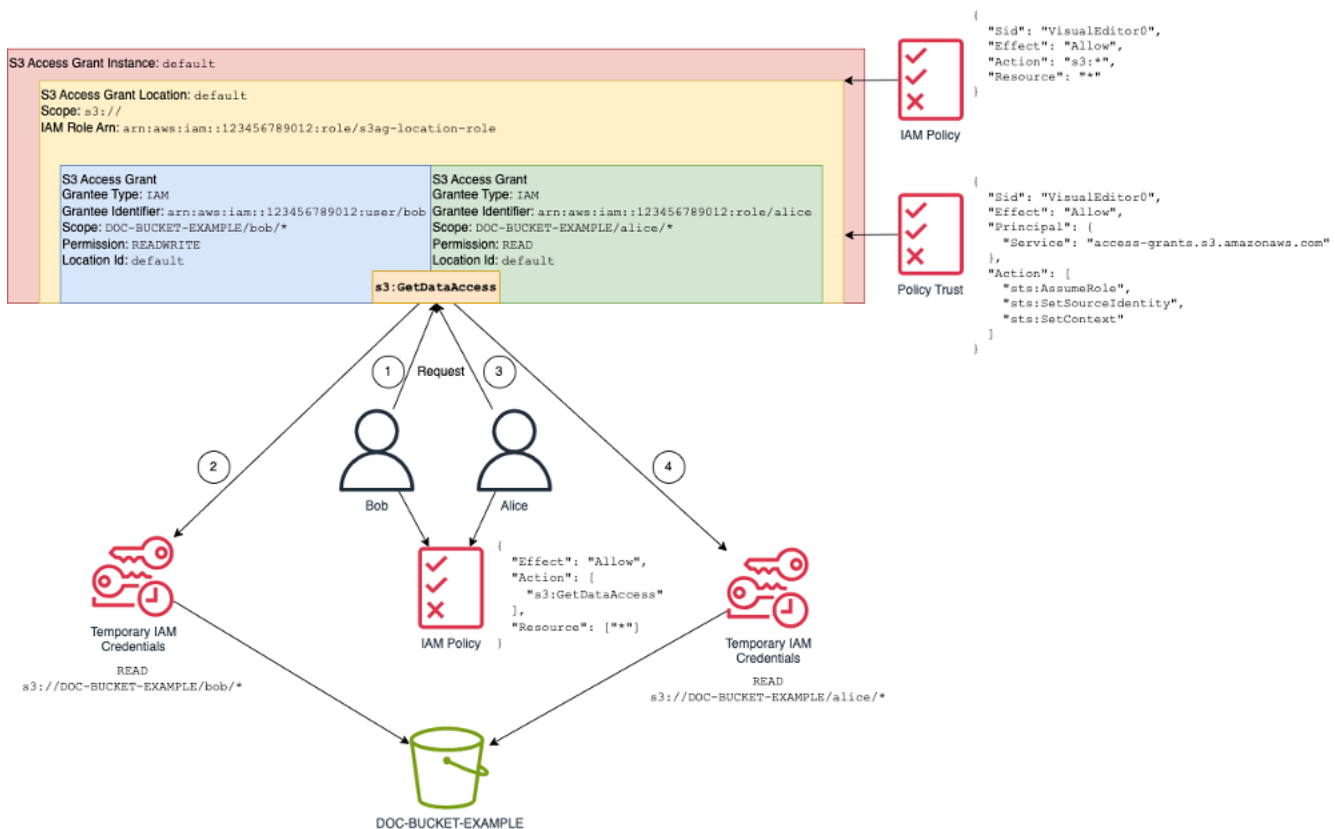
Como funciona

No diagrama a seguir, um local padrão do Amazon S3 com o escopo `s3://` é registrado com o perfil do IAM `s3ag-location-role`. Esse perfil do IAM tem permissões para realizar ações do Amazon S3 dentro da conta quando suas credenciais são obtidas por meio da funcionalidade Concessões de Acesso do S3.

Nesse local, duas concessões de acesso individuais são criadas para dois usuários do IAM. O usuário do IAM Bob recebe acesso READ e WRITE ao prefixo bob/ no bucket DOC-BUCKET-EXAMPLE. Outro perfil do IAM, Alice, só recebe acesso READ ao prefixo alice/ no bucket DOC-BUCKET-EXAMPLE. Uma concessão, mostrada em azul, é definida para que Bob acesse o prefixo bob/ no bucket DOC-BUCKET-EXAMPLE. Uma concessão, mostrada em verde, é definida para que Alice acesse o prefixo alice/ no bucket DOC-BUCKET-EXAMPLE.

Quando chega a hora de Bob coletar dados de READ, o perfil do IAM associado ao local em que sua concessão está chama a operação de API [GetDataAccess](#) da funcionalidade Concessões de Acesso do S3. Se Bob tentar usar READ em qualquer prefixo ou objeto do S3 que comece com s3://DOC-BUCKET-EXAMPLE/bob/*, a solicitação GetDataAccess retornará um conjunto de credenciais de sessão temporária do IAM com permissão para s3://DOC-BUCKET-EXAMPLE/bob/*. Da mesma forma, Bob pode usar WRITE em qualquer prefixo ou objeto do S3 que comece com s3://DOC-BUCKET-EXAMPLE/bob/*, porque a concessão também permite isso.

Da mesma forma, Alice pode usar READ em qualquer coisa que comece com s3://DOC-BUCKET-EXAMPLE/alice/. No entanto, se ela tentar usar WRITE em qualquer bucket, prefixo ou objeto em s3://, receberá um erro de Acesso negado (403 Forbidden), pois não há nenhuma concessão que lhe dê acesso WRITE a nenhum dado. Além disso, se Alice solicitar qualquer nível de acesso (READ ou WRITE) a dados fora de s3://DOC-BUCKET-EXAMPLE/alice/, ela receberá novamente um erro de Acesso negado.



Esse padrão se expande para um grande número de usuários e buckets e simplifica o gerenciamento dessas permissões. Em vez de editar políticas potencialmente grandes de bucket do S3 toda vez que quiser adicionar ou remover uma relação individual de acesso entre prefixo e usuário, você pode adicionar e remover concessões individuais e discretas.

Concessões de Acesso do S3 e identidades de diretórios corporativos

Você pode usar a funcionalidade Concessões de Acesso do Amazon S3 para conceder acesso às entidades principais do AWS Identity and Access Management (IAM) (usuários ou perfis), tanto na mesma Conta da AWS quanto em outras. No entanto, em muitos casos, a entidade que acessa os dados é um usuário final de um diretório corporativo. Em vez de conceder acesso às entidades principais do IAM, você pode usar a funcionalidade Concessões de Acesso do S3 para conceder acesso diretamente aos usuários e grupos corporativos. Com a funcionalidade Concessões de Acesso do S3, você não precisa mais mapear as identidades corporativas para entidades principais intermediárias do IAM a fim de acessar os dados do S3 por meio de aplicações corporativas.

Essa nova funcionalidade de suporte ao uso de identidades de usuário final para acessar dados é fornecida pela associação de uma instância da funcionalidade Concessões de Acesso do S3 a uma instância do AWS IAM Identity Center. O Centro de Identidade do IAM oferece suporte a

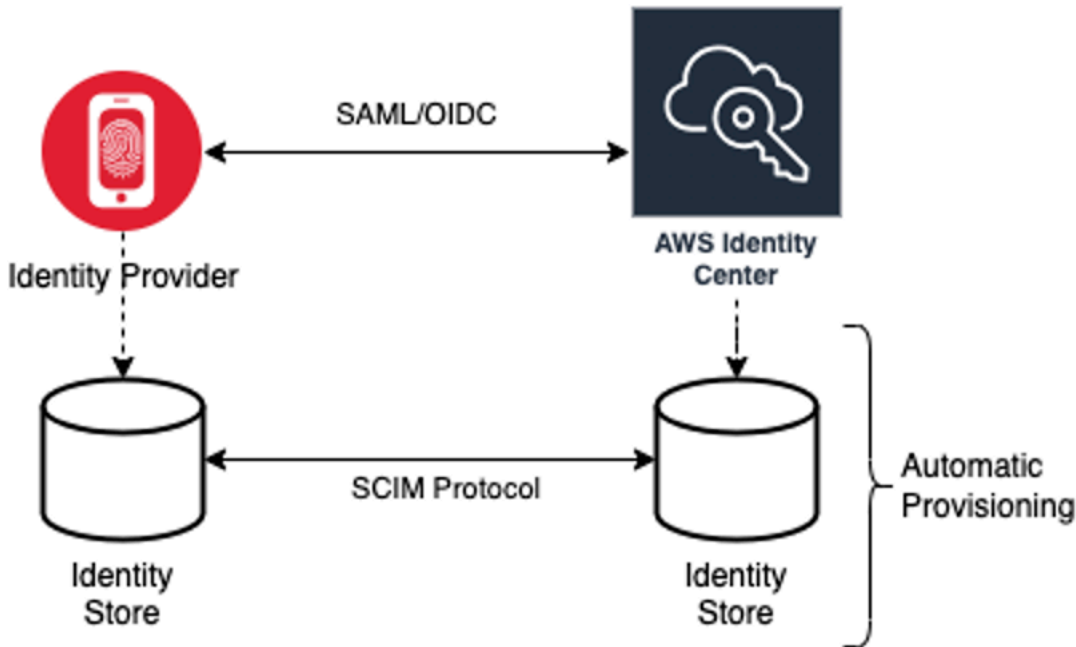
provedores de identidades baseados em padrões e é o centro de todos os serviços ou recursos na AWS, incluindo a funcionalidade Concessões de Acesso do S3, que oferecem suporte a identidades de usuário final. O Centro de Identidade do IAM oferece suporte de autenticação para identidades corporativas por meio de seu recurso de propagação de identidades confiáveis. Para obter mais informações, consulte [Trusted identity propagation across applications](#).

Para começar a usar o suporte a identidades da força de trabalho na funcionalidade Concessões de Acesso do S3, como pré-requisito, comece no Centro de Identidade do IAM, configurando o provisionamento de identidades entre um provedor de identidades corporativas e o Centro de Identidade do IAM. O Centro de Identidade do IAM oferece suporte a provedores de identidades corporativas, como Okta, Microsoft Entra ID (antes chamado de Azure Active Directory) ou qualquer outro provedor de identidades (IdP) externo que ofereça suporte ao protocolo SCIM (Sistema de Gerenciamento de Usuários entre Domínios). Quando você conecta o Centro de Identidade do IAM ao IdP e habilita o provisionamento automático, os usuários e grupos do IdP são sincronizados no repositório de identidades do Centro de Identidade do IAM. Após essa etapa, o Centro de Identidade do IAM tem uma visão própria dos usuários e grupos, para que você possa consultá-los usando outros recurso e Serviços da AWS, como a funcionalidade Concessões de Acesso do S3. Para obter mais informações sobre como configurar o provisionamento automático do Centro de Identidade do IAM, consulte [Automatic provisioning](#) no Guia do usuário do AWS IAM Identity Center.

O Centro de Identidade do IAM é integrado com o AWS Organizations, o que permite que você gerencie centralmente as permissões em várias Contas da AWS sem configurar cada uma das contas manualmente. Em uma organização típica, o administrador de identidades configura uma instância do Centro de Identidade do IAM para toda a organização, como um único ponto de sincronização de identidades. Essa instância do Centro de Identidade do IAM normalmente é executada em uma Conta da AWS dedicada na organização. Nessa configuração comum, você pode se referir às identidades de usuários e grupos na funcionalidade Concessões de Acesso do S3 de qualquer Conta da AWS na organização.

No entanto, se o administrador do AWS Organizations ainda não configurou uma instância central do Centro de Identidade do IAM, você pode criar uma local na mesma conta da instância da funcionalidade Concessões de Acesso do S3. Essa configuração é mais comum para casos de uso de prova de conceito ou desenvolvimento local. Em todos os casos, a instância do Centro de Identidade do IAM deve estar na mesma Região da AWS que a instância da funcionalidade Concessões de Acesso do S3 à qual será associada.

No diagrama a seguir de uma configuração do Centro de Identidade do IAM com um IdP externo, o IdP é configurado com SCIM para sincronizar o repositório de identidades do IdP com o repositório de identidades no Centro de Identidade do IAM.



Para usar as identidades de diretório corporativo com a funcionalidade Concessões de Acesso do S3, faça o seguinte:

- Configure o [provisionamento automático](#) no Centro de Identidade do IAM para sincronizar as informações de usuários e grupos do seu IdP no Centro de Identidade do IAM.
- Configure a fonte de identidades externa no Centro de Identidade do IAM como um emissor de token confiável. Para obter mais informações, consulte [Trusted identity propagation across applications](#) no Guia do usuário do AWS IAM Identity Center.
- Associe a instância da funcionalidade Concessões de Acesso do S3 à instância do Centro de Identidade do IAM. Você pode fazer isso ao [criar a instância da funcionalidade Concessões de Acesso do S3](#). Se você já criou a instância da funcionalidade Concessões de Acesso do S3, consulte [Associar ou desassociar uma instância do Centro de Identidade do IAM](#).

Como as identidades do diretório podem acessar os dados do S3

Suponha que você tenha usuários do diretório corporativo que precisem acessar os dados do S3 por meio de uma aplicação corporativa, por exemplo, uma aplicação de visualização de documentos, integrada ao IdP externo (por exemplo, Okta) para autenticar usuários. A autenticação do usuário nessas aplicações geralmente é feita por meio de redirecionamentos no navegador da web do

usuário. Como os usuários no diretório não são entidades principais do IAM, a aplicação precisa de credenciais do IAM com as quais possa chamar a operação de API `GetDataAccess` da funcionalidade Concessões de Acesso do S3 para [obter credenciais de acesso aos dados do S3](#) em nome dos usuários. Ao contrário dos usuários e perfis do IAM que recebem credenciais diretamente, a aplicação precisa de uma forma de representar um usuário do diretório, que não está mapeado para um perfil do IAM, para que o usuário possa obter acesso aos dados por meio da funcionalidade Concessões de Acesso do S3.

Essa transição, de usuário de diretório autenticado para um chamador do IAM que pode fazer solicitações à funcionalidade Concessões de Acesso do S3 em nome do usuário do diretório, é feita pela aplicação por meio do recurso de emissor de token confiável do Centro de Identidade do IAM. Depois de autenticar o usuário do diretório, a aplicação tem um token de identidade do IdP (por exemplo, Okta) que representa o usuário do diretório de acordo com Okta. A configuração do emissor de token confiável no Centro de Identidade do IAM permite que a aplicação troque esse token do Okta (o locatário do Okta é configurado como o “emissor confiável”) por outro token de identidade do Centro de Identidade do IAM que representará com segurança o usuário do diretório em Serviços da AWS. A aplicação de dados assumirá um perfil do IAM, fornecendo o token de usuário do diretório do Centro de Identidade do IAM como contexto adicional. A aplicação pode usar a sessão do IAM resultante para chamar a funcionalidade Concessões de Acesso do S3. O token representa tanto a identidade da aplicação (a própria entidade principal do IAM) quanto a identidade do usuário do diretório.

A principal etapa dessa transição é a troca de tokens. A aplicação realiza essa troca de tokens chamando a operação de API `CreateTokenWithIAM` no Centro de Identidade do IAM. É evidente que isso também é uma chamada de API da AWS e requer que uma entidade principal do IAM a assine. A entidade principal do IAM que faz essa solicitação geralmente é um perfil do IAM associado à aplicação. Por exemplo, se a aplicação é executada no Amazon EC2, a solicitação `CreateTokenWithIAM` normalmente é executada pelo perfil do IAM associado à instância do EC2 na qual a aplicação é executada. O resultado de uma chamada `CreateTokenWithIAM` bem-sucedida é um novo token de identidade, que será reconhecido em Serviços da AWS.

A próxima etapa, antes que a aplicação possa chamar `GetDataAccess` em nome do usuário do diretório, é que a aplicação obtenha uma sessão do IAM que inclua a identidade do usuário do diretório. A aplicação faz isso com uma solicitação `AssumeRole` do AWS Security Token Service (AWS STS) que também inclui o token do Centro de Identidade do IAM para o usuário do diretório como contexto de identidade adicional. Esse contexto adicional é o que permite que o Centro de Identidade do IAM propague a identidade do usuário do diretório para a próxima etapa. O perfil do

IAM que a aplicação assume é o perfil que precisará de permissões do IAM para chamar a operação `GetDataAccess`.

Tendo assumido o perfil do IAM do portador da identidade com o token do Centro de Identidade do IAM para o usuário do diretório como contexto adicional, a aplicação agora tem tudo o que precisa para fazer uma solicitação `GetDataAccess` assinada em nome do usuário do diretório autenticado.

A propagação do token se baseia nas seguintes etapas:

Criar uma aplicação do Centro de Identidade do IAM

Primeiro, crie uma aplicação no Centro de Identidade do IAM. Essa aplicação usará um modelo que permite que o Centro de Identidade do IAM identifique quais tipos de configurações de aplicação você pode usar. O comando para criar a aplicação exige que você forneça o nome do recurso da Amazon (ARN) da instância do Centro de Identidade do IAM, um nome de aplicação e o ARN do provedor de aplicações. O provedor de aplicações é o provedor SAML ou OAuth que a aplicação usará para fazer chamadas ao Centro de Identidade do IAM.

Para usar o comando a seguir, substitua os *user input placeholders* por suas próprias informações:

```
aws sso-admin create-application \  
  --instance-arn "arn:aws:sso:::instance/ssoins-ssoins-1234567890abcdef" \  
  --application-provider-arn "arn:aws:sso::aws:applicationProvider/custom" \  
  --name MyDataApplication
```

Resposta:

```
{  
  "ApplicationArn": "arn:aws:sso:::123456789012:application/ssoins-  
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d"  
}
```

Criar um emissor de token confiável

Agora que você tem uma aplicação do Centro de Identidade do IAM, a próxima etapa é configurar um emissor de token confiável que será usado para trocar os valores de `IdToken` do IdP por tokens do Centro de Identidade do IAM. Nesta etapa, você precisa fornecer os seguintes itens:

- O URL do emissor do provedor de identidades
- O nome do emissor de token confiável

- O caminho do atributo de reivindicação
- O caminho do atributo do repositório de identidades
- A opção de recuperação do JSON Web Key Set (JWKS)

O caminho do atributo de reivindicação é o atributo do provedor de identidades que será usado para mapear o atributo do repositório de identidades. Normalmente, o caminho do atributo de reivindicação é o endereço de e-mail do usuário, mas você pode usar outros atributos para realizar o mapeamento.

Crie um arquivo chamado `oidc-configuration.json` com o texto a seguir. Para usar esse arquivo, substitua os *user input placeholders* por suas próprias informações.

```
{
  "OidcJwtConfiguration":
    {
      "IssuerUrl": "https://login.microsoftonline.com/a1b2c3d4-abcd-1234-b7d5-
b154440ac123/v2.0",
      "ClaimAttributePath": "preferred_username",
      "IdentityStoreAttributePath": "userName",
      "JwksRetrievalOption": "OPEN_ID_DISCOVERY"
    }
}
```

Para criar o emissor de token confiável, execute o comando a seguir. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws sso-admin create-trusted-token-issuer \
  --instance-arn "arn:aws:sso::instance/ssoins-1234567890abcdef" \
  --name MyEntraIDTrustedIssuer \
  --trusted-token-issuer-type OIDC_JWT \
  --trusted-token-issuer-configuration file://./oidc-configuration.json
```

Resposta

```
{
  "TrustedTokenIssuerArn": "arn:aws:sso::123456789012:trustedTokenIssuer/
ssoins-1234567890abcdef/tti-43b4a822-1234-1234-1234-a1b2c3d41234"
}
```

Conectar a aplicação do Centro de Identidade do IAM ao emissor de token confiável

O emissor de token confiável precisa de mais algumas configurações para funcionar. Defina o público em que o emissor de token confiável confiará. O público é o valor dentro do IdToken que é identificado pela chave e pode ser encontrado nas configurações do provedor de identidades. Por exemplo:

```
1234973b-abcd-1234-abcd-345c5a9c1234
```

Crie um arquivo chamado `grant.json` com o conteúdo a seguir. Para usar esse arquivo, altere o público para que corresponda às configurações do seu provedor de identidades e forneça o ARN do emissor de token confiável que foi retornado pelo comando anterior.

```
{
  "JwtBearer":
  {
    "AuthorizedTokenIssuers":
    [
      {
        "TrustedTokenIssuerArn": "arn:aws:sso::123456789012:trustedTokenIssuer/
ssoins-1234567890abcdef/tti-43b4a822-1234-1234-1234-a1b2c3d41234",
        "AuthorizedAudiences":
        [
          "1234973b-abcd-1234-abcd-345c5a9c1234"
        ]
      }
    ]
  }
}
```

Execute o exemplo de comando a seguir. Para usar esse comando, substitua *user input placeholders* por suas informações.

```
aws sso-admin put-application-grant \
  --application-arn "arn:aws:sso::123456789012:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d" \
  --grant-type "urn:ietf:params:oauth:grant-type:jwt-bearer" \
  --grant file://./grant.json \
```

Esse comando define o emissor de token confiável com configurações para confiar no público do arquivo `grant.json` e vincular esse público à aplicação criada na primeira etapa para a troca de tokens do tipo `jwt-bearer`. A string `urn:ietf:params:oauth:grant-type:jwt-bearer` não

é uma string arbitrária. É um namespace registrado nos perfis de declaração de JSON Web Token (JWT) OAuth. Você pode encontrar mais informações sobre esse namespace em [RFC 7523](#).

Depois, use o comando a seguir para configurar quais escopos o emissor de token confiável incluirá ao trocar valores de IdToken do seu provedor de identidades. Para a funcionalidade Concessões de Acesso do S3, o valor do parâmetro `--scope` é `s3:access_grants:read_write`.

```
aws sso-admin put-application-access-scope \  
  --application-arn "arn:aws:sso::111122223333:application/ssoins-  
ssoins-111122223333abcdef/apl-abcd1234a1b2c3d" \  
  --scope "s3:access_grants:read_write"
```

A última etapa é anexar uma política de recursos à aplicação do Centro de Identidade do IAM. Essa política permitirá que o perfil do IAM da aplicação faça solicitações para a operação de API `sso-oauth:CreateTokenWithIAM` e receba os valores de IdToken do Centro de Identidade do IAM.

Crie um arquivo chamado `authentication-method.json` com o conteúdo a seguir. Substitua `123456789012` pelo ID da sua conta.

```
{  
  "Iam":  
    {  
      "ActorPolicy":  
        {  
          "Version": "2012-10-17",  
          "Statement":  
            [  
              {  
                "Effect": "Allow",  
                "Principal":  
                  {  
                    "AWS": "arn:aws:iam::123456789012:role/webapp"  
                  },  
                "Action": "sso-oauth:CreateTokenWithIAM",  
                "Resource": "*"   
              }   
            ]   
        }   
    }   
}
```

Para anexar a política à aplicação do Centro de Identidade do IAM, execute o seguinte comando:

```
aws sso-admin put-application-authentication-method \  
  --application-arn "arn:aws:sso::123456789012:application/ssoins-  
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d" \  
  --authentication-method-type IAM \  
  --authentication-method file://./authentication-method.json
```

Isso conclui as configurações para usar a funcionalidade Concessões de Acesso do S3 com usuários de diretório por meio de uma aplicação web. Você pode testar essa configuração diretamente na aplicação ou chamar a operação de API `CreateTokenWithIAM` usando o seguinte comando de um perfil do IAM permitido na política da aplicação do Centro de Identidade do IAM:

```
aws sso-oidc create-token-with-iam \  
  --client-id "arn:aws:sso::123456789012:application/ssoins-ssoins-1234567890abcdef/  
apl-abcd1234a1b2c3d" \  
  --grant-type urn:ietf:params:oauth:grant-type:jwt-bearer \  
  --assertion IdToken
```

A resposta será semelhante a esta:

```
{  
  "accessToken": "<suppressed long string to reduce space>",  
  "tokenType": "Bearer",  
  "expiresIn": 3600,  
  "refreshToken": "<suppressed long string to reduce space>",  
  "idToken": "<suppressed long string to reduce space>",  
  "issuedTokenType": "urn:ietf:params:oauth:token-type:refresh_token",  
  "scope": [  
    "sts:identity_context",  
    "s3:access_grants:read_write",  
    "openid",  
    "aws"  
  ]  
}
```

Se você decodificar o valor de `IdToken` que está codificado com base64, poderá ver os pares de valor e chave no formato JSON. A chave `sts:identity_context` contém o valor que a aplicação precisa enviar na solicitação `sts:AssumeRole` para incluir as informações de identidade do usuário do diretório. Veja a seguir um exemplo do `IdToken` decodificado:

```
{  
  "aws:identity_store_id": "d-996773e796",
```

```

"sts:identity_context": "AQoJb3JpZ2luX2VjE0Tt1;<SUPRESSED>",
"sub": "83d43802-00b1-7054-db02-f1d683aacba5",
"aws:instance_account": "123456789012",
"iss": "https://identitycenter.amazonaws.com/ssoins-1234567890abcdef",
"sts:audit_context": "AQoJb3JpZ2luX2VjE0T<SUPRESSED>==",
"aws:identity_store_arn": "arn:aws:identitystore::232642235904:identitystore/
d-996773e796",
"aud": "abcd12344U0gi7n4Yyp0-WV1LWNlbnRyYWwtMQ",
"aws:instance_arn": "arn:aws:sso:::instance/ssoins-6987d7fb04cf7a51",
"aws:credential_id": "EXAMPLEHI5glPh40y9TpApJn8...",
"act": {
  "sub": "arn:aws:sso::232642235904:trustedTokenIssuer/
ssoins-6987d7fb04cf7a51/43b4a822-1020-7053-3631-cb2d3e28d10e"
},
"auth_time": "2023-11-01T20:24:28Z",
"exp": 1698873868,
"iat": 1698870268
}

```

Você pode obter o valor de `sts:identity_context` e enviar essas informações em uma chamada de `sts:AssumeRole`. Veja a seguir um exemplo da sintaxe na CLI. O perfil a ser assumido é um perfil temporário com permissões para invocar `s3:GetDataAccess`.

```

aws sts assume-role \
  --role-arn "arn:aws:iam::123456789012:role/temp-role" \
  --role-session-name "TempDirectoryUserRole" \
  --provided-contexts ProviderArn="arn:aws:iam::aws:contextProvider/
IdentityCenter",ContextAssertion="value from sts:identity_context"

```

Agora você pode usar as credenciais recebidas dessa chamada para invocar a operação de API `s3:GetDataAccess` e receber as credenciais finais com acesso aos recursos do S3.

Conceitos básicos da funcionalidade Concessões de Acesso do S3

A funcionalidade Concessões de Acesso do Amazon S3 é um recurso do Amazon S3 que fornece uma solução de controle de acesso escalável para os dados do S3. A funcionalidade Concessões de Acesso do S3 é um provedor de credenciais do S3, o que significa que você registra sua lista de concessões e respectivo nível na funcionalidade Concessões de Acesso do S3. Depois disso, quando os usuários ou clientes precisam acessar dados do S3, primeiro eles solicitam as credenciais à funcionalidade Concessões de Acesso do S3. Se houver uma concessão correspondente que autorize o acesso, a funcionalidade Concessões de Acesso do S3 proverá credenciais de acesso

temporário com privilégio mínimo. Os usuários ou clientes podem usar as credenciais providas pela funcionalidade Concessões de Acesso do S3 para acessar dados do S3. Com isso em mente, se seus requisitos de dados do S3 exigirem uma configuração de permissão complexa ou grande, você poderá usar a funcionalidade Concessões de Acesso do S3 para escalar as permissões de dados do S3 para usuários, grupos, perfis e aplicações.

Para a maioria dos casos de uso, você pode gerenciar o controle de acesso aos dados do S3 usando o AWS Identity and Access Management (IAM) com políticas de bucket ou políticas baseadas em identidade do IAM.

No entanto, se você tiver requisitos complexos de controle de acesso ao S3, como os seguintes, o uso da funcionalidade Concessões de Acesso do S3 poderá ser muito vantajoso:

- Você está atingindo o limite de tamanho da política de bucket de 20 KB.
- Você concede a identidades humanas, por exemplo, usuários e grupos do Microsoft Entra ID (antes chamado de Azure Active Directory), Okta ou Ping, acesso aos dados do S3 para análises e big data.
- Você precisa fornecer acesso entre contas sem fazer atualizações frequentes nas políticas do IAM.
- Seus dados são não estruturados e estão em nível de objeto, em vez de estruturados e em formato de linha e coluna.

O fluxo de trabalho da funcionalidade Concessões de Acesso do S3 é o seguinte:

Etapas	Descrição
1	<p>Criar uma instância da funcionalidade Concessões de Acesso do S3</p> <p>Para começar, inicie uma instância da funcionalidade Concessões de Acesso do S3 que conterà as concessões de acesso individuais.</p>
2	<p>Registrar um local</p> <p>Depois, registre um local de dados do S3 (como o padrão <code>s3://</code>) e especifique um perfil do IAM padrão que a funcionalidade Concessões de Acesso do S3 assumirá ao fornecer acesso ao local de dados do S3. Você também pode adicionar</p>

Etapas	Descrição
	locais personalizados a buckets ou prefixos específicos e mapeá-los para perfis do IAM personalizados.
3	<p>Criar concessões</p> <p>Crie concessões de permissão individuais. Especifique nessas concessões de permissão o local registrado do S3, o escopo de acesso aos dados dentro do local, a identidade do beneficiário e o nível de acesso (READ, WRITE ou READWRITE).</p>
4	<p>Solicitar acesso aos dados do S3</p> <p>Quando usuários, aplicações e Serviços da AWS desejam acessar dados do S3, primeiro eles fazem uma solicitação de acesso. A funcionalidade Concessões de Acesso do S3 determina se a solicitação deve ser autorizada. Se houver uma concessão correspondente que autorize o acesso, a funcionalidade Concessões de Acesso do S3 usará o perfil do IAM do local registrado associado a essa concessão para prover credenciais temporárias ao solicitante.</p>
5	<p>Acessar dados do S3</p> <p>As aplicações usam as credenciais temporárias fornecidas pela funcionalidade Concessões de Acesso do S3 para acessar os dados do S3.</p>

Criar uma instância da funcionalidade Concessões de Acesso do S3

Para começar a usar a funcionalidade Concessões de Acesso do Amazon S3, comece criando uma instância da funcionalidade Concessões de Acesso do S3. Você só pode criar uma instância da funcionalidade Concessões de Acesso do S3 por Região da AWS por conta. A instância da funcionalidade Concessões de Acesso do S3 serve como contêiner para as funcionalidades desse recurso, que incluem locais registrados e concessões.

Com a funcionalidade Concessões de Acesso do S3, você pode criar concessões de permissão para os dados do S3 para usuários e perfis do AWS Identity and Access Management (IAM). Se

Se você [adicionou seu diretório de identidades corporativas](#) ao AWS IAM Identity Center, poderá associar essa instância do Centro de Identidade do IAM do seu diretório corporativo à instância da funcionalidade Concessões de Acesso do S3. Depois de fazer isso, você poderá criar concessões de acesso para os usuários e grupos corporativos. Se você ainda não adicionou seu diretório de identidades corporativas ao Centro de Identidade do IAM, poderá associar a instância da funcionalidade Concessões de Acesso do S3 a uma instância do Centro de Identidade do IAM posteriormente.

Você pode criar uma instância da funcionalidade Concessões de Acesso do S3 usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI), AWS SDKs ou a API REST do Amazon S3.

Uso do console do S3

Antes de conceder acesso aos dados do S3 com a funcionalidade Concessões de Acesso do S3, primeiro é necessário criar uma instância da funcionalidade Concessões de Acesso do S3 na mesma Região da AWS dos dados do S3.

Pré-requisitos

Se você quiser conceder acesso aos dados do S3 usando identidades de um diretório corporativo, [adicione o diretório de identidades corporativas](#) ao AWS IAM Identity Center. Se você ainda não estiver com tudo pronto para fazer isso, poderá associar a instância da funcionalidade Concessões de Acesso do S3 a uma instância do Centro de Identidade do IAM posteriormente.

Como criar uma instância da funcionalidade Concessões 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. Na barra de navegação, escolha o nome da Região da AWS exibida no momento. Em seguida, escolha a região para a qual você deseja mudar.
3. No painel de navegação, escolha Access Grants.
4. Na página Concessões de Acesso do S3, escolha Criar instância da funcionalidade Concessões de Acesso do S3.
 - a. Na Etapa 1 do assistente Configurar instância do Access Grants, verifique se você deseja criar a instância na Região da AWS atual. Garanta que seja a mesma Região da AWS em que os dados do S3 estão localizados. Você pode criar uma instância da funcionalidade Concessões de Acesso do S3 por Região da AWS por conta.

- b. (Opcional) Se você [adicionou seu diretório de identidades corporativas](#) ao AWS IAM Identity Center, poderá associar essa instância do Centro de Identidade do IAM do seu diretório corporativo à instância da funcionalidade Concessões de Acesso do S3.

Para fazer isso, selecione Adicionar instância do Centro de Identidade do IAM em **região**. Depois, insira o nome do recurso da Amazon (ARN) da instância do Centro de Identidade do IAM.

Se você ainda não adicionou seu diretório de identidades corporativas ao Centro de Identidade do IAM, poderá associar a instância da funcionalidade Concessões de Acesso do S3 a uma instância do Centro de Identidade do IAM posteriormente.

- c. Para criar a instância da funcionalidade Concessões de Acesso do S3, escolha Próximo. Para registrar um local, consulte [Etapa 2: Registrar um local](#).
5. Se as opções Próximo ou Criar instância da funcionalidade Concessões de Acesso do S3 estiverem desabilitadas:

Não é possível criar a instância

- Talvez você já tenha uma instância da funcionalidade Concessões de Acesso do S3 na mesma Região da AWS. No painel de navegação, escolha Access Grants. Na página Concessões de Acesso do S3, role para baixo até a seção Instância da funcionalidade Concessões de Acesso do S3 em sua conta para determinar se já existe uma instância.
- Talvez você não tenha a permissão `s3:CreateAccessGrantsInstance` necessária para criar uma instância da funcionalidade Concessões de Acesso do S3. Entre em contato com o administrador da conta. Para conferir as permissões adicionais que são necessárias ao associar uma instância do Centro de Identidade do IAM a uma instância da funcionalidade Concessões de Acesso do S3, consulte [CreateAccessGrantsInstance](#).

Usando a AWS CLI

Para instalar a AWS CLI, consulte [Instalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Para usar o comando a seguir, substitua os *user input placeholders* por suas próprias informações.

Exemplo Criar uma instância da funcionalidade Concessões de Acesso do S3

```
aws s3control create-access-grants-instance \  
--account-id 111122223333 \  
--region us-east-2
```

Resposta:

```
{  
  "CreatedAt": "2023-05-31T17:54:07.893000+00:00",  
  "AccessGrantsInstanceId": "default",  
  "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/  
default"  
}
```

Uso da API REST

Você pode usar a API REST do Amazon S3 para criar uma instância da funcionalidade Concessões de Acesso do S3. Para obter informações sobre o suporte à API REST para gerenciar instâncias da funcionalidade Concessões de Acesso do S3, consulte as seguintes seções na Referência da API do Amazon Simple Storage Service:

- [AssociateAccessGrantsIdentityCenter](#)
- [CreateAccessGrantsInstance](#)
- [DeleteAccessGrantsInstance](#)
- [DissociateAccessGrantsIdentityCenter](#)
- [GetAccessGrantsInstance](#)
- [GetAccessGrantsInstanceForPrefix](#)
- [GetAccessGrantsInstanceResourcePolicy](#)
- [ListAccessGrantsInstances](#)
- [PutAccessGrantsInstanceResourcePolicy](#)

Uso dos AWS SDKs

Esta seção fornece um exemplo de como criar uma instância da funcionalidade Concessões de Acesso do S3 usando os AWS SDKs.

Java

Este exemplo cria a instância da funcionalidade Concessões de Acesso do S3, que serve como um contêiner para as concessões de acesso individuais. Você pode ter uma instância da funcionalidade Concessões de Acesso do S3 por Região da AWS em sua conta. A resposta inclui o ID de instância default e um nome do recurso da Amazon (ARN) que é gerado para a instância da funcionalidade Concessões de Acesso do S3.

Example Criar uma solicitação de instância da funcionalidade Concessões de Acesso do S3

```
public void createAccessGrantsInstance() {
    CreateAccessGrantsInstanceRequest createRequest =
        CreateAccessGrantsInstanceRequest.builder().accountId("111122223333").build();
    CreateAccessGrantsInstanceResponse createResponse =
        s3Control.createAccessGrantsInstance(createRequest);LOGGER.info("CreateAccessGrantsInstance
    " + createResponse);
}
```

Resposta:

```
CreateAccessGrantsInstanceResponse(
    CreatedAt=2023-06-07T01:46:20.507Z,
    AccessGrantsInstanceId=default,
    AccessGrantsInstanceArn=arn:aws:s3:us-east-2:111122223333:access-grants/default)
```

Tópicos

- [Ver os detalhes de uma instância da funcionalidade Concessões de Acesso do S3](#)
- [Associar ou desassociar uma instância do Centro de Identidade do IAM](#)
- [Excluir uma instância da funcionalidade Concessões de Acesso do S3](#)

Ver os detalhes de uma instância da funcionalidade Concessões de Acesso do S3

Você pode ver os detalhes de uma instância da funcionalidade Concessões de Acesso do Amazon S3 em determinada Região da AWS. Você também pode listar as instâncias da funcionalidade Concessões de Acesso do S3, incluindo as instâncias que foram compartilhadas com você por meio do AWS Resource Access Manager (AWS RAM).

Você pode ver os detalhes de uma instância da funcionalidade Concessões de Acesso do S3 ou listar as instâncias da funcionalidade Concessões de Acesso do S3 usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI), a API REST do Amazon S3 e os AWS SDKs.

Uso do console do S3

Como visualizar uma instância da funcionalidade Concessões 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 Grants.
3. Na página Concessões de Acesso do S3, escolha a região que contém a instância da funcionalidade Concessões de Acesso do S3 com a qual você deseja trabalhar.
4. A página Concessões de Acesso do S3 lista as instâncias da funcionalidade Concessões de Acesso do S3 e todas as instâncias entre contas que foram compartilhadas com sua conta. Para ver os detalhes de uma instância, escolha Visualizar detalhes.

Usando a AWS CLI

Para instalar a AWS CLI, consulte [Instalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Para usar o comando a seguir, substitua os *user input placeholders* por suas próprias informações.

Example : obter os detalhes de uma instância da funcionalidade Concessões de Acesso do S3

```
aws s3control get-access-grants-instance \  
  --account-id 111122223333 \  
  --region us-east-2
```

Resposta:

```
{  
  "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/  
default",  
  "AccessGrantsInstanceId": "default",  
  "CreatedAt": "2023-05-31T17:54:07.893000+00:00"
```

```
}
```

Example : listar todas as instâncias da funcionalidade Concessões de Acesso do S3 para uma conta

Essa ação lista as instâncias da funcionalidade Concessões de Acesso do S3 para uma conta. Você só pode ter uma instância da funcionalidade Concessões de Acesso do S3 por Região da AWS. Essa ação também lista outras instâncias da funcionalidade Concessões de Acesso do S3 entre contas às quais sua conta tem acesso.

```
aws s3control list-access-grants-instances \  
  --account-id 111122223333 \  
  --region us-east-2
```

Resposta:

```
{  
  "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/  
default",  
  "AccessGrantsInstanceId": "default",  
  "CreatedAt": "2023-05-31T17:54:07.893000+00:00"  
}
```

Uso da API REST

Para obter informações sobre o suporte à API REST do Amazon S3 para gerenciar instâncias da funcionalidade Concessões de Acesso do S3, consulte as seguintes seções na Referência da API do Amazon Simple Storage Service:

- [GetAccessGrantsInstance](#)
- [GetAccessGrantsInstanceForPrefix](#)
- [ListAccessGrantsInstances](#)

Uso dos AWS SDKs

Esta seção fornece exemplos de como obter os detalhes de uma instância da funcionalidade Concessões de Acesso do S3 usando os AWS SDKs.

Para usar os exemplos a seguir, substitua *user input placeholders* por suas próprias informações.

Java

Example : obter uma instância da funcionalidade Concessões de Acesso do S3

```
public void getAccessGrantsInstance() {
    GetAccessGrantsInstanceRequest getRequest = GetAccessGrantsInstanceRequest.builder()
        .accountId("111122223333")
        .build();
    GetAccessGrantsInstanceResponse getResponse =
        s3Control.getAccessGrantsInstance(getRequest);
    LOGGER.info("GetAccessGrantsInstanceResponse: " + getResponse);
}
```

Resposta:

```
GetAccessGrantsInstanceResponse(
    AccessGrantsInstanceArn=arn:aws:s3:us-east-2: 111122223333:access-grants/default,
    CreatedAt=2023-06-07T01:46:20.507Z)
```

Example : listar todas as instâncias da funcionalidade Concessões de Acesso do S3 para uma conta

Essa ação lista as instâncias da funcionalidade Concessões de Acesso do S3 para uma conta. Você só pode ter uma instância da funcionalidade Concessões de Acesso do S3 por região. Essa ação também pode listar outras instâncias da funcionalidade Concessões de Acesso do S3 entre contas às quais sua conta tem acesso.

```
public void listAccessGrantsInstances() {
    ListAccessGrantsInstancesRequest listRequest =
        ListAccessGrantsInstancesRequest.builder()
        .accountId("111122223333")
        .build();
    ListAccessGrantsInstancesResponse listResponse =
        s3Control.listAccessGrantsInstances(listRequest);
    LOGGER.info("ListAccessGrantsInstancesResponse: " + listResponse);
}
```

Resposta:

```
ListAccessGrantsInstancesResponse(
    AccessGrantsInstancesList=[
    ListAccessGrantsInstanceEntry(
```

```
AccessGrantsInstanceId=default,  
AccessGrantsInstanceArn=arn:aws:s3:us-east-2:111122223333:access-grants/default,  
CreatedAt=2023-06-07T04:28:11.728Z  
)  
]  
)
```

Associar ou desassociar uma instância do Centro de Identidade do IAM

Na funcionalidade Concessões de Acesso do Amazon S3, você pode associar a instância do AWS IAM Identity Center do diretório de identidades corporativas a uma instância da funcionalidade Concessões de Acesso do S3. Depois de fazer isso, você pode criar concessões de acesso para os usuários e grupos do diretório corporativo, além de usuários e perfis do AWS Identity and Access Management (IAM).

Se você não quiser mais criar concessões de acesso para os usuários e grupos do diretório corporativo, poderá desassociar a instância do Centro de Identidade do IAM da sua instância da funcionalidade Concessões de Acesso do S3.

Você pode associar ou desassociar uma instância do Centro de Identidade do IAM usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI), a API REST do Amazon S3 e AWS SDKs.

Uso do console do S3

Antes de associar uma instância do Centro de Identidade do IAM a uma instância da funcionalidade Concessões de Acesso do S3, é necessário adicionar o diretório de identidades corporativas ao Centro de Identidade do IAM. Para ter mais informações, consulte [the section called “Concessões de Acesso do S3 e identidades de diretórios corporativos”](#).

Como associar uma instância do Centro de Identidade do IAM a uma instância da funcionalidade Concessões 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 Grants.
3. Na página Concessões de Acesso do S3, escolha a região que contém a instância da funcionalidade Concessões de Acesso do S3 com a qual você deseja trabalhar.
4. Escolha Visualizar detalhes da instância.

5. Na página de detalhes, na seção Centro de Identidade do IAM, escolha Adicionar uma instância do Centro de Identidade do IAM ou Cancelar registro de uma instância do Centro de Identidade do IAM já associada.

Usando a AWS CLI

Para instalar a AWS CLI, consulte [Instalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Para usar o comando a seguir, substitua os *user input placeholders* por suas próprias informações.

Example : associar uma instância do Centro de Identidade do IAM a uma instância da funcionalidade Concessões de Acesso do S3

```
aws s3control associate-access-grants-identity-center \  
  --account-id 111122223333 \  
  --identity-center-arn arn:aws:sso:::instance/ssoins-1234a567bb89012c \  
  --profile access-grants-profile \  
  --region eu-central-1  
  
// No response body
```

Example : desassociar uma instância do Centro de Identidade do IAM de uma instância da funcionalidade Concessões de Acesso do S3

```
aws s3control dissociate-access-grants-identity-center \  
  --account-id 111122223333 \  
  --profile access-grants-profile \  
  --region eu-central-1  
  
// No response body
```

Uso da API REST

Para obter informações sobre o suporte à API REST do Amazon S3 para gerenciar a associação entre uma instância do Centro de Identidade do IAM e uma instância da funcionalidade Concessões de Acesso do S3, consulte as seguintes seções na Referência da API do Amazon Simple Storage Service:

- [AssociateAccessGrantsIdentityCenter](#)

- [DissociateAccessGrantsIdentityCenter](#)

Excluir uma instância da funcionalidade Concessões de Acesso do S3

Você pode excluir uma instância da funcionalidade Concessões de Acesso do Amazon S3 de uma Região da AWS em sua conta. No entanto, antes de excluir uma instância da funcionalidade Concessões de Acesso do S3, é necessário fazer o seguinte:

- Exclua todos os recursos na instância da funcionalidade Concessões de Acesso do S3, incluindo todas as concessões e locais. Para obter mais informações, consulte [Delete a grant](#) e [Delete a location](#).
- Se você associou uma instância do AWS IAM Identity Center à sua instância da funcionalidade Concessões de Acesso do S3, deverá desassociar a instância do Centro de Identidade do IAM. Para obter mais informações, consulte [Associar ou desassociar uma instância do Centro de Identidade do IAM](#).

Important

Se você excluir uma instância da funcionalidade Concessões de Acesso do S3, a exclusão será permanente e não poderá ser desfeita. Todos os beneficiários que receberam acesso por meio das concessões nessa instância da funcionalidade Concessões de Acesso do S3 perderão o acesso aos dados do S3.

Você pode excluir uma instância da funcionalidade Concessões de Acesso do S3 usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI), a API REST do Amazon S3 ou os AWS SDKs.

Uso do console do S3

Como excluir uma instância da funcionalidade Concessões 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 Grants.
3. Na página Concessões de Acesso do S3, escolha a região que contém a instância da funcionalidade Concessões de Acesso do S3 com a qual você deseja trabalhar.

4. Escolha Visualizar detalhes da instância.
5. Na página de detalhes da instância, escolha Excluir instância no canto superior direito.
6. Na caixa de diálogo exibida, escolha Excluir. Essa ação não pode ser desfeita.

Usando a AWS CLI

Para instalar a AWS CLI, consulte [Instalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Para usar o comando a seguir, substitua os *user input placeholders* por suas próprias informações.

Note

Antes de excluir uma instância da funcionalidade Concessões de Acesso do S3, primeiro é necessário excluir todas as concessões e locais criados na instância da funcionalidade Concessões de Acesso do S3. Se você associou uma instância do Centro de Identidade do IAM à sua instância da funcionalidade Concessões de Acesso do S3, deverá desassociá-la primeiro.

Example : excluir uma instância da funcionalidade Concessões de Acesso do S3

```
aws s3control delete-access-grants-instance \  
--account-id 111122223333 \  
--profile access-grants-profile \  
--region us-east-2 \  
--endpoint-url https://s3-control.us-east-2.amazonaws.com \  
  
// No response body
```

Uso da API REST

Para obter informações sobre o suporte à API REST do Amazon S3 para excluir uma instância da funcionalidade Concessões de Acesso do S3, consulte [DeleteAccessGrantsInstance](#) na Referência da API do Amazon Simple Storage Service.

Uso dos AWS SDKs

Esta seção fornece exemplos de como excluir uma instância da funcionalidade Concessões de Acesso do S3 usando os AWS SDKs.

Para usar o exemplo a seguir, substitua os *user input placeholders* por suas próprias informações.

Java

Note

Antes de excluir uma instância da funcionalidade Concessões de Acesso do S3, primeiro é necessário excluir todas as concessões e locais criados na instância da funcionalidade Concessões de Acesso do S3. Se você associou uma instância do Centro de Identidade do IAM à sua instância da funcionalidade Concessões de Acesso do S3, deverá desassociá-la primeiro.

Example : excluir uma instância da funcionalidade Concessões de Acesso do S3

```
public void deleteAccessGrantsInstance() {
    DeleteAccessGrantsInstanceRequest deleteRequest =
        DeleteAccessGrantsInstanceRequest.builder()
            .accountId("111122223333")
            .build();
    DeleteAccessGrantsInstanceResponse deleteResponse =
        s3Control.deleteAccessGrantsInstance(deleteRequest);
    LOGGER.info("DeleteAccessGrantsInstanceResponse: " + deleteResponse);
}
```

Registrar um local

Depois de [criar uma instância da funcionalidade Concessões de Acesso do Amazon S3](#) em uma Região da AWS em sua conta, você pode registrar um local do S3 nessa instância. Um local é um recurso do S3 que contém dados aos quais você deseja conceder acesso. Você pode registrar o local padrão `s3://`, que incluir todos os seus buckets na Região da AWS, depois restringir o escopo de acesso por meio de concessões de acesso individuais. Você também pode registrar um bucket específico, ou um bucket e um prefixo, como um local.

Primeiro, você deve registrar pelo menos um local com a instância da funcionalidade Concessões de Acesso do S3 antes de poder criar concessões de acesso. Ao registrar um local, você também deve especificar o perfil do AWS Identity and Access Management (IAM) que a funcionalidade Concessões de Acesso do S3 assume para atender às solicitações de tempo de execução do local e reduzir o escopo das permissões até a concessão específica no tempo de execução.

URI do S3	IAM role (Perfil do IAM)	Descrição
<code>s3://</code>	<i>Default-IAM-role</i>	O local padrão <code>s3://</code> inclui todos os buckets na Região da AWS.
<code>s3://DOC-EXAMP LE-BUCKET1 /</code>	<i>IAM-role-For- bucket</i>	Esse local inclui todos os objetos no bucket especificado.

Antes de poder registrar um local, faça o seguinte:

- Crie um ou mais buckets que contenham os dados aos quais você deseja conceder acesso. Esses buckets devem estar localizados na mesma Região da AWS da instância da funcionalidade Concessões de Acesso do S3. Para mais informações, consulte [Criar um bucket](#).

Para adicionar um prefixo a um bucket, consulte [Criar nomes de chave de objeto](#).

- Crie um perfil do IAM e conceda à entidade principal de serviço da funcionalidade Concessões de Acesso do S3 acesso a esse perfil no arquivo da política de recursos. Para fazer isso, você pode criar um arquivo JSON que contém as instruções a seguir. Para adicionar a política de recursos à sua conta, consulte [Criar e anexar sua primeira política gerenciada pelo cliente](#).

TestRolePolicy.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1234567891011",
      "Action": ["sts:AssumeRole", "sts:SetSourceIdentity", "sts:SetContext"],
      "Effect": "Allow",
      "Principal": {"Service": "access-grants.s3.amazonaws.com"}
    }
  ]
}
```

```
]
}
```

- Crie uma política do IAM para anexar permissões do Amazon S3 ao perfil do IAM. Confira o exemplo de arquivo `iam-policy.json` a seguir e substitua os *user input placeholders* por suas próprias informações.

Note

- Se você usa criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) para criptografar os dados, o exemplo a seguir inclui as permissões do AWS KMS necessárias para o perfil do IAM na política. Se você não usa esse recurso, remova essas permissões da política do IAM.
- Você pode restringir um perfil do IAM para acessar dados do S3 somente se as credenciais forem fornecidas pela funcionalidade Concessões de Acesso do S3. Este exemplo mostra como adicionar uma instrução `Condition` para uma instância específica da funcionalidade Concessões de Acesso do S3. Para fazer isso, substitua o ARN da instância da funcionalidade Concessões de Acesso do S3 na declaração de condição pelo ARN da instância da funcionalidade Concessões de Acesso do S3, que tem o formato `arn:aws:s3:region:accountId:access-grants/default`.

iam-policy.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ObjectLevelReadPermissions",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectVersionAcl",
        "s3:ListMultipartUploadParts"
      ],
      "Resource": [
        "arn:aws:s3:::*"
      ]
    }
  ]
}
```

```

    ],
    "Condition":{
      "StringEquals": { "aws:ResourceAccount": "accountId" },
      "ArnEquals": {
        "s3:AccessGrantsInstanceArn": ["arn:aws:s3:região:accountId:access-
grants/default"]
      }
    }
  },
  {
    "Sid": "ObjectLevelWritePermissions",
    "Effect":"Allow",
    "Action":[
      "s3:PutObject",
      "s3:PutObjectAcl",
      "s3:PutObjectVersionAcl",
      "s3>DeleteObject",
      "s3>DeleteObjectVersion",
      "s3:AbortMultipartUpload"
    ],
    "Resource":[
      "arn:aws:s3:::*"
    ],
    "Condition":{
      "StringEquals": { "aws:ResourceAccount": "accountId" },
      "ArnEquals": {
        "s3:AccessGrantsInstanceArn": ["arn:aws:s3:Região da
AWS:accountId:access-grants/default"]
      }
    }
  },
  {
    "Sid": "BucketLevelReadPermissions",
    "Effect":"Allow",
    "Action":[
      "s3:ListBucket"
    ],
    "Resource":[
      "arn:aws:s3:::*"
    ],
    "Condition":{
      "StringEquals": { "aws:ResourceAccount": "accountId" },
      "ArnEquals": {

```

```

        "s3:AccessGrantsInstanceArn": ["arn:aws:s3:Região da
AWS:accountId:access-grants/default"]
    }
}
},
{
    "Sid": "KMSPermissions",
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": [
        "*"
    ]
}
]
}

```

Você pode registrar um local na instância da funcionalidade Concessões de Acesso do S3 usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI), a API REST do Amazon S3 ou os AWS SDKs.

Uso do console do S3

Antes de poder conceder acesso aos dados do S3 com a funcionalidade Concessões de Acesso do S3, é necessário ter pelo menos um local registrado.

Como registrar um local em uma instância da funcionalidade Concessões 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 Grants.
3. Na página Concessões de Acesso do S3, escolha a região que contém a instância da funcionalidade Concessões de Acesso do S3 com a qual você deseja trabalhar.

Se você estiver usando a instância da funcionalidade Concessões de Acesso do S3 pela primeira vez, conclua a [Etapa 1: Criar uma instância da funcionalidade Concessões de Acesso do S3](#) e navegue até a Etapa 2 do assistente Configurar instância do Access Grants. Se você já

tiver uma instância da funcionalidade Concessões de Acesso do S3, escolha Visualizar detalhes, depois selecione Registrar local na guia Locais.

- a. Em Escopo do local, escolha Procurar no S3 ou insira o caminho do URI do S3 para o local que você deseja registrar. Para conferir os formatos de URI do S3, consulte a tabela de [formatos de local](#). Depois de inserir um URI, você pode escolher Visualizar para procurar o local.
- b. Em Perfil do IAM, escolha uma das seguintes opções:

- Escolher entre perfis do IAM existentes

Escolha um perfil do IAM na lista suspensa. Depois de escolher um perfil, selecione Visualizar para garantir que esse perfil tenha as permissões necessárias para gerenciar o local que você está registrando. Especificamente, garanta que esse perfil conceda à funcionalidade Concessões de Acesso do S3 as permissões `sts:AssumeRole` e `sts:SetSourceIdentity`.

- Inserir ARN do perfil do IAM

Navegue até o [console do IAM](#). Copie o nome do recurso da Amazon (ARN) do perfil do IAM e cole-o nesta caixa.

- c. Para finalizar, escolha Próximo ou Registrar local.

4. Solucionar problemas:

Não é possível registrar o local

- O local pode já estar registrado.

Talvez você não tenha a permissão `s3:CreateAccessGrantsLocation` para registrar locais. Entre em contato com o administrador da conta.

Usando a AWS CLI

Para instalar a AWS CLI, consulte [Instalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Você pode registrar o local padrão `s3://` ou um local personalizado na instância da funcionalidade Concessões de Acesso do S3. Primeiro, crie um perfil do IAM com acesso de entidade principal ao local, depois conceda permissão à funcionalidade Concessões de Acesso do S3 para assumir esse perfil.

Para usar os comandos a seguir, substitua os *user input placeholders* por suas próprias informações.

Example Criar uma política de recursos

Crie uma política que permita que a funcionalidade Concessões de Acesso do S3 assuma o perfil do IAM. Para fazer isso, você pode criar um arquivo JSON que contém as instruções a seguir. Para adicionar a política de recursos à sua conta, consulte [Criar e anexar sua primeira política gerenciada pelo cliente](#).

TestRolePolicy.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1234567891011",
      "Action": ["sts:AssumeRole", "sts:SetSourceIdentity"],
      "Effect": "Allow",
      "Principal": {"Service": "access-grants.s3.amazonaws.com"}
    }
  ]
}
```

Example Criar o perfil

Execute o comando do IAM a seguir para criar o perfil.

```
aws iam create-role --role-name accessGrantsTestRole \
  --region us-east-2 \
  --assume-role-policy-document file://TestRolePolicy.json
```

A execução do comando `create-role` retorna a política:

```
{
  "Role": {
    "Path": "/",
    "RoleName": "accessGrantsTestRole",
    "RoleId": "AROASRDGX4WM4GH55GIDA",
    "Arn": "arn:aws:iam::111122223333:role/accessGrantsTestRole",
```

```
"CreateDate": "2023-05-31T18:11:06+00:00",
"AssumeRolePolicyDocument": {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1685556427189",
      "Action": [
        "sts:AssumeRole",
        "sts:SetSourceIdentity"
      ],
      "Effect": "Allow",
      "Principal": {
        "Service": "access-grants.s3.amazonaws.com"
      }
    }
  ]
}
```

Example

Crie uma política do IAM para anexar permissões do Amazon S3 ao perfil do IAM. Confira o exemplo de arquivo `iam-policy.json` a seguir e substitua os *user input placeholders* por suas próprias informações.

Note

Se você usa criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) para criptografar os dados, o exemplo a seguir adiciona as permissões do AWS KMS necessárias para o perfil do IAM na política. Se você não usa esse recurso, remova essas permissões da política do IAM.

Para garantir que o perfil do IAM só possa ser usado para acessar dados no S3 se as credenciais forem fornecidas pela funcionalidade Concessões de Acesso do S3, esse exemplo mostra como adicionar uma instrução `Condition` que especifica a instância da funcionalidade Concessões de Acesso do S3 (`s3:AccessGrantsInstance:InstanceArn`) na política do IAM. Ao usar o exemplo de política a seguir, substitua os *user input placeholders* por suas próprias informações.

iam-policy.json


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ObjectLevelReadPermissions",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectVersionAcl",
        "s3:ListMultipartUploadParts"
      ],
      "Resource": [
        "arn:aws:s3:::*"
      ],
      "Condition": {
        "StringEquals": { "aws:ResourceAccount": "accountId" },
        "ArnEquals": {
          "s3:AccessGrantsInstanceArn": ["arn:aws:s3:region:accountId:access-  
grants/default"]
        }
      }
    },
    {
      "Sid": "ObjectLevelWritePermissions",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionAcl",
        "s3>DeleteObject",
        "s3>DeleteObjectVersion",
        "s3:AbortMultipartUpload"
      ],
      "Resource": [
        "arn:aws:s3:::*"
      ],
      "Condition": {
        "StringEquals": { "aws:ResourceAccount": "accountId" },
        "ArnEquals": {
```

```

        "s3:AccessGrantsInstanceArn": ["arn:aws:s3:Região da
AWS:accountId:access-grants/default"]
    }
}
},
{
    "Sid": "BucketLevelReadPermissions",
    "Effect": "Allow",
    "Action": [
        "s3:ListBucket"
    ],
    "Resource": [
        "arn:aws:s3:::*"
    ],
    "Condition": {
        "StringEquals": { "aws:ResourceAccount": "accountId" },
        "ArnEquals": {
            "s3:AccessGrantsInstanceArn": ["arn:aws:s3:Região da
AWS:accountId:access-grants/default"]
        }
    }
},
{
    "Sid": "KMSPermissions",
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": [
        "*"
    ]
}
]
}

```

Example

Execute o seguinte comando:

```

aws iam put-role-policy \
--role-name accessGrantsTestRole \
--policy-name accessGrantsTestRole \

```

```
--policy-document file://iam-policy.json
```

Exemplo Registrar o local padrão

```
aws s3control create-access-grants-location \  
  --account-id 111122223333 \  
  --location-scope s3:// \  
  --iam-role-arn arn:aws:iam::111122223333:role/accessGrantsTestRole
```

Resposta:

```
{"CreatedAt": "2023-05-31T18:23:48.107000+00:00",  
  "AccessGrantsLocationId": "default",  
  "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/  
default/location/default",  
  "LocationScope": "s3://"  
  "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"  
}
```

Exemplo Registrar um local personalizado

```
aws s3control create-access-grants-location \  
  --account-id 111122223333 \  
  --location-scope s3://DOC-BUCKET-EXAMPLE/ \  
  --iam-role-arn arn:aws:iam::123456789012:role/accessGrantsTestRole
```

Resposta:

```
{"CreatedAt": "2023-05-31T18:23:48.107000+00:00",  
  "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb456",  
  "AccessGrantsLocationArn": "arn:aws:s3:us-east-2: 111122223333:access-grants/  
default/location/635f1139-1af2-4e43-8131-a4de006eb888",  
  "LocationScope": "s3://DOC-BUCKET-EXAMPLE/",  
  "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"  
}
```

Uso da API REST

Para obter informações sobre o suporte à API REST do Amazon S3 para gerenciar instâncias da funcionalidade Concessões de Acesso do S3, consulte as seguintes seções na Referência da API do Amazon Simple Storage Service:

- [CreateAccessGrantsLocation](#)
- [DeleteAccessGrantsLocation](#)
- [GetAccessGrantsLocation](#)
- [ListAccessGrantsLocations](#)
- [UpdateAccessGrantsLocation](#)

Uso dos AWS SDKs

Esta seção fornece exemplos de como registrar locais usando os AWS SDKs.

Para usar os exemplos a seguir, substitua *user input placeholders* por suas próprias informações.

Java

Você pode registrar o local padrão `s3://` ou um local personalizado na instância da funcionalidade Concessões de Acesso do S3. Primeiro, crie um perfil do IAM com acesso de entidade principal ao local, depois conceda permissão à funcionalidade Concessões de Acesso do S3 para assumir esse perfil.

Para usar os comandos a seguir, substitua os *user input placeholders* por suas próprias informações.

Example Registrar um local padrão

Solicitação:

```
public void createAccessGrantsLocation() {
    CreateAccessGrantsLocationRequest createRequest =
        CreateAccessGrantsLocationRequest.builder()
            .accountId("111122223333")
            .locationScope("s3://")
            .iamRoleArn("arn:aws:iam::123456789012:role/accessGrantsTestRole")
            .build();
    CreateAccessGrantsLocationResponse createResponse =
        s3Control.createAccessGrantsLocation(createRequest);
    LOGGER.info("CreateAccessGrantsLocationResponse: " + createResponse);
}
```

Resposta:

```

CreateAccessGrantsLocationResponse(
  CreatedAt=2023-06-07T04:35:11.027Z,
  AccessGrantsLocationId=default,
  AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
  location/default,
  LocationScope=s3://,
  IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)

```

Example Registrar um local personalizado

Solicitação:

```

public void createAccessGrantsLocation() {
  CreateAccessGrantsLocationRequest createRequest =
    CreateAccessGrantsLocationRequest.builder()
      .accountId("111122223333")
      .locationScope("s3://DOC-BUCKET-EXAMPLE/")
      .iamRoleArn("arn:aws:iam::111122223333:role/accessGrantsTestRole")
      .build();
  CreateAccessGrantsLocationResponse createResponse =
    s3Control.createAccessGrantsLocation(createRequest);
  LOGGER.info("CreateAccessGrantsLocationResponse: " + createResponse);
}

```

Resposta:

```

CreateAccessGrantsLocationResponse(
  CreatedAt=2023-06-07T04:35:10.027Z,
  AccessGrantsLocationId=18cfe6fb-eb5a-4ac5-aba9-8d79f04c2012,
  AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
  location/18cfe6fb-eb5a-4ac5-aba9-8d79f04c2666,
  LocationScope= s3://test-bucket-access-grants-user123/,
  IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)

```

Tópicos

- [Visualizar os detalhes de um local registrado](#)
- [Atualizar um local registrado](#)
- [Excluir um local registrado](#)

Visualizar os detalhes de um local registrado

Você pode obter os detalhes de um local registrado na instância da funcionalidade Concessões de Acesso do S3 usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI), a API REST do Amazon S3 e os AWS SDKs.

Uso do console do S3

Como visualizar os locais registrados na instância da funcionalidade Concessões 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 Grants.
3. Na página Concessões de Acesso do S3, escolha a região que contém a instância da funcionalidade Concessões de Acesso do S3 com a qual você deseja trabalhar.
4. Escolha Visualizar detalhes da instância.
5. Na página de detalhes da instância, escolha a guia Locais.
6. Encontre o local registrado que você deseja visualizar. Para filtrar a lista de locais registrados, use a caixa de pesquisa.

Usando a AWS CLI

Para instalar a AWS CLI, consulte [Instalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Para usar o comando a seguir, substitua os *user input placeholders* por suas próprias informações.

Example : obter os detalhes de um local registrado

```
aws s3control get-access-grants-location \  
--account-id 111122223333 \  
--access-grants-location-id default
```

Resposta:

```
{  
  "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
```

```

    "AccessGrantsLocationId": "default",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/location/default",
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
  }

```

Example : listar todos os locais registrados em uma instância da funcionalidade Concessões de Acesso do S3

Para restringir os resultados a um prefixo ou bucket do S3, há a opção de usar o parâmetro `--location-scope s3://bucket-and-or-prefix`.

```

aws s3control list-access-grants-locations \
--account-id 111122223333 \
--region us-east-2

```

Resposta:

```

{"AccessGrantsLocationsList": [
  {
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "default",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/location/default",
    "LocationScope": "s3://"
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
  },
  {
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb456",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/location/635f1139-1af2-4e43-8131-a4de006eb888",
    "LocationScope": "s3://DOC-EXAMPLE-BUCKET/prefixA*",
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
  }
]
}

```

Uso da API REST

Para obter informações sobre o suporte à API REST do Amazon S3 para obter os detalhes de um local registrado ou listar todos os locais registrados em uma instância da funcionalidade Concessões

de Acesso do S3, consulte as seguintes seções na Referência da API do Amazon Simple Storage Service:

- [GetAccessGrantsLocation](#)
- [ListAccessGrantsLocations](#)

Uso dos AWS SDKs

Esta seção fornece exemplos de como obter os detalhes de um local registrado ou listar todos os locais registrados em uma instância da funcionalidade Concessões de Acesso do S3 usando os AWS SDKs.

Para usar os exemplos a seguir, substitua *user input placeholders* por suas próprias informações.

Java

Example : obter os detalhes de um local registrado

```
public void getAccessGrantsLocation() {
    GetAccessGrantsLocationRequest getAccessGrantsLocationRequest =
        GetAccessGrantsLocationRequest.builder()
            .accountId("111122223333")
            .accessGrantsLocationId("default")
            .build();
    GetAccessGrantsLocationResponse getAccessGrantsLocationResponse =
        s3Control.getAccessGrantsLocation(getAccessGrantsLocationRequest);
    LOGGER.info("GetAccessGrantsLocationResponse: " + getAccessGrantsLocationResponse);
}
```

Resposta:

```
GetAccessGrantsLocationResponse(
    CreatedAt=2023-06-07T04:35:10.027Z,
    AccessGrantsLocationId=default,
    AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
    location/default,
    LocationScope= s3://,
    IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
```


Example : listar todos os locais registrados em uma instância da funcionalidade Concessões de Acesso do S3

Para restringir os resultados a um prefixo ou bucket do S3, há a opção de enviar um URI do S3, como `s3://bucket-and-or-prefix`, no parâmetro LocationScope.

```
public void listAccessGrantsLocations() {

    ListAccessGrantsLocationsRequest listRequest =
        ListAccessGrantsLocationsRequest.builder()
            .accountId("111122223333")
            .build();

    ListAccessGrantsLocationsResponse listResponse =
        s3Control.listAccessGrantsLocations(listRequest);
    LOGGER.info("ListAccessGrantsLocationsResponse: " + listResponse);
}
```

Resposta:

```
ListAccessGrantsLocationsResponse(
    AccessGrantsLocationsList=[
    ListAccessGrantsLocationsEntry(
        CreatedAt=2023-06-07T04:35:11.027Z,
        AccessGrantsLocationId=default,
        AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
        location/default,
        LocationScope=s3://,
        IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
    ),
    ListAccessGrantsLocationsEntry(
        CreatedAt=2023-06-07T04:35:10.027Z,
        AccessGrantsLocationId=635f1139-1af2-4e43-8131-a4de006eb456,
        AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
        location/635f1139-1af2-4e43-8131-a4de006eb888,
        LocationScope=s3://DOC-EXAMPLE-BUCKET/prefixA*,
        IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
    )
    ]
)
```

Atualizar um local registrado

Você pode atualizar o perfil do AWS Identity and Access Management (IAM) de um local registrado na instância da funcionalidade Concessões de Acesso do Amazon S3. Para cada novo perfil do IAM que você usa para registrar um local na funcionalidade Concessões de Acesso do S3, conceda à entidade principal de serviço da funcionalidade Concessões de Acesso do S3 (`access-grants.s3.amazonaws.com`) acesso a esse perfil. Para isso, adicione uma entrada para o novo perfil do IAM no mesmo arquivo JSON de política de confiança que você usou quando [registrou o local](#) pela primeira vez.

Você pode atualizar um local na instância da funcionalidade Concessões de Acesso do S3 usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI), a API REST do Amazon S3 e os AWS SDKs.

Uso do console do S3

Como atualizar o perfil do IAM de um local registrado na instância da funcionalidade Concessões 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 Grants.
3. Na página Concessões de Acesso do S3, escolha a região que contém a instância da funcionalidade Concessões de Acesso do S3 com a qual você deseja trabalhar.
4. Escolha Visualizar detalhes da instância.
5. Na página de detalhes da instância, escolha a guia Locais.
6. Encontre o local que deseja atualizar. Para filtrar a lista de locais, use a caixa de pesquisa.
7. Selecione o botão de opções ao lado do local registrado que deseja atualizar.
8. Atualize o perfil do IAM e escolha Salvar alterações.

Usando a AWS CLI

Para instalar a AWS CLI, consulte [Instalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Para usar o comando a seguir, substitua os *user input placeholders* por suas próprias informações.

Example : atualizar o perfil do IAM de um local registrado

```
aws s3control update-access-grants-location \  
--account-id 111122223333 \  
--access-grants-location-id 635f1139-1af2-4e43-8131-a4de006eb999 \  
--iam-role-arn arn:aws:iam::777788889999:role/accessGrantsTestRole
```

Resposta:

```
{  
  "CreatedAt": "2023-05-31T18:23:48.107000+00:00",  
  "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb999",  
  "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:777788889999:access-grants/  
default/location/635f1139-1af2-4e43-8131-a4de006eb888",  
  "LocationScope": "s3://DOC-EXAMPLE-BUCKET/prefixB*",  
  "IAMRoleArn": "arn:aws:iam::777788889999:role/accessGrantsTestRole"  
}
```

Uso da API REST

Para obter informações sobre o suporte à API REST do Amazon S3 para atualizar um local na instância da funcionalidade Concessões de Acesso do S3, consulte [UpdateAccessGrantsLocation](#) na Referência da API do Amazon Simple Storage Service.

Uso dos AWS SDKs

Esta seção fornece exemplos de como atualizar o perfil do IAM de um local registrado usando os AWS SDKs.

Para usar o exemplo a seguir, substitua os *user input placeholders* por suas próprias informações.

Java

Example : atualizar o perfil do IAM de um local registrado

```
public void updateAccessGrantsLocation() {  
    UpdateAccessGrantsLocationRequest updateRequest =  
        UpdateAccessGrantsLocationRequest.builder()  
            .accountId("111122223333")  
            .accessGrantsLocationId("635f1139-1af2-4e43-8131-a4de006eb999")  
            .iamRoleArn("arn:aws:iam::777788889999:role/accessGrantsTestRole")  
            .build();
```

```
UpdateAccessGrantsLocationResponse updateResponse =
    s3Control.updateAccessGrantsLocation(updateRequest);
LOGGER.info("UpdateAccessGrantsLocationResponse: " + updateResponse);
}
```

Resposta:

```
UpdateAccessGrantsLocationResponse(
    CreatedAt=2023-06-07T04:35:10.027Z,
    AccessGrantsLocationId=635f1139-1af2-4e43-8131-a4de006eb999,
    AccessGrantsLocationArn=arn:aws:s3:us-east-2:777788889999:access-grants/default/
    location/635f1139-1af2-4e43-8131-a4de006eb888,
    LocationScope=s3://DOC-EXAMPLE-BUCKET/prefixB*,
    IAMRoleArn=arn:aws:iam::777788889999:role/accessGrantsTestRole
)
```

Excluir um local registrado

Você pode excluir um registro de local de uma instância da funcionalidade Concessões de Acesso do Amazon S3. A exclusão do local cancela o registro na instância da funcionalidade Concessões de Acesso do S3.

Antes de remover um registro de local de uma instância da funcionalidade Concessões de Acesso do S3, é necessário excluir todas as concessões associadas a esse local. Para obter informações sobre como excluir concessões, consulte [Delete a grant](#).

Você pode excluir um local na instância da funcionalidade Concessões de Acesso do S3 usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI), a API REST do Amazon S3 e os AWS SDKs.

Uso do console do S3

Como excluir um registro de local de uma instância da funcionalidade Concessões 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 Grants.
3. Na página Concessões de Acesso do S3, escolha a região que contém a instância da funcionalidade Concessões de Acesso do S3 com a qual você deseja trabalhar.
4. Escolha Visualizar detalhes da instância.

5. Na página de detalhes da instância, escolha a guia Locais.
6. Encontre o local que deseja atualizar. Para filtrar a lista de locais, use a caixa de pesquisa.
7. Selecione o botão de opções ao lado do local registrado que deseja excluir.
8. Escolha Cancelar registro.
9. É exibida uma caixa de diálogo avisando que essa ação não poderá ser desfeita. Para excluir o local, escolha Cancelar registro.

Usando a AWS CLI

Para instalar a AWS CLI, consulte [Instalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Para usar o comando a seguir, substitua os *user input placeholders* por suas próprias informações.

Example : excluir um registro de local

```
aws s3control delete-access-grants-location \  
--account-id 111122223333 \  
--access-grants-location-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  
// No response body
```

Uso da API REST

Para obter informações sobre o suporte à API REST do Amazon S3 para excluir um local de uma instância da funcionalidade Concessões de Acesso do S3, consulte [DeleteAccessGrantsLocation](#) na Referência da API do Amazon Simple Storage Service.

Uso dos AWS SDKs

Esta seção fornece um exemplo de como excluir um local usando os AWS SDKs.

Para usar o exemplo a seguir, substitua os *user input placeholders* por suas próprias informações.

Java

Example : excluir um registro de local

```
public void deleteAccessGrantsLocation() {
```

```
DeleteAccessGrantsLocationRequest deleteRequest =
    DeleteAccessGrantsLocationRequest.builder()
        .accountId("111122223333")
        .accessGrantsLocationId("a1b2c3d4-5678-90ab-cdef-EXAMPLE11111")
        .build();
DeleteAccessGrantsLocationResponse deleteResponse =
    s3Control.deleteAccessGrantsLocation(deleteRequest);
LOGGER.info("DeleteAccessGrantsLocationResponse: " + deleteResponse);
}
```

Resposta:

```
DeleteAccessGrantsLocationResponse()
```

Criar concessões

Depois de [registrar pelo menos um local](#) na instância da funcionalidade Concessões de Acesso do Amazon S3, você poderá criar uma concessão de acesso. Uma concessão de acesso dá ao beneficiário permissão para acessar um local registrado.

O beneficiário pode ser um usuário ou perfil do AWS Identity and Access Management (IAM) ou um usuário ou grupo de diretório. Um usuário de diretório é um usuário do seu diretório corporativo ou fonte de identidades externa que você [adicionou à instância do AWS IAM Identity Center](#) que está [associada à instância da funcionalidade Concessões de Acesso do S3](#). Para criar uma concessão para um usuário ou grupo específico do Centro de Identidade do IAM, encontre o GUID usado para identificar esse usuário no Centro de Identidade do IAM, por exemplo, a1b2c3d4-5678-90ab-cdef-EXAMPLE11111.

Você pode conceder acesso a um bucket, prefixo ou objeto. Um prefixo no Amazon S3 é uma string de caracteres no início de um nome de chave de objeto que é usado para organizar os objetos dentro de um bucket. Pode ser qualquer string de caracteres permitidos, por exemplo, nomes de chave de objeto em um bucket que comece com o prefixo `engineering/`.

Subprefixo

Ao conceder acesso a um local registrado, você pode usar o campo `Subprefix` para restringir o escopo a um prefixo ou objeto específico em um bucket.

Não é possível criar uma concessão de acesso para o local padrão `s3://`, o que concederia ao beneficiário acesso a todos os buckets em uma região. Se você escolher o local `s3://` padrão como

local da concessão, deverá restringir o escopo da concessão usando o campo `Subprefix` para especificar uma das seguintes opções:

- Um bucket: `s3://bucket/*`
- Um prefixo em um bucket: `s3://bucket/prefix*`
- Um prefixo em um prefixo: `s3://bucket/prefixA/prefixB*`
- Um objeto: `s3://bucket/object-key-name`

Se você criar uma concessão de acesso em que o local registrado seja um bucket, poderá enviar uma das seguintes opções no campo `Subprefix`:

- Um prefixo no bucket: `prefix*`
- Um prefixo em um prefixo: `prefixA/prefixB*`
- Um objeto: `/object-key-name`

O escopo da concessão mostrado no console do Amazon S3 ou o `GrantScope` que é retornado na resposta da API ou AWS Command Line Interface (AWS CLI) é o resultado da concatenação do caminho do local com o `Subprefix`. Garanta que esse caminho concatenado seja mapeado corretamente para o bucket, prefixo ou objeto do S3 ao qual você deseja conceder acesso.

Se você estiver criando uma concessão de acesso que dá acesso a apenas um objeto, especifique na chamada de API ou no comando da CLI que `s3PrefixType` é `Object`.

Note

Não é possível criar uma concessão para um bucket que ainda não existe. No entanto, é possível criar uma concessão para um prefixo que ainda não existe.

Você pode criar uma concessão de acesso usando o console do Amazon S3, a AWS CLI, a API REST do Amazon S3 ou os AWS SDKs.

Uso do console do S3

Como criar uma concessão 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 Grants.
3. Na página Concessões de Acesso do S3, escolha a região que contém a instância da funcionalidade Concessões de Acesso do S3 com a qual você deseja trabalhar.

Se você estiver usando a instância da funcionalidade Concessões de Acesso do S3 pela primeira vez, conclua a [Etapa 2: Registrar um local](#) e navegue até a Etapa 3 do assistente Configurar instância do Access Grants. Se você já tiver uma instância da funcionalidade Concessões de Acesso do S3, escolha Visualizar detalhes, depois selecione Criar concessão na guia Concessões.

- a. Na seção Escopo da concessão, selecione ou insira um local registrado.

Se você selecionou o local `s3://` padrão, use a caixa Subprefixo para restringir o escopo da concessão de acesso. Para obter mais informações, consulte [Subprefix](#). Se você estiver concedendo acesso somente a um objeto, selecione O escopo da concessão é um objeto.

- b. Em Permissões e acesso, selecione o nível de Permissão: Leitura, Gravação ou ambos.

Depois, escolha o Tipo de beneficiário. Se você adicionou seu diretório corporativo ao Centro de Identidade do IAM e associou essa instância do Centro de Identidade do IAM à instância da funcionalidade Concessões de Acesso do S3, poderá escolher a opção Identidade de diretório do Centro de Identidade do IAM. Se você escolher essa opção, obtenha o ID do usuário ou grupo no Centro de Identidade do IAM e insira-o nesta seção.

Se o Tipo de beneficiário for um usuário ou perfil do IAM, escolha Entidade principal do IAM. Em Tipo de entidade principal do IAM, escolha Usuário ou Perfil. Depois, em Usuário da entidade principal do IAM, escolha na lista ou insira o ID da identidade.

- c. Para criar a concessão da funcionalidade Concessões de Acesso do S3, escolha Próximo ou Criar concessão.

4. Se as opções Próximo ou Criar concessão estiverem desabilitadas:

Não é possível criar concessão

- Poderá ser necessário [registrar um local](#) na instância da funcionalidade Concessões de Acesso do S3 primeiro.
- Talvez você não tenha a permissão `s3:CreateAccessGrant` para criar uma concessão de acesso. Entre em contato com o administrador da conta.

Usando a AWS CLI

Para instalar a AWS CLI, consulte [Instalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Os exemplos a seguir mostram como criar uma solicitação de concessão de acesso para uma entidade principal do IAM e como criar uma solicitação de concessão de acesso para um usuário ou grupo de diretório corporativo.

Para usar os comandos a seguir, substitua os *user input placeholders* por suas próprias informações.

Note

Se você estiver criando uma concessão de acesso que dá acesso a apenas um objeto, inclua o parâmetro `--s3-prefix-type Object` necessário.

Example Criar uma solicitação de concessão de acesso para uma entidade principal do IAM

```
aws s3control create-access-grant \  
--account-id 111122223333 \  
--access-grants-location-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \  
--access-grants-location-configuration S3SubPrefix=prefixB* \  
--permission READ \  
--grantee GranteeType=IAM,GranteeIdentifier=arn:aws:iam::123456789012:user/data-consumer-3
```

Example Criar uma resposta de concessão de acesso

```
{"CreatedAt": "2023-05-31T18:41:34.663000+00:00",  
  "AccessGrantId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/  
grant/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "Grantee": {  
    "GranteeType": "IAM",  
    "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"  
  },  
  "AccessGrantsLocationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "AccessGrantsLocationConfiguration": {  
    "S3SubPrefix": "prefixB*"br/>  },  
}
```

```
"GrantScope": "s3://DOC-BUCKET-EXAMPLE/prefix*",  
"Permission": "READ"  
}
```

Criar uma solicitação de concessão de acesso para um usuário ou grupo de diretório

Para criar uma solicitação de concessão de acesso para um usuário ou grupo de diretório, primeiro é necessário obter o GUID do usuário ou grupo de diretório por meio da execução de um dos comandos a seguir.

Example Obter o GUID de um usuário ou grupo de diretório

Você pode encontrar o GUID de um usuário do Centro de Identidade do IAM por meio do console do Centro de Identidade do IAM ou usando a AWS CLI ou os AWS SDKs. O comando a seguir lista os usuários na instância especificada do Centro de Identidade do IAM, com os respectivos nomes e identificadores.

```
aws identitystore list-users --identity-store-id d-1a2b3c4d1234
```

Esse comando lista os grupos na instância especificada do Centro de Identidade do IAM.

```
aws identitystore list-groups --identity-store-id d-1a2b3c4d1234
```

Example Criar uma concessão de acesso para um usuário ou grupo de diretório

Esse comando é semelhante à criação de uma concessão para usuários ou perfis do IAM, exceto que o tipo de beneficiário é `DIRECTORY_USER` ou `DIRECTORY_GROUP` e o identificador do beneficiário é o GUID do usuário ou grupo de diretório.

```
aws s3control create-access-grant \  
--account-id 123456789012 \  
--access-grants-location-id default \  
--access-grants-location-configuration S3SubPrefix="DOC-EXAMPLE-BUCKET/rafael/*" \  
--permission READWRITE \  
--grantee GranteeType=DIRECTORY_USER,GranteeIdentifier=83d43802-00b1-7054-db02-f1d683aacba5 \  

```

Uso da API REST

Para obter informações sobre o suporte à API REST do Amazon S3 para gerenciar concessões de acesso, consulte as seguintes seções na Referência da API do Amazon Simple Storage Service:

- [CreateAccessGrant](#)
- [DeleteAccessGrant](#)
- [GetAccessGrant](#)
- [ListAccessGrants](#)

Uso dos AWS SDKs

Esta seção fornece exemplos de como criar uma concessão de acesso usando os AWS SDKs.

Java

Para usar o exemplo a seguir, substitua *user input placeholders* por suas próprias informações.

Note

Se você estiver criando uma concessão de acesso que dá acesso a apenas um objeto, inclua o parâmetro `.s3PrefixType(S3PrefixType.Object)` necessário.

Example Criar uma solicitação de concessão de acesso

```
public void createAccessGrant() {
    CreateAccessGrantRequest createRequest = CreateAccessGrantRequest.builder()
        .accountId("111122223333")
        .accessGrantsLocationId("a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaa")
        .permission("READ")
        .accessGrantsLocationConfiguration(AccessGrantsLocationConfiguration.builder().s3SubPrefix("
        .grantee(Grantee.builder().granteeType("IAM").granteeIdentifier("arn:aws:iam::111122223333:u
        data-consumer-3").build())
        .build();
    CreateAccessGrantResponse createResponse =
        s3Control.createAccessGrant(createRequest);
    LOGGER.info("CreateAccessGrantResponse: " + createResponse);
}
```

Example Criar uma resposta de concessão de acesso

```
CreateAccessGrantResponse(
    CreatedAt=2023-06-07T05:20:26.330Z,
```

```
AccessGrantId=a1b2c3d4-5678-90ab-cdef-EXAMPLE33333,  
AccessGrantArn=arn:aws:s3:us-east-2:444455556666:access-grants/default/grant/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE33333,  
Grantee=Grantee(  
  GranteeType=IAM,  
  GranteeIdentifier=arn:aws:iam::111122223333:user/data-consumer-3  
)  
,  
AccessGrantsLocationId=a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa,  
AccessGrantsLocationConfiguration=AccessGrantsLocationConfiguration(  
  S3SubPrefix=prefixB*  
)  
,  
GrantScope=s3://DOC-BUCKET-EXAMPLE/prefixB,  
Permission=READ  
)
```

Tópicos

- [Visualizar uma concessão](#)
- [Excluir uma concessão](#)

Visualizar uma concessão

Você pode ver os detalhes de uma concessão de acesso na instância da funcionalidade Concessões de Acesso do S3 usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI), a API REST do Amazon S3 e os AWS SDKs.

Uso do console do S3

Como visualizar os detalhes de uma concessão 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 Grants.
3. Na página Concessões de Acesso do S3, escolha a região que contém a instância da funcionalidade Concessões de Acesso do S3 com a qual você deseja trabalhar.
4. Escolha Visualizar detalhes da instância.
5. Na página de detalhes, escolha a guia Concessões.
6. Na seção Concessões, encontre a concessão de acesso que deseja visualizar. Use a caixa de pesquisa para filtrar a lista de concessões.

Usando a AWS CLI

Para instalar a AWS CLI, consulte [Instalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Para usar os comandos a seguir, substitua os *user input placeholders* por suas próprias informações.

Example : visualizar os detalhes de uma concessão de acesso

```
aws s3control get-access-grant \  
--account-id 111122223333 \  
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Resposta:

```
{  
  "CreatedAt": "2023-05-31T18:41:34.663000+00:00",  
  "AccessGrantId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/  
grant-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "Grantee": {  
    "GranteeType": "IAM",  
    "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"  
  },  
  "Permission": "READ",  
  "AccessGrantsLocationId": "12a6710f-5af8-41f5-b035-0bc795bf1a2b",  
  "AccessGrantsLocationConfiguration": {  
    "S3SubPrefix": "prefixB*"  
  },  
  "GrantScope": "s3://DOC-EXAMPLE-BUCKET/"  
}
```

Example : listar todas as concessões de acesso em uma instância da funcionalidade Concessões de Acesso do S3

Opcionalmente, você pode usar os seguintes parâmetros para restringir os resultados a um prefixo do S3 ou a uma identidade do AWS Identity and Access Management (IAM):

- Subprefixo: `--grant-scope s3://bucket-name/prefix*`
- Identidade do IAM: `--grantee-type IAM` e `--grantee-identifier arn:aws:iam::123456789000:role/accessGrantsConsumerRole`

```
aws s3control list-access-grants \
--account-id 111122223333
```

Resposta:

```
{
  "AccessGrantsList": [
    {
      "CreatedAt": "2023-06-14T17:54:46.542000+00:00",
      "AccessGrantId": "dd8dd089-b224-4d82-95f6-975b4185bbaa",
      "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/grant/dd8dd089-b224-4d82-95f6-975b4185bbaa",
      "Grantee": {
        "GranteeType": "IAM",
        "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"
      },
      "Permission": "READ",
      "AccessGrantsLocationId": "23514a34-ea2e-4ddf-b425-d0d4bfcada1",
      "GrantScope": "s3://DOC-EXAMPLE-BUCKET/prefixA*"
    },
    {
      "CreatedAt": "2023-06-24T17:54:46.542000+00:00",
      "AccessGrantId": "ee8ee089-b224-4d72-85f6-975b4185a1b2",
      "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/grant/ee8ee089-b224-4d72-85f6-975b4185a1b2",
      "Grantee": {
        "GranteeType": "IAM",
        "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-9"
      },
      "Permission": "READ",
      "AccessGrantsLocationId": "12414a34-ea2e-4ddf-b425-d0d4bfcacao0",
      "GrantScope": "s3://DOC-EXAMPLE-BUCKET/prefixB*"
    }
  ]
}
```

Uso da API REST

Você pode usar as operações de API do Amazon S3 para visualizar os detalhes de uma concessão de acesso e listar todas as concessões de acesso em uma instância da funcionalidade Concessões de Acesso do S3. Para obter informações sobre o suporte à API REST para gerenciar concessões de acesso, consulte as seguintes seções na Referência da API do Amazon Simple Storage Service:

- [GetAccessGrant](#)

- [ListAccessGrants](#)

Uso dos AWS SDKs

Esta seção fornece exemplos de como obter os detalhes de uma concessão de acesso usando os AWS SDKs.

Para usar os exemplos a seguir, substitua *user input placeholders* por suas próprias informações.

Java

Example : visualizar os detalhes de uma concessão de acesso

```
public void getAccessGrant() {
    GetAccessGrantRequest getRequest = GetAccessGrantRequest.builder()
        .accountId("111122223333")
        .accessGrantId("a1b2c3d4-5678-90ab-cdef-EXAMPLE2222")
        .build();
    GetAccessGrantResponse getResponse = s3Control.getAccessGrant(getRequest);
    LOGGER.info("GetAccessGrantResponse: " + getResponse);
}
```

Resposta:

```
GetAccessGrantResponse(
    CreatedAt=2023-06-07T05:20:26.330Z,
    AccessGrantId=a1b2c3d4-5678-90ab-cdef-EXAMPLE2222,
    AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant-fd3a5086-42f7-4b34-9fad-472e2942c70e,
    Grantee=Grantee(
    GranteeType=IAM,
    GranteeIdentifier=arn:aws:iam::111122223333:user/data-consumer-3
    ),
    Permission=READ,
    AccessGrantsLocationId=12a6710f-5af8-41f5-b035-0bc795bf1a2b,
    AccessGrantsLocationConfiguration=AccessGrantsLocationConfiguration(
    S3SubPrefix=prefixB*
    ),
    GrantScope=s3://DOC-EXAMPLE-BUCKET/
)
```

Example : listar todas as concessões de acesso em uma instância da funcionalidade Concessões de Acesso do S3

Opcionalmente, você pode usar os seguintes parâmetros para restringir os resultados a um prefixo do S3 ou a uma identidade do IAM:

- Escopo: GrantScope=s3://*bucket-name/prefix**
- Beneficiário: GranteeType=IAM e GranteeIdentifier=arn:aws:iam::*111122223333*:role/*accessGrantsConsumerRole*

```
public void listAccessGrants() {
    ListAccessGrantsRequest listRequest = ListAccessGrantsRequest.builder()
        .accountId("111122223333")
        .build();
    ListAccessGrantsResponse listResponse = s3Control.listAccessGrants(listRequest);
    LOGGER.info("ListAccessGrantsResponse: " + listResponse);
}
```

Resposta:

```
ListAccessGrantsResponse(
    AccessGrantsList=[
        ListAccessGrantEntry(
            CreatedAt=2023-06-14T17:54:46.540z,
            AccessGrantId=dd8dd089-b224-4d82-95f6-975b4185bbaa,
            AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
            grant/dd8dd089-b224-4d82-95f6-975b4185bbaa,
            Grantee=Grantee(
                GranteeType=IAM, GranteeIdentifier= arn:aws:iam::111122223333:user/data-consumer-3
            ),
            Permission=READ,
            AccessGrantsLocationId=23514a34-ea2e-4ddf-b425-d0d4bfcada1,
            GrantScope=s3://DOC-EXAMPLE-BUCKET/prefixA
        ),
        ListAccessGrantEntry(
            CreatedAt=2023-06-24T17:54:46.540z,
            AccessGrantId=ee8ee089-b224-4d72-85f6-975b4185a1b2,
```



```
AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/ee8ee089-b224-4d72-85f6-975b4185a1b2,
Grantee=Grantee(
  GranteeType=IAM, GranteeIdentifier= arn:aws:iam::111122223333:user/data-consumer-9
),
Permission=READ,
AccessGrantsLocationId=12414a34-ea2e-4ddf-b425-d0d4bfcacao0,
GrantScope=s3://DOC-EXAMPLE-BUCKET/prefixB*
)
]
)
```

Excluir uma concessão

Você pode excluir concessões de acesso de uma instância da funcionalidade Concessões de Acesso do Amazon S3. Não é possível desfazer a exclusão de uma concessão de acesso. Depois de excluir uma concessão de acesso, o beneficiário perderá o acesso aos dados do Amazon S3.

Você pode excluir uma concessão de acesso usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI), a API REST do Amazon S3 ou os AWS SDKs.

Uso do console do S3

Como excluir uma concessão 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 Grants.
3. Na página Concessões de Acesso do S3, escolha a região que contém a instância da funcionalidade Concessões de Acesso do S3 com a qual você deseja trabalhar.
4. Escolha Visualizar detalhes da instância.
5. Na página de detalhes, escolha a guia Concessões.
6. Procure a concessão que deseja excluir. Ao encontrar a concessão, escolha o botão de opções ao lado dela.
7. Escolha Excluir. É exibida uma caixa de diálogo avisando que a ação não poderá ser desfeita. Escolha Excluir novamente para excluir a concessão.

Usando a AWS CLI

Para instalar a AWS CLI, consulte [Instalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Para usar o comando a seguir, substitua os *user input placeholders* por suas próprias informações.

Example : excluir uma concessão de acesso

```
aws s3control delete-access-grant \  
--account-id 111122223333 \  
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  
  
// No response body
```

Uso da API REST

Para obter informações sobre o suporte à API REST do Amazon S3 para gerenciar concessões de acesso, consulte [DeleteAccessGrant](#) na Referência da API do Amazon Simple Storage Service.

Uso dos AWS SDKs

Esta seção fornece exemplos de como excluir uma concessão de acesso usando os AWS SDKs. Para usar o exemplo a seguir, substitua os *user input placeholders* por suas próprias informações.

Java

Example : excluir uma concessão de acesso

```
public void deleteAccessGrant() {  
    DeleteAccessGrantRequest deleteRequest = DeleteAccessGrantRequest.builder()  
        .accountId("111122223333")  
        .accessGrantId("a1b2c3d4-5678-90ab-cdef-EXAMPLE11111")  
        .build();  
    DeleteAccessGrantResponse deleteResponse =  
        s3Control.deleteAccessGrant(deleteRequest);  
    LOGGER.info("DeleteAccessGrantResponse: " + deleteResponse);  
}
```

Resposta:

```
DeleteAccessGrantResponse()
```

Solicitar acesso aos dados do Amazon S3 por meio da funcionalidade Concessões de Acesso do S3

Depois de usar a funcionalidade Concessões de Acesso do Amazon S3 para [criar uma concessão de acesso](#) que dá acesso aos dados do S3 para as entidades principais do AWS Identity and Access Management (IAM), suas identidades de diretório corporativo ou as aplicações autorizadas, os beneficiários poderão solicitar credenciais para acessar esses dados.

Quando uma aplicação ou AWS service (Serviço da AWS) usa a operação de API `GetDataAccess` para pedir à funcionalidade Concessões de Acesso do S3 acesso aos dados do S3 em nome de um beneficiário, primeiro a funcionalidade Concessões de Acesso do S3 verifica se você concedeu acesso aos dados para essa entidade. Em seguida, a funcionalidade Concessões de Acesso do S3 usa a operação de API [AssumeRole](#) para obter um token de credencial temporário e o envia ao solicitante. Esse token temporário de credencial é um token do AWS Security Token Service (AWS STS).

A solicitação `GetDataAccess` deve incluir o parâmetro `target`, que especifica o escopo dos dados do S3 aos quais as credenciais temporárias se aplicam. Esse escopo `target` pode ser igual ao escopo da concessão ou um subconjunto desse escopo, mas o escopo `target` deve estar dentro do escopo da concessão concedida ao solicitante. A solicitação também deve especificar o parâmetro `permission` para indicar o nível de permissão das credenciais temporárias, seja `READ`, `WRITE` ou `READWRITE`.

O solicitante pode especificar o nível de privilégio do token temporário na solicitação de credencial. Usando o parâmetro `privilege`, o solicitante pode reduzir ou aumentar o escopo de acesso das credenciais temporárias, dentro dos limites do escopo da concessão. O valor padrão do parâmetro `privilege` é `Default`, o que significa que o escopo `target` da credencial retornada é o escopo original da concessão. O outro valor possível para `privilege` é `Minimal`. Se o escopo `target` for reduzido do escopo original da concessão, a credencial temporária será reduzida para corresponder ao escopo `target`, desde que o escopo `target` esteja dentro do escopo da concessão.

A tabela a seguir detalha o efeito do parâmetro `privilege` em duas concessões. Uma concessão tem o escopo `S3://DOC-EXAMPLE-BUCKET1/bob/*`, que inclui todo o prefixo `bob/` no bucket `DOC-EXAMPLE-BUCKET1`. A outra concessão tem o escopo `S3://DOC-EXAMPLE-BUCKET1/bob/reports/*`, que inclui apenas o prefixo `bob/reports/` no bucket `DOC-EXAMPLE-BUCKET1`.

Escopo da concessão	Escopo solicitado	Privilege	Escopo retornado	Efeito
S3://DOC-EXAMPLE-BUCKET1/bob/*	DOC-EXAMPLE-BUCKET1/bob/*	Default	DOC-EXAMPLE-BUCKET1/bob/*	O solicitante tem acesso a todos os objetos que têm nomes de chave que começam com o prefixo <i>bob/</i> no bucket <i>DOC-EXAMPLE-BUCKET1</i> .
S3://DOC-EXAMPLE-BUCKET1/bob/*	DOC-EXAMPLE-BUCKET1/bob/	Minimal	DOC-EXAMPLE-BUCKET1/bob/	Sem um caractere curinga * depois do nome do prefixo <i>bob/</i> , o solicitante tem acesso somente ao objeto chamado <i>bob/</i> no bucket <i>DOC-EXAMPLE-BUCKET1</i> . Não é comum ter esse objeto. O solicitante não tem acesso a nenhum outro objeto, incluindo aqueles que têm nomes de chave que começam com o prefixo <i>bob/</i> .
S3://DOC-EXAMPLE-BUCKET1/bob/*	DOC-EXAMPLE-BUCKET1/bob/images/*	Minimal	DOC-EXAMPLE-BUCKET1/bob/images/*	O solicitante tem acesso a todos os objetos que têm nomes de chave que começam com o prefixo <i>bob/images/*</i> no bucket <i>DOC-EXAMPLE-BUCKET1</i> .
S3://DOC-EXAMPLE-BUCKET1/bob/*	DOC-EXAMPLE-BUCKET1/bob/reports/*	Default	DOC-EXAMPLE-BUCKET1/bob/reports/*	O solicitante tem acesso a todos os objetos que têm nomes de chave

Escopo da concessão	Escopo solicitado	Privilege	Escopo retornado	Efeito
<i>BUCKET</i> 1 /bob/ repo rts/*	<i>BUCKET</i> 1 /bob/ repo rts/ file. txt			que começam com o prefixo bob/reports no bucket <i>DOC-EXAMPLE-BUCKET1</i> , que é o escopo da concessão correspondente.
S3:// <i>DOC-EXAMPLE-BUCKET1</i> /bob/ repo rts/*	<i>DOC-EXAMPLE-BUCKET1</i> /bob/ repo rts/ file. txt	Minimal	<i>DOC-EXAMPLE-BUCKET1</i> /bob/reports/ file.txt	O solicitante tem acesso somente ao objeto com o nome de chave bob/reports/file.txt no bucket <i>DOC-EXAMPLE-BUCKET1</i> . O solicitante não tem acesso a nenhum outro objeto.

O parâmetro `durationSeconds` define a duração da credencial temporária, em segundos. O valor padrão é 3600 segundos (1 hora), mas o solicitante (o beneficiário) pode especificar um intervalo que varia de 900 segundos (15 minutos) até 43200 segundos (12 horas). Se o beneficiário solicitar um valor maior do que esse máximo, a solicitação falhará.

Note

Na solicitação de token temporário, se o local for um objeto, defina o valor do parâmetro `targetType` na solicitação como `Object`. Esse parâmetro só será necessário se o local for um objeto e o nível de privilégio for `Minimal`. Se o local for um bucket ou um prefixo, você não precisará especificar esse parâmetro.

Para obter mais informações, consulte [GetDataAccess](#) na Referência da API do Amazon Simple Storage Service.

Você pode solicitar credenciais temporárias usando a AWS Command Line Interface (AWS CLI), a API REST do Amazon S3 e os AWS SDKs.

Usando a AWS CLI

Para instalar a AWS CLI, consulte [Instalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Para usar o comando a seguir, substitua os *user input placeholders* por suas próprias informações.

Exemplo Solicitar credenciais temporárias

Solicitação:

```
aws s3control get-data-access \  
--account-id 111122223333 \  
--target s3://DOC-EXAMPLE-BUCKET/prefixA* \  
--permission READ \  
--privilege Default \  
--region us-east-2
```

Resposta:

```
{  
  "Credentials": {  
    "AccessKeyId": "Example-key-id",  
    "SecretAccessKey": "Example-access-key",  
    "SessionToken": "Example-session-token",  
    "Expiration": "2023-06-14T18:56:45+00:00",  
    "MatchedGrantTarget": "s3://DOC-EXAMPLE-BUCKET/prefixA**"  
  }  
}
```

Uso da API REST

Para obter informações sobre o suporte à API REST do Amazon S3 para solicitar credenciais temporárias da funcionalidade Concessões de Acesso do S3, consulte [GetDataAccess](#) na Referência da API do Amazon Simple Storage Service.

Uso dos AWS SDKs

Esta seção fornece um exemplo de como os beneficiários solicitam credenciais temporárias da funcionalidade Concessões de Acesso do S3 usando os AWS SDKs.

Java

O exemplo de código a seguir retorna as credenciais temporárias que o beneficiário usa para acessar os dados do S3. Para usar esse exemplo de código, substitua os *user input placeholders* por suas próprias informações.

Example Obter credenciais temporárias

Solicitação:

```
public void getDataAccess() {
    GetDataAccessRequest getDataAccessRequest = GetDataAccessRequest.builder()
        .accountId("111122223333")
        .permission(Permission.READ)
        .privilege(Privilege.MINIMAL)
        .target("s3://DOC-EXAMPLE-BUCKET/prefixA*")
        .build();
    GetDataAccessResponse getDataAccessResponse =
        s3Control.getDataAccess(getDataAccessRequest);
    LOGGER.info("GetDataAccessResponse: " + getDataAccessResponse);
}
```

Resposta:

```
GetDataAccessResponse(
    Credentials=Credentials(
    AccessKeyId="Example-access-key-id",
    SecretAccessKey="Example-secret-access-key",
    SessionToken="Example-session-token",
    Expiration=2023-06-07T06:55:24Z
    ))
```

Acessar dados do S3 por meio de uma concessão de acesso

Depois que um beneficiário [obtem credenciais temporárias](#) por meio de uma concessão de acesso, ele pode usar essas credenciais temporárias para chamar as operações de API do Amazon S3 para acessar os dados.

Os beneficiários podem acessar dados do S3 usando a AWS Command Line Interface (AWS CLI), os AWS SDKs e a API REST do Amazon S3.

Usando a AWS CLI

Depois que o beneficiário obtém as credenciais temporárias da funcionalidade Concessões de Acesso do S3, ele pode configurar um perfil com essas credenciais para recuperar os dados.

Para instalar a AWS CLI, consulte [Instalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Para usar os comandos a seguir, substitua os *user input placeholders* por suas próprias informações.

Example : configurar um perfil

```
aws configure set aws_access_key_id "$accessKey" --profile access-grants-consumer-access-profile
aws configure set aws_secret_access_key "$secretKey" --profile access-grants-consumer-access-profile
aws configure set aws_session_token "$sessionToken" --profile access-grants-consumer-access-profile
```

Para usar o comando a seguir, substitua os *user input placeholders* por suas próprias informações.

Example : obter os dados do S3

O beneficiário pode usar o comando [get-object](#) da AWS CLI para acessar os dados. O beneficiário também pode usar [put-object](#), [ls](#) e outros comandos da AWS CLI do S3.

```
aws s3api get-object \
--bucket DOC-EXAMPLE-BUCKET1 \
--key myprefix \
--region us-east-2 \
--profile access-grants-consumer-access-profile
```

Uso dos AWS SDKs

Esta seção fornece exemplos de como os beneficiários podem acessar os dados do S3 usando os AWS SDKs.

Java

Para obter exemplos de como obter dados do S3 usando credenciais temporárias, consulte como [obter um objeto usando os AWS SDKs](#) e [exemplos de código do Amazon S3 para o AWS SDK for Java 2.x](#).

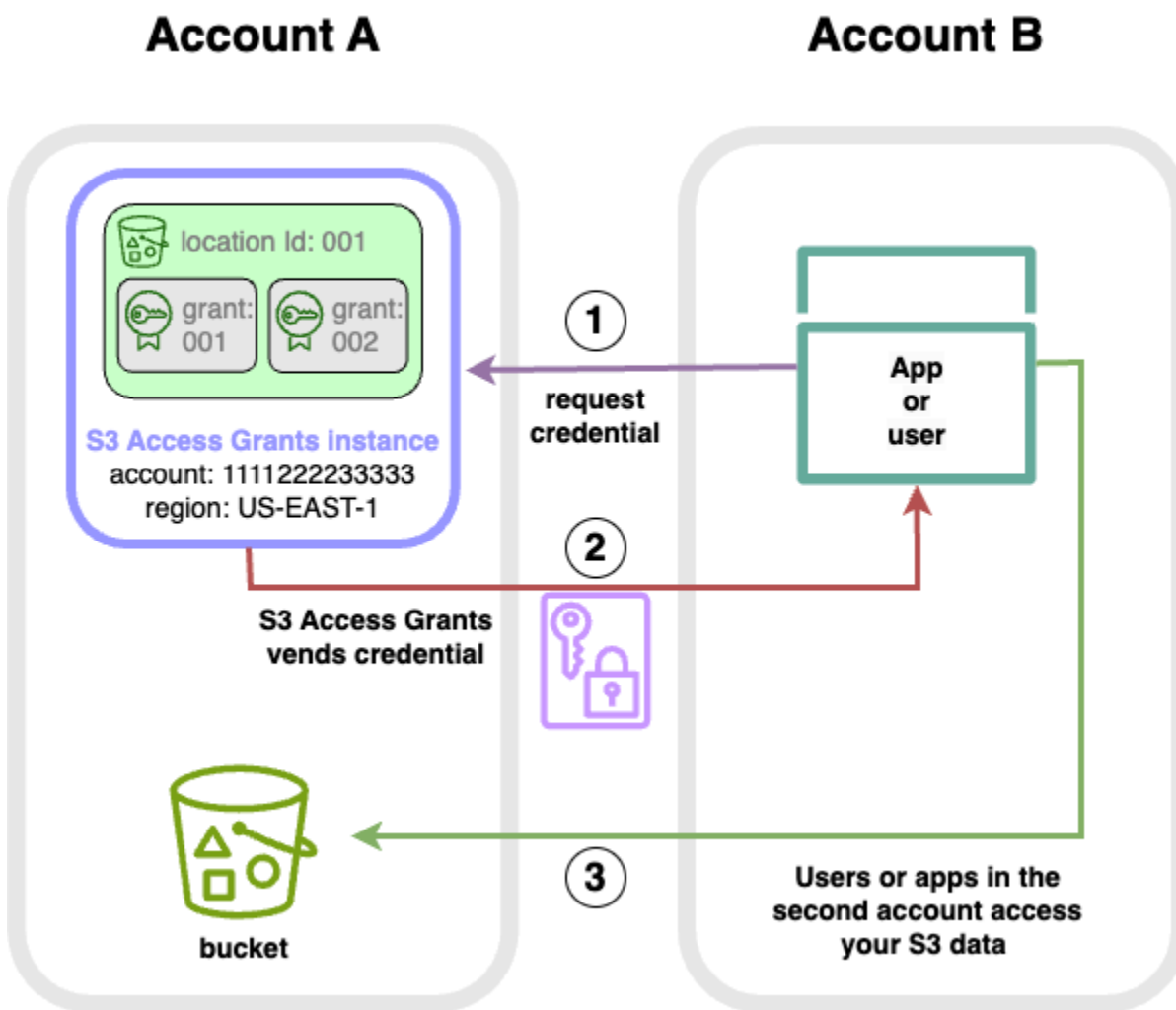
Acesso entre contas da funcionalidade Concessões de Acesso do S3

Com a funcionalidade Concessões de Acesso do S3, você pode conceder acesso aos dados do Amazon S3 ao seguinte:

- Identidades do AWS Identity and Access Management (IAM) em sua conta
- Identidades do IAM em outras contas da AWS
- Diretório de usuários ou grupos na instância do AWS IAM Identity Center

Primeiro, configure o acesso entre contas para a outra conta. Essa ação requer a concessão de acesso à instância da funcionalidade Concessões de Acesso do S3 por meio de uma política de recursos. Em seguida, conceda acesso aos seus dados do S3 (buckets, prefixos ou objetos) usando concessões.

Após a configuração do acesso entre contas, a outra conta pode solicitar credenciais de acesso temporário aos seus dados do Amazon S3 por meio da funcionalidade Concessões de Acesso do S3. A imagem a seguir mostra o fluxo de usuários para acesso entre contas ao S3 por meio da funcionalidade Concessões de Acesso do S3:



1. Os usuários ou as aplicações em uma segunda conta (B) solicitam credenciais da instância da funcionalidade Concessões de Acesso do S3 em sua conta (A), na qual os dados do Amazon S3 são armazenados. Para ter mais informações, consulte [Solicitar acesso aos dados do Amazon S3 por meio da funcionalidade Concessões de Acesso do S3](#).
2. A instância da funcionalidade Concessões de Acesso do S3 em sua conta (A) retornará credenciais temporárias se houver uma concessão que dê à segunda conta acesso aos seus dados do Amazon S3. Para ter mais informações, consulte [the section called “Criar concessões”](#).
3. Os usuários ou as aplicações na segunda conta (B) usam as credenciais fornecidas pela funcionalidade Concessões de Acesso do S3 para acessar os dados do S3 em sua conta (A).

Acesso entre contas da funcionalidade Concessões de Acesso do S3

Para conceder acesso entre contas ao S3 por meio da funcionalidade Concessões de Acesso do S3, siga estas etapas:

- Etapa 1: configure uma instância da funcionalidade Concessões de Acesso do S3 em sua conta (por exemplo, ID de conta 111122223333), na qual os dados do S3 são armazenados.
- Etapa 2: configure a política de recursos para a instância da funcionalidade Concessões de Acesso do S3 em sua conta 111122223333 para dar acesso à segunda conta (por exemplo, ID de conta 444455556666).
- Etapa 3: configure as permissões do IAM para a entidade principal do IAM na segunda conta (444455556666) para solicitar credenciais da instância da funcionalidade Concessões de Acesso do S3 em sua conta 111122223333.
- Etapa 4: crie uma concessão em sua conta 111122223333 que dê à entidade principal do IAM na segunda conta (444455556666) acesso a alguns dos dados do S3 em sua conta 111122223333.

Etapa 1: configurar uma instância da funcionalidade Concessões de Acesso do S3 em sua conta

Primeiro, você deve ter uma instância da funcionalidade Concessões de Acesso do S3 em sua conta 111122223333 para gerenciar o acesso aos dados do Amazon S3. É necessário criar uma instância da funcionalidade Concessões de Acesso do S3 em cada Região da AWS na qual os dados do S3 que você deseja compartilhar são armazenados. Se você compartilhar dados em mais de uma Região da AWS, repita cada uma dessas etapas de configuração para cada Região da AWS. Se você já tiver uma instância da funcionalidade Concessões de Acesso do S3 na Região da AWS local em que os dados do S3 são armazenados, passe para a próxima etapa. Se você não configurou uma instância da funcionalidade Concessões de Acesso do S3, consulte [Criar uma instância da funcionalidade Concessões de Acesso do S3](#) para concluir esta etapa.

Etapa 2: configurar a política de recursos da instância da funcionalidade Concessões de Acesso do S3 para conceder acesso entre contas

Depois de criar uma instância da funcionalidade Concessões de Acesso do S3 em sua conta 111122223333 para acesso entre contas, configure a política baseada em recursos para a instância da funcionalidade Concessões de Acesso do S3 em sua conta 111122223333 a fim de conceder acesso entre contas. A instância da funcionalidade Concessões de Acesso do S3 é compatível com políticas baseadas em recurso. Com a política baseada em recursos correta em vigor, você pode

conceder a usuários ou perfis do AWS Identity and Access Management (IAM) de outras Contas da AWS acesso à sua instância da funcionalidade Concessões de Acesso do S3. O acesso entre contas concede somente estas permissões (ações):

- `s3:GetAccessGrantsInstanceForPrefix`: o usuário, o perfil ou a aplicação pode recuperar a instância da funcionalidade Concessões de Acesso do S3 que contém um prefixo específico.
- `s3:ListAccessGrants`
- `s3:ListAccessLocations`
- `s3:GetDataAccess`: o usuário, o perfil ou a aplicação pode solicitar credenciais temporárias com base no acesso concedido por meio da funcionalidade Concessões de Acesso do S3. Use essas credenciais para acessar os dados do S3 aos quais você recebeu acesso.

Você pode escolher quais dessas permissões deseja incluir na política de recursos. Essa política de recursos na instância da funcionalidade Concessões de Acesso do S3 é uma política normal baseada em recursos e é compatível com tudo o que a [linguagem de política do IAM](#) permite. Na mesma política, você pode conceder acesso a identidades do IAM específicas na conta 111122223333 (por exemplo, usando a condição `aws:PrincipalArn`), mas não precisa fazer isso com a funcionalidade Concessões de Acesso do S3. Em vez disso, na instância da funcionalidade Concessões de Acesso do S3, é possível criar concessões para identidades do IAM individuais de sua conta, bem como para a outra conta. Ao gerenciar cada concessão de acesso por meio da funcionalidade Concessões de Acesso do S3, você pode escalar suas permissões.

Se você já usa o [AWS Resource Access Manager](#) (AWS RAM), é possível utilizá-lo para compartilhar recursos do `s3:AccessGrants` com outras contas ou dentro da sua organização. Para obter mais informações, consulte [Working with shared AWS resources](#). Se você não usa o AWS RAM, também pode adicionar a política de recursos usando as operações de API da funcionalidade Concessões de Acesso do S3 ou a AWS Command Line Interface (AWS CLI).

Uso do console do S3

Recomendamos que você use o console do AWS Resource Access Manager (AWS RAM) para compartilhar recursos do `s3:AccessGrants` com outras contas ou dentro da sua organização. Para compartilhar a funcionalidade Concessões de Acesso do S3 entre contas, faça o seguinte:

Para configurar a política de recursos da instância da funcionalidade Concessões 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. Selecione a Região da AWS no seletor de Região da AWS.
3. No painel de navegação esquerdo, escolha Concessões de acesso.
4. Na página da instância da funcionalidade Concessões de Acesso, na seção Instância nesta conta, selecione Compartilhar instância. Isso redirecionará você para o console do AWS RAM.
5. Selecione Criar compartilhamento de recursos.
6. Siga as etapas do AWS RAM para criar um compartilhamento de recursos. Para obter mais informações, consulte [Creating a resource share in AWS RAM](#).

Usando a AWS CLI

Para instalar a AWS CLI, consulte [Instalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Você pode adicionar a política de recursos usando o comando `put-access-grants-instance-resource-policy` da CLI.

Quanto ao acesso entre contas para a instância da funcionalidade Concessões de Acesso do S3 que está em sua conta 111122223333, se você quiser concedê-lo à segunda conta (444455556666), a política de recursos da instância da funcionalidade Concessões de Acesso do S3 em sua conta 111122223333 deve dar à segunda conta (444455556666) permissão para realizar as seguintes ações:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Na política de recursos da instância da funcionalidade Concessões de Acesso do S3, especifique o ARN da sua instância da funcionalidade Concessões de Acesso do S3 como o `Resource` e a segunda conta (444455556666) como a `Principal`. Para usar o exemplo a seguir, substitua os *espaços reservados para entrada do usuário* por suas próprias informações.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```

"Principal": {
  "AWS": "444455556666"
},
"Action": [
  "s3:ListAccessGrants",
  "s3:ListAccessGrantsLocations",
  "s3:GetDataAccess",
  "s3:GetAccessGrantsInstanceForPrefix"
],
"Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
} ]
}

```

Para adicionar ou atualizar a política de recursos da instância da funcionalidade Concessões de Acesso do S3, use o comando a seguir. Ao usar o comando de exemplo a seguir, substitua os *user input placeholders* por suas próprias informações.

Example Adicionar ou atualizar uma política de recursos da instância da funcionalidade Concessões de Acesso do S3

```

aws s3control put-access-grants-instance-resource-policy \
--account-id 111122223333 \
--policy file://resourcePolicy.json \
--region us-east-2
{
  "Policy": "{\n
    \"Version\": \"2012-10-17\",\n
    \"Statement\": [{\n
      \"Effect\": \"Allow\",\n
      \"Principal\": {\n
        \"AWS\": \"444455556666\"\n
      },\n
      \"Action\": [\n
        \"s3:ListAccessGrants\",\n
        \"s3:ListAccessGrantsLocations\",\n
        \"s3:GetDataAccess\",\n
        \"s3:GetAccessGrantsInstanceForPrefix\"\n
      ],\n
      \"Resource\": \"arn:aws:s3:us-east-2:111122223333:access-grants/default\"\n
    }]\n
  }"

```

```
"CreatedAt": "2023-06-16T00:07:47.473000+00:00"
}
```

Exemplo Obter uma política de recursos da funcionalidade Concessões de Acesso do S3

Você também pode usar a CLI para obter ou excluir uma política de recursos para uma instância da funcionalidade Concessões de Acesso do S3.

Para obter uma política de recursos da funcionalidade Concessões de Acesso do S3, use o comando de exemplo a seguir. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control get-access-grants-instance-resource-policy \
--account-id 111122223333 \
--region us-east-2

{
  "Policy": "{\n\"Version\":\n\"2012-10-17\", \"Statement\": [\n{\n\"Effect\":\n\"Allow\", \"Principal\":\n{\n\"AWS\":\n\"arn:aws:iam:\n:111122223333:root\", \"Action\": [\n\"s3:ListAccessGrants\", \n\"s3:ListAccessGrantsLocations\", \n\"s3:GetDataAccess\"], \"Resource\":\n\"arn:aws:s3:\n:us-east-2:\n111122223333:access-grants/default\"}]}\n], \n\"CreatedAt\": \"2023-06-16T00:07:47.473000+00:00\"
}
```

Exemplo Excluir uma política de recursos da funcionalidade Concessões de Acesso do S3

Para excluir uma política de recursos da funcionalidade Concessões de Acesso do S3, use o comando de exemplo a seguir. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control delete-access-grants-instance-resource-policy \
--account-id 111122223333 \
--region us-east-2

// No response body
```

Uso da API REST

Você pode adicionar a política de recursos usando a API [PutAccessGrantsInstanceResourcePolicy](#).

Quanto ao acesso entre contas para a instância da funcionalidade Concessões de Acesso do S3 que está em sua conta 111122223333, se você quiser concedê-lo à segunda conta (444455556666),

a política de recursos da instância da funcionalidade Concessões de Acesso do S3 em sua conta 111122223333 deve dar à segunda conta (444455556666) permissão para realizar as seguintes ações:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Na política de recursos da instância da funcionalidade Concessões de Acesso do S3, especifique o ARN da sua instância da funcionalidade Concessões de Acesso do S3 como o `Resource` e a segunda conta (444455556666) como a `Principal`. Para usar o exemplo a seguir, substitua os *espaços reservados para entrada do usuário* por suas próprias informações.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "444455556666"
      },
      "Action": [
        "s3:ListAccessGrants",
        "s3:ListAccessGrantsLocations",
        "s3:GetDataAccess",
        "s3:GetAccessGrantsInstanceForPrefix"
      ],
      "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
    }
  ]
}
```

Em seguida, você pode usar a API [PutAccessGrantsInstanceResourcePolicy](#) para configurar a política.

Para obter informações sobre o suporte à API REST para atualizar, obter ou excluir uma política de recursos para a instância da funcionalidade Concessões de Acesso do S3, consulte as seguintes seções na Referência da API do Amazon Simple Storage Service:

- [PutAccessGrantsInstanceResourcePolicy](#)

- [GetAccessGrantsInstanceResourcePolicy](#)
- [DeleteAccessGrantsInstanceResourcePolicy](#)

Uso dos AWS SDKs

Esta seção fornece exemplos de SDK da AWS sobre como configurar sua política de recursos da funcionalidade Concessões de Acesso do S3 para conceder a uma segunda conta da AWS acesso a alguns dos seus dados do S3.

Java

Adicione, atualize, obtenha ou exclua uma política de recursos para gerenciar o acesso entre contas a uma instância da funcionalidade Concessões de Acesso do S3.

Example Adicionar ou atualizar uma política de recursos da instância da funcionalidade Concessões de Acesso do S3

Quanto ao acesso entre contas para a instância da funcionalidade Concessões de Acesso do S3 que está em sua conta 111122223333, se você quiser concedê-lo à segunda conta (444455556666), a política de recursos da instância da funcionalidade Concessões de Acesso do S3 em sua conta 111122223333 deve dar à segunda conta (444455556666) permissão para realizar as seguintes ações:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Na política de recursos da instância da funcionalidade Concessões de Acesso do S3, especifique o ARN da sua instância da funcionalidade Concessões de Acesso do S3 como o `Resource` e a segunda conta (444455556666) como a `Principal`. Para usar o exemplo a seguir, substitua os *espaços reservados para entrada do usuário* por suas próprias informações.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```

"Principal": {
  "AWS": "444455556666"
},
"Action": [
  "s3:ListAccessGrants",
  "s3:ListAccessGrantsLocations",
  "s3:GetDataAccess",
  "s3:GetAccessGrantsInstanceForPrefix"
],
"Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
} ]
}

```

Para adicionar ou atualizar uma política de recursos da instância da funcionalidade Concessões de Acesso do S3, você pode usar o seguinte código de exemplo:

```

public void putAccessGrantsInstanceResourcePolicy() {
  PutAccessGrantsInstanceResourcePolicyRequest putRequest =
  PutAccessGrantsInstanceResourcePolicyRequest.builder()
  .accountId(111122223333)
  .policy(RESOURCE_POLICY)
  .build();
  PutAccessGrantsInstanceResourcePolicyResponse putResponse =
  s3Control.putAccessGrantsInstanceResourcePolicy(putRequest);
  LOGGER.info("PutAccessGrantsInstanceResourcePolicyResponse: " + putResponse);
}

```

Resposta:

```

PutAccessGrantsInstanceResourcePolicyResponse(
  Policy={
    "Version": "2012-10-17",
    "Statement": [{
      "Effect": "Allow",
      "Principal": {
        "AWS": "444455556666"
      },
      "Action": [
        "s3:ListAccessGrants",
        "s3:ListAccessGrantsLocations",
        "s3:GetDataAccess",
        "s3:GetAccessGrantsInstanceForPrefix"
      ],

```

```
"Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
}]
}
)
```

Example Obter uma política de recursos da funcionalidade Concessões de Acesso do S3

Para obter uma política de recursos da funcionalidade Concessões de Acesso do S3, você pode usar o código de exemplo a seguir. Para usar o comando a seguir, substitua os *user input placeholders* por suas próprias informações.

```
public void getAccessGrantsInstanceResourcePolicy() {
    GetAccessGrantsInstanceResourcePolicyRequest getRequest =
        GetAccessGrantsInstanceResourcePolicyRequest.builder()
            .accountId(111122223333)
            .build();
    GetAccessGrantsInstanceResourcePolicyResponse getResponse =
        s3Control.getAccessGrantsInstanceResourcePolicy(getRequest);
    LOGGER.info("GetAccessGrantsInstanceResourcePolicyResponse: " + getResponse);
}
```

Resposta:

```
GetAccessGrantsInstanceResourcePolicyResponse(
    Policy={"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":
{"AWS":"arn:aws:iam:444455556666:root"},"Action":
["s3:ListAccessGrants","s3:ListAccessGrantsLocations","s3:GetDataAccess"],"Resource":["arn:aws:
east-2:111122223333:access-grants/default"]}],
    CreatedAt=2023-06-15T22:54:44.319Z
)
```

Example Excluir uma política de recursos da funcionalidade Concessões de Acesso do S3

Para excluir uma política de recursos da funcionalidade Concessões de Acesso do S3, você pode usar o código de exemplo a seguir. Para usar o comando a seguir, substitua os *user input placeholders* por suas próprias informações.

```
public void deleteAccessGrantsInstanceResourcePolicy() {
    DeleteAccessGrantsInstanceResourcePolicyRequest deleteRequest =
        DeleteAccessGrantsInstanceResourcePolicyRequest.builder()
            .accountId(111122223333)
```

```
.build();
DeleteAccessGrantsInstanceResourcePolicyResponse deleteResponse =
s3Control.putAccessGrantsInstanceResourcePolicy(deleteRequest);
LOGGER.info("DeleteAccessGrantsInstanceResourcePolicyResponse: " + deleteResponse);
}
```

Resposta:

```
DeleteAccessGrantsInstanceResourcePolicyResponse()
```

Etapa 3: conceder às identidades do IAM em uma segunda conta permissão para chamar a instância da funcionalidade Concessões de Acesso do S3 em sua conta

Depois que o proprietário dos dados do Amazon S3 configurar a política entre contas para a instância da funcionalidade Concessões de Acesso do S3 na conta 111122223333, o proprietário da segunda conta (444455556666) deverá criar uma política baseada em identidade para seus usuários ou perfis do IAM e o proprietário deverá conceder a eles acesso à instância da funcionalidade Concessões de Acesso do S3. Na política baseada em identidade, inclua uma ou mais das seguintes ações, dependendo do que é concedido na política de recursos da instância da funcionalidade Concessões de Acesso do S3 e das permissões que você deseja conceder:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

De acordo com o [padrão de acesso entre contas da AWS](#), os usuários ou perfis do IAM na segunda conta (444455556666) devem ter explicitamente uma ou mais dessas permissões. Por exemplo, conceda a permissão `s3:GetDataAccess` para que o usuário ou perfil do IAM possa chamar a instância da funcionalidade Concessões de Acesso do S3 na conta 111122223333 para solicitar credenciais.

Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
{
  "Effect": "Allow",
  "Action": [
    "s3:GetDataAccess",
  ],
  "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
}
]
}
```

Para obter mais informações sobre como editar uma política do IAM baseada em identidade, consulte [Edição de políticas do IAM](#) no AWS Identity and Access Management Guia do usuário do IAM.

Etapa 4: criar uma concessão na instância da funcionalidade Concessões de Acesso do S3 de sua conta que dê à entidade principal do IAM na segunda conta acesso a alguns de seus dados do S3.

Na etapa final de configuração, você pode criar uma concessão na instância da funcionalidade Concessões de Acesso do S3 em sua conta 111122223333 que dê acesso à identidade do IAM na segunda conta (444455556666) a alguns dos dados do S3 em sua conta. Isso pode ser feito usando o console do Amazon S3, a CLI, a API e os SDKs. Para ter mais informações, consulte [Criar concessões](#).

Na concessão, especifique o ARN da AWS da identidade do IAM da segunda conta e especifique a qual local nos seus dados do S3 (um bucket, prefixo ou objeto) você está concedendo acesso. Esse local já deve estar registrado na instância da funcionalidade Concessões de Acesso do S3. Para ter mais informações, consulte [Registrar um local](#). Você tem a opção de especificar um subprefixo. Por exemplo, se o local ao qual você está concedendo acesso for um bucket e quiser limitar ainda mais o acesso a um objeto específico nesse bucket, passe o nome da chave do objeto no campo `S3SubPrefix`. Ou, se você quiser limitar o acesso aos objetos no bucket com nomes de chave que começam com um prefixo específico, como `2024-03-research-results/`, passe `S3SubPrefix=2024-03-research-results/`.

Veja a seguir um exemplo de comando da CLI para criar uma concessão de acesso para uma identidade na segunda conta. Consulte [Criar concessões](#) Para mais informações. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control create-access-grant \  
--account-id 111122223333 \  
--access-grants-location-id default \  
--access-grants-location-configuration S3SubPrefix=prefixA* \  
--permission READ \  
--grantee GranteeType=IAM,GranteeIdentifier=arn:aws:iam::444455556666:role/data-  
consumer-1
```

Depois de configurar o acesso entre contas, o usuário ou perfil na segunda conta pode fazer o seguinte:

- Chamar `ListAccessGrantsInstances` para listar as instâncias da funcionalidade Concessões de Acesso do S3 compartilhadas com ele por meio do AWS RAM. Para ter mais informações, consulte [Ver os detalhes de uma instância da funcionalidade Concessões de Acesso do S3](#).
- Solicitar credenciais temporárias da funcionalidade Concessões de Acesso do S3. Para obter informações sobre como fazer essas solicitações, consulte [Solicitar acesso aos dados do Amazon S3 por meio da funcionalidade Concessões de Acesso do S3](#).

Usar etiquetas da AWS com a funcionalidade Concessões de Acesso do S3

As etiquetas na funcionalidade Concessões de Acesso do Amazon S3 têm características semelhantes às [etiquetas de objeto](#) no Amazon S3. Cada tag é um par de chave/valor. Os recursos na funcionalidade Concessões de Acesso do S3 que você pode marcar são [instâncias](#), [locais](#) e [concessões](#) da funcionalidade Concessões de Acesso do S3.

Note

A marcação na funcionalidade Concessões de Acesso do S3 usa operações de API diferentes da marcação de objetos. A funcionalidade Concessões de Acesso do S3 usa as operações de API [TagResource](#), [UntagResource](#) e [ListTagsForResource](#), em que um recurso pode ser uma instância, um local registrado ou uma concessão de acesso da funcionalidade Concessões de Acesso do S3.

Semelhante às [etiquetas de objeto](#), as seguintes limitações se aplicam:

- Você pode adicionar etiquetas a novos recursos da funcionalidade Concessões de Acesso do S3 ao criá-los, ou pode adicionar etiquetas a recursos existentes.

- Você pode associar até 10 etiquetas a um recurso. Se várias etiquetas estiverem associadas ao mesmo recurso, elas deverão ter chaves de etiqueta exclusivas.
- Um chave de tag pode ter até 128 caracteres Unicode e os valores de tag podem ter até 256 caracteres Unicode. As etiquetas são representadas internamente em UTF-16. Em UTF-16, os caracteres utilizam posições de um ou de dois caracteres.
- As chaves e os valores diferenciam letras maiúsculas de minúsculas.

Para obter mais informações sobre restrições de etiquetas, consulte [Restrições de tags definidas pelo usuário](#) no Guia do usuário do AWS Billing.

Você pode marcar recursos na funcionalidade Concessões de Acesso do S3 usando a AWS Command Line Interface (AWS CLI), a API REST do Amazon S3 ou os AWS SDKs.

Usando a AWS CLI

Para instalar a AWS CLI, consulte [Instalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Você pode marcar um recurso da funcionalidade Concessões de Acesso do S3 durante a criação ou depois de criá-lo. Os exemplos a seguir mostram como marcar ou desmarcar uma instância da funcionalidade Concessões de Acesso do S3. Você pode realizar operações semelhantes para locais registrados e concessões de acesso.

Para usar os comandos a seguir, substitua os *user input placeholders* por suas próprias informações.

Example : criar uma instância da funcionalidade Concessões de Acesso do S3

```
aws s3control create-access-grants-instance \  
  --account-id 111122223333 \  
  --profile access-grants-profile \  
  --region us-east-2 \  
  --tags Key=tagKey1,Value=tagValue1
```

Resposta:

```
{  
  "CreatedAt": "2023-10-25T01:09:46.719000+00:00",  
  "AccessGrantsInstanceId": "default",
```

```
"AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
}
```

Example : marcar uma instância já criada da funcionalidade Concessões de Acesso do S3

```
aws s3control tag-resource \
--account-id 111122223333 \
--resource-arn "arn:aws:s3:us-east-2:111122223333:access-grants/default" \
--profile access-grants-profile \
--region us-east-2 \
--tags Key=tagKey2,Value=tagValue2
```

Example : listar as etiquetas da instância da funcionalidade Concessões de Acesso do S3

```
aws s3control list-tags-for-resource \
--account-id 111122223333 \
--resource-arn "arn:aws:s3:us-east-2:111122223333:access-grants/default" \
--profile access-grants-profile \
--region us-east-2
```

Resposta:

```
{
  "Tags": [
    {
      "Key": "tagKey1",
      "Value": "tagValue1"
    },
    {
      "Key": "tagKey2",
      "Value": "tagValue2"
    }
  ]
}
```

Example : desmarcar a instância da funcionalidade Concessões de Acesso do S3

```
aws s3control untag-resource \
--account-id 111122223333 \
--resource-arn "arn:aws:s3:us-east-2:111122223333:access-grants/default" \
```



```
--profile access-grants-profile \  
--region us-east-2 \  
--tag-keys "tagKey2"
```

Uso da API REST

Você pode usar a API do Amazon S3 para marcar, desmarcar ou listar as etiquetas de uma instância, um local registrado ou uma concessão de acesso da funcionalidade Concessões de Acesso do S3. Para obter informações sobre o suporte à API REST para gerenciar a funcionalidade Concessões de Acesso do S3, consulte as seguintes seções na Referência da API do Amazon Simple Storage Service:

- [TagResource](#)
- [UntagResource](#)
- [ListTagsForResource](#)

Limitações da funcionalidade Concessões de Acesso do S3

A funcionalidade [Concessões de Acesso do S3](#) tem as seguintes limitações:

Note

Se o seu caso de uso exceder essas limitações, [entre em contato com o AWS Support](#) para solicitar limites maiores.

Instância da funcionalidade Concessões de Acesso do S3

Você pode criar uma instância da funcionalidade Concessões de Acesso do S3 por Região da AWS por conta. Consulte [Criar uma instância da funcionalidade Concessões de Acesso do S3](#).

Local da funcionalidade Concessões de Acesso do S3

Você pode registrar 1.000 locais da funcionalidade Concessões de Acesso do S3 por instância da funcionalidade Concessões de Acesso do S3. Consulte [Register an S3 Access Grants location](#).

Concessão

Você pode criar 100.000 concessões por instância da funcionalidade Concessões de Acesso do S3. Consulte [Create a grant](#).

Integrações da funcionalidade Concessões de Acesso do S3

A funcionalidade Concessões de Acesso do S3 pode ser usada com os seguintes serviços e recursos da AWS. Esta página será atualizada à medida que novas integrações forem disponibilizadas.

AWS IAM Identity Center

[Trusted identity propagation across applications](#)

Amazon EMR

[Launch an Amazon EMR cluster with S3 Access Grants](#)

Amazon EMR no EKS

[Launch an Amazon EMR on EKS cluster with S3 Access Grants](#)

Aplicação do Amazon EMR Sem Servidor

[Launch an Amazon EMR Serverless application with S3 Access Grants](#)

Amazon Athena

[Uso de grupos de trabalho do Athena habilitados para o Centro de Identidade do IAM](#)

Gerenciar o acesso com ACLs

As listas de controle de acesso (ACLs) são uma das opções baseadas em recursos que podem ser usadas para gerenciar o acesso a 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.

A Propriedade de objetos do S3 é uma configuração no nível do bucket do Amazon S3 que você pode usar para controlar a propriedade de objetos carregados no bucket e para desabilitar ou habilitar as ACLs. Por padrão, a Propriedade de Objetos está definida com a configuração

Imposto pelo proprietário do bucket e todas as ACLs estão desabilitadas. Quando as ACLs são desabilitadas, o proprietário do bucket possui todos os objetos do bucket e gerencia o acesso a eles exclusivamente usando políticas de gerenciamento de acesso.

A maioria dos casos de uso modernos no Amazon S3 não exige mais o uso de ACLs. Recomendamos manter as ACLs desabilitadas, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com as ACLs desabilitadas, é possível usar políticas para controlar o acesso a todos os objetos no bucket, independentemente de quem carregou os objetos para o bucket. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

Important

Se o bucket usar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos do S3, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Quando a configuração Imposto pelo proprietário do bucket estiver habilitada, as solicitações para definir listas de controle de acesso (ACLs) ou atualizar ACLs falharão e retornarão 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 \(ACL\)](#)
- [Configurar ACLs](#)
- [Exemplos de política para ACLs](#)

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.

A Propriedade de objetos do S3 é uma configuração no nível do bucket do Amazon S3 que você pode usar para controlar a propriedade de objetos carregados no bucket e para desabilitar

ou habilitar as ACLs. Por padrão, a Propriedade de Objetos está definida com a configuração Imposto pelo proprietário do bucket e todas as ACLs estão desabilitadas. Quando as ACLs são desabilitadas, o proprietário do bucket possui todos os objetos do bucket e gerencia o acesso a eles exclusivamente usando políticas de gerenciamento de acesso.

A maioria dos casos de uso modernos no Amazon S3 não exige mais o uso de ACLs. Recomendamos manter as ACLs desabilitadas, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com as ACLs desabilitadas, é possível usar políticas para controlar o acesso a todos os objetos no bucket, independentemente de quem carregou os objetos para o bucket. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

Important

Se o bucket usar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos do S3, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Quando a configuração Imposto pelo proprietário do bucket estiver habilitada, as solicitações para definir listas de controle de acesso (ACLs) ou atualizar ACLs falharão e retornarão 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 encontrar seu ID de usuário canônico, consulte [Encontrar um ID de usuário canônico da Conta da AWS](#). 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?](#)
- [Quais permissões posso conceder?](#)
- [Valores `aclRequired` para solicitações comuns do Amazon S3](#)
- [Amostra de ACL](#)
- [ACL pré-configurada](#)

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 `type="value"`, em que `type` 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 compatíveis com o Amazon S3, consulte [Regiões e endpoints](#) na 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@example.com", emailAddress="abc@example.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. Esse ID é uma longa string de caracteres, como:

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Para obter informações sobre como encontrar o ID de usuário canônico de sua conta, consulte [Find the canonical user ID for your Conta da AWS](#) no Guia de referência de gerenciamento de contas da AWS.

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 poderá carregar objetos para o bucket. Esses usuários anônimos não têm uma Conta da AWS. Quando um usuário anônimo carrega um objeto no bucket, o Amazon S3 adiciona um ID de usuário canônico especial (65a011a29cdf8ec533ec3d1c0aae921c) como o proprietário do objeto na ACL. Para ter mais informações, consulte [Propriedade de bucket e objeto do Amazon S3](#).

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 URIs do Amazon S3 em vez do ID de usuário canônico. O Amazon S3 fornece 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 Usuários autenticados, 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?](#).

- 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](#)) 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 usuários da AWS e permissões no IAM, consulte [Usar o AWS Identity and Access Management](#).

Quais permissões posso conceder?

A tabela a seguir lista o conjunto de permissões para as quais o Amazon S3 oferece suporte em uma ACL. O conjunto de permissões da ACL é o mesmo para a ACL de objetos e para a ACL de bucket. No entanto, dependendo do contexto (ACL de buckets ou ACL de objetos), essas permissões da ACL concedem permissão para operações de bucket ou objeto específicas. A tabela lista as permissões e descreve seus significados no contexto de objetos e buckets.

Para obter mais informações sobre permissões de ACL no console do Amazon S3, consulte [Configurar ACLs](#).

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	Permite ao beneficiário as permissões READ, WRITE, READ_ACP e WRITE_ACP no bucket	Permite ao beneficiário 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 toda a seção [Visão geral da lista de controle de acesso \(ACL\)](#) antes de conceder 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 de políticas para o Amazon S3](#)). 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 [Gerenciamento de identidade e acesso para o Amazon S3](#).

Para obter mais informações sobre permissões de ACL no console do Amazon S3, consulte [Configurar ACLs](#).

Permissão da ACL	Permissões correspondentes da política de acesso quando permissões da ACL são concedidas em um bucket	Permissões correspondentes da política de acesso quando permissões da ACL são concedidas em um objeto
READ	<code>s3:ListBucket</code> , <code>s3:ListBucketVersions</code> , e <code>s3:ListBucketMultipartUploads</code>	<code>s3:GetObject</code> e <code>s3:GetObjectVersion</code>
WRITE	<code>s3:PutObject</code> O proprietário do bucket pode criar, substituir e excluir qualquer objeto no bucket, e o proprietário do objeto tem <code>FULL_CONTROL</code> sobre seu objeto. Além disso, quando o favorecido é o proprietário do bucket, conceder	Não aplicável

Permissão da ACL	Permissões correspondentes da política de acesso quando permissões da ACL são concedidas em um bucket	Permissões correspondentes da política de acesso quando permissões da ACL são concedidas em um objeto
	a permissão WRITE em uma ACL do bucket permite que a ação <code>s3:DeleteObjectVersion</code> seja executada em qualquer versão naquele bucket.	
READ_ACP	<code>s3:GetBucketAc1</code>	<code>s3:GetObjectAc1</code> e <code>s3:GetObjectVersionAc1</code>
WRITE_ACP	<code>s3:PutBucketAc1</code>	<code>s3:PutObjectAc1</code> e <code>s3:PutObjectVersionAc1</code>
FULL_CONTROL	Equivalente a conceder as permissões READ, WRITE, READ_ACP e WRITE_ACP da ACL. Assim, essa permissão da ACL equivale à combinação das permissões correspondentes da política de acesso.	Equivalente a conceder as permissões READ, READ_ACP e WRITE_ACP da ACL. Assim, essa permissão da ACL equivale à combinação das permissões correspondentes da política de acesso.

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 a seguir 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](#).

Para obter exemplos de políticas que envolvem cabeçalhos específicos de ACL, consulte [Conceder a permissão `s3:PutObject` com uma condição que exige que o proprietário do bucket obtenha controle total](#). Consulte uma lista completa de chaves de condição específicas do Amazon S3 em [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço.

Valores `aclRequired` para solicitações comuns do Amazon S3

Para identificar solicitações do Amazon S3 que exigiam ACLs para autorização, você pode usar o valor `aclRequired` nos logs de acesso do servidor do Amazon S3 ou do AWS CloudTrail. O valor `aclRequired` que aparece nos logs de acesso ao servidor do CloudTrail ou do Amazon S3 depende de quais operações foram chamadas e de determinadas informações sobre o solicitante, o proprietário do objeto e o proprietário do bucket. Se nenhuma ACLs for necessária, ou se você estiver configurando a ACL `bucket-owner-full-control` predefinida, ou se as solicitações forem permitidas pela política de bucket, a string de valor `aclRequired` será “-” nos logs de acesso ao servidor do Amazon S3 e estará ausente no CloudTrail.

As tabelas a seguir listam os valores esperados de `aclRequired` nos logs de acesso ao servidor do CloudTrail ou do Amazon S3 para as várias operações de API do Amazon S3. É possível usar essas informações para entender quais operações do Amazon S3 dependem de ACLs para autorização. Nas tabelas a seguir, A, B e C representam as diferentes contas associadas ao solicitante, ao proprietário do objeto e ao proprietário do bucket. As entradas com um asterisco (*) indicam qualquer uma das contas A, B ou C.

Note

As operações `PutObject` na tabela a seguir, a menos que especificado de outra forma, indicam solicitações que não definem uma ACL, a não ser que a ACL seja `bucket-owner-full-control`. Um valor nulo para `aclRequired` indica que `aclRequired` está ausente nos logs do AWS CloudTrail.

Os valores de **aclRequired** para o CloudTrail


Nome de operação	Solicitante	Proprietário do objeto	Bucket owner	A política de bucket concede acesso	Valor do aclRequired	Motivo
GetObject	A	A	A	Yes ou No	nulo	Acesso na mesma conta
	A	B	A	Yes ou No	nulo	Acesso à mesma conta com o proprietário do bucket aplicado
	A	A	B	Sim	nulo	Acesso entre contas concedido pela política de bucket
	A	A	B	Não	Sim	O acesso entre contas depende da ACL
	A	A	B	Sim	nulo	Acesso entre contas concedido pela

Nome de operação	Solicitante	Proprietário do objeto	Bucket owner	A política de bucket concede acesso	Valor do aclRequired	Motivo
						política de bucket
	A	B	B	Não	Sim	O acesso entre contas depende da ACL
	A	B	C	Sim	nulo	Acesso entre contas concedido pela política de bucket
	A	B	C	Não	Sim	O acesso entre contas depende da ACL
PutObject	A	Não aplicável	A	Yes ou No	nulo	Acesso na mesma conta

Nome de operação	Solicitante	Proprietário do objeto	Bucket owner	A política de bucket concede acesso	Valor do aclRequired	Motivo
	A	Não aplicável	B	Sim	nulo	Acesso entre contas concedido pela política de bucket
	A	Não aplicável	B	Não	Sim	O acesso entre contas depende da ACL
PutObject com uma ACL (exceto por bucket-owner-full-control)	*	Não aplicável	*	Yes ou No	Sim	ACL de concessões de solicitação
ListObjects	A	Não aplicável	A	Yes ou No	nulo	Acesso na mesma conta

Nome de operação	Solicitante	Proprietário do objeto	Bucket owner	A política de bucket concede acesso	Valor do aclRequired	Motivo
	A	Não aplicável	B	Sim	nulo	Acesso entre contas concedido pela política de bucket
	A	Não aplicável	B	Não	Sim	O acesso entre contas depende da ACL
DeleteObject	A	Não aplicável	A	Yes ou No	nulo	Acesso na mesma conta
	A	Não aplicável	B	Sim	nulo	Acesso entre contas concedido pela política de bucket
	A	Não aplicável	B	Não	Sim	O acesso entre contas depende da ACL

Nome de operação	Solicitante	Proprietário do objeto	Bucket owner	A política de bucket concede acesso	Valor do aclRequired	Motivo
PutObject Acl	*	*	*	Yes ou No	Sim	ACL de concessões de solicitação
PutBucket Acl	*	Não aplicável	*	Yes ou No	Sim	ACL de concessões de solicitação

 Note

As operações REST.PUT.OBJECT na tabela a seguir, a menos que especificado de outra forma, indicam solicitações que não definem uma ACL, a não ser que a ACL seja bucket-owner-full-control. Uma string de valor aclRequired de "-" indica um valor nulo nos logs de acesso ao servidor do Amazon S3.

Valores de **aclRequired** para logs de acesso ao servidor do Amazon S3

Nome de operação	Solicitante	Proprietário do objeto	Bucket owner	A política de bucket concede acesso	Valor do aclRequired	Motivo
REST.GET.OBJECT	A	A	A	Yes ou No	-	Acesso na mesma conta
	A	B	A	Yes ou No	-	Acesso à mesma conta com

Nome de operação	Solicitante	Proprietário do objeto	Bucket owner	A política de bucket concede acesso	Valor do aclRequired	Motivo
						o proprietário do bucket aplicado
	A	A	B	Sim	-	Acesso entre contas concedido pela política de bucket
	A	A	B	Não	Sim	O acesso entre contas depende da ACL
	A	B	B	Sim	-	Acesso entre contas concedido pela política de bucket
	A	B	B	Não	Sim	O acesso entre contas depende da ACL

Nome de operação	Solicitante	Proprietário do objeto	Bucket owner	A política de bucket concede acesso	Valor do aclRequired	Motivo
	A	B	C	Sim	-	Acesso entre contas concedido pela política de bucket
	A	B	C	Não	Sim	O acesso entre contas depende da ACL
REST.PUT.OBJECT	A	Não aplicável	A	Yes ou No	-	Acesso na mesma conta
	A	Não aplicável	B	Sim	-	Acesso entre contas concedido pela política de bucket
	A	Não aplicável	B	Não	Sim	O acesso entre contas depende da ACL

Nome de operação	Solicitante	Proprietário do objeto	Bucket owner	A política de bucket concede acesso	Valor do aclRequired	Motivo
REST.PUT.OBJECT (exceto por bucket-owner-full-control)	*	Não aplicável	*	Yes ou No	Sim	ACL de concessões de solicitação
REST.GET.BUCKET	A	Não aplicável	A	Yes ou No	-	Acesso na mesma conta
	A	Não aplicável	B	Sim	-	Acesso entre contas concedido pela política de bucket
	A	Não aplicável	B	Não	Sim	O acesso entre contas depende da ACL
REST.DELETE.OBJECT	A	Não aplicável	A	Yes ou No	-	Acesso na mesma conta

Nome de operação	Solicitante	Proprietário do objeto	Bucket owner	A política de bucket concede acesso	Valor do aclRequired	Motivo
	A	Não aplicável	B	Sim	-	Acesso entre contas concedido pela política de bucket
	A	Não aplicável	B	Não	Sim	O acesso entre contas depende da ACL
REST.PUT.ACL	*	*	*	Yes ou No	Sim	ACL de concessões de solicitação

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>
<AccessControllList>
  <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>

</AccessControllList>
```

```
</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?) 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 <code>GET</code> 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 obtém <code>FULL_CONTROL</code> sobre o objeto. Se você especific

ACL pré-configurada	Aplica-se a	Permissões adicionadas à ACL
		<p>ar essa ACL pré-configurada ao criar um bucket, o Amazon S3 a ignorará.</p>
log-delivery-write	Bucket	<p>O grupo LogDelivery obtém as permissões WRITE e READ_ACP no bucket. Para obter mais informações sobre logs, consulte (Registrar em log as solicitações com registro em log de acesso ao servidor).</p>

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.

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 à ACL 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.

A Propriedade de objetos do S3 é uma configuração no nível do bucket do Amazon S3 que você pode usar para controlar a propriedade de objetos carregados no bucket e para desabilitar ou habilitar as ACLs. Por padrão, a Propriedade de Objetos está definida com a configuração Imposto pelo proprietário do bucket e todas as ACLs estão desabilitadas. Quando as ACLs são desabilitadas, o proprietário do bucket possui todos os objetos do bucket e gerencia o acesso a eles exclusivamente usando políticas de gerenciamento de acesso.

A maioria dos casos de uso modernos no Amazon S3 não exige mais o uso de ACLs. Recomendamos manter as ACLs desabilitadas, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com as ACLs desabilitadas, é possível usar políticas para controlar o acesso a todos os objetos no bucket, independentemente de quem carregou os objetos para o bucket. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

Important

Se o bucket usar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos do S3, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Quando a configuração Imposto pelo proprietário do bucket estiver habilitada, as solicitações para definir listas de controle de acesso (ACLs) ou atualizar ACLs falharão e retornarão 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](#).

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 \(ACL\)](#).

 Important

Se o bucket usar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos do S3, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Quando a configuração Imposto pelo proprietário do bucket estiver habilitada,

as solicitações para definir listas de controle de acesso (ACLs) ou atualizar ACLs falharão e retornarão 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 Proprietário do bucket (sua conta da 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 raiz da conta da AWS, e não a um usuário do AWS Identity and Access Management (IAM). Para obter mais informações sobre o usuário raiz, consulte [O Usuário raiz 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 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](#).

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 Acesso a outras Contas da AWS, selecione Remove.
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.

Permissão do console	Permissão da ACL	Acesso
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 \(ACL\)](#).

 Important

Se o bucket usar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos do S3, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Quando a configuração Imposto pelo proprietário do bucket estiver habilitada, as solicitações para definir listas de controle de acesso (ACLs) ou atualizar ACLs falharão e retornarão 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 raiz da conta da AWS, e não a um usuário do AWS Identity and Access Management (IAM). Para obter mais informações sobre o usuário raiz, consulte [O Usuário raiz da conta da AWS](#) no Guia do usuário do IAM.

Para alterar as permissões de acesso ao objeto do proprietário, em Acesso para o proprietário do objeto, selecione Sua 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 Acesso a outras Contas da AWS, selecione 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 Conta da AWS](#) no 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 dos 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 Imposto pelo proprietário do bucket para a Propriedade de Objetos do S3, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Quando a configuração Imposto pelo proprietário do bucket estiver habilitada, as solicitações para definir listas de controle de acesso (ACLs) ou atualizar ACLs falharão e retornarão 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](#)), 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);
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?](#)) 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](#).

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

namespace Amazon.DocSamples.S3
```

```
{
    class ManagingBucketACLTest
    {
        private const string newBucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            CreateBucketUseCannedACLAsync().Wait();
        }

        private static async Task CreateBucketUseCannedACLAsync()
        {
            try
            {
                // Add bucket (specify canned ACL).
                PutBucketRequest putBucketRequest = new PutBucketRequest()
                {
                    BucketName = newBucketName,
                    BucketRegion = S3Region.EUW1, // S3Region.US,
                                                // Add canned ACL.
                    CannedACL = S3CannedACL.LogDeliveryWrite
                };
                PutBucketResponse putBucketResponse = await
client.PutBucketAsync(putBucketRequest);

                // Retrieve bucket ACL.
                GetACLResponse getACLResponse = await client.GetACLAsync(new
GetACLRequest
                {
                    BucketName = newBucketName
                });
            }
            catch (AmazonS3Exception amazonS3Exception)
            {
                Console.WriteLine("S3 error occurred. Exception: " +
amazonS3Exception.ToString());
            }
            catch (Exception e)
            {

```

```
        Console.WriteLine("Exception: " + e.ToString());
    }
}
}
```

Exemplo 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 PutAc1.

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

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class ManagingObjectACLTest
    {
        private const string bucketName = "**** bucket name ****";
        private const string keyName = "**** object key name ****";
        private const string emailAddress = "**** email address ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
```

```
        TestObjectACLTestAsync().Wait();
    }
    private static async Task TestObjectACLTestAsync()
    {
        try
        {
            // Retrieve the ACL for the object.
            GetACLResponse aclResponse = await client.GetACLAsync(new
GetACLRequest
            {
                BucketName = bucketName,
                Key = keyName
            });

            S3AccessControlList acl = aclResponse.AccessControlList;

            // Retrieve the owner (we use this to re-add permissions after
we clear the ACL).
            Owner owner = acl.Owner;

            // Clear existing grants.
            acl.Grants.Clear();

            // Add a grant to reset the owner's full permission (the
previous clear statement removed all permissions).
            S3Grant fullControlGrant = new S3Grant
            {
                Grantee = new S3Grantee { CanonicalUser = owner.Id },
                Permission = S3Permission.FULL_CONTROL
            };

            // Describe the grant for the permission using an email address.
            S3Grant grantUsingEmail = new S3Grant
            {
                Grantee = new S3Grantee { EmailAddress = emailAddress },
                Permission = S3Permission.WRITE_ACP
            };
            acl.Grants.AddRange(new List<S3Grant> { fullControlGrant,
grantUsingEmail });

            // Set a new ACL.
            PutACLResponse response = await client.PutACLAsync(new
PutACLRequest
```

```
        {
            BucketName = bucketName,
            Key = keyName,
            AccessControlList = acl
        });
    }
    catch (AmazonS3Exception amazonS3Exception)
    {
        Console.WriteLine("An AmazonS3Exception was thrown. Exception: " +
amazonS3Exception.ToString());
    }
    catch (Exception e)
    {
        Console.WriteLine("Exception: " + e.ToString());
    }
}
}
```

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 Imposto pelo proprietário do bucket para a Propriedade de Objetos do S3, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Quando a configuração Imposto pelo proprietário do bucket estiver habilitada, as solicitações para definir listas de controle de acesso (ACLs) ou atualizar ACLs falharão e retornarão 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 \(ACL\)](#).

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](#).
- 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 \(ACL\)](#). 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

Usando a 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 Imposto pelo proprietário do bucket para a Propriedade de Objetos do S3, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Quando a configuração Imposto pelo proprietário do bucket estiver habilitada, as solicitações para definir listas de controle de acesso (ACLs) ou atualizar ACLs falharão e retornarão o código de erro `AccessControlListNotSupported`. Ainda há suporte para solicitações de leitura de ACLs.

Exemplos de política para ACLs

É possível usar chaves de condição em políticas de bucket para controlar o acesso ao Amazon S3.

Tópicos

- [Conceder a permissão `s3:PutObject` com uma condição que exige que o proprietário do bucket obtenha controle total](#)
- [Conceder a permissão `s3:PutObject` com uma condição no cabeçalho `x-amz-acl`](#)

Conceder a 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](#).

```
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 AccountBAdmin
```

Conceder a permissão s3:PutObject com uma condição no cabeçalho x-amz-acl

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 perfil ou usuário do IAM e incluir uma condição

do Amazon S3 semanticamente inválida, nenhum erro será relatado porque o IAM não pode validar condições do Amazon S3.

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](#). 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.

Como o ponto de acesso do Object Lambda é compatível com o CORS

Ao receber uma solicitação de um navegador ou quando a solicitação inclui um cabeçalho `Origin`, o S3 Object Lambda sempre adiciona um campo de cabeçalho `"AllowedOrigins": "*"` .

Para obter mais informações sobre como usar o CORS, consulte os tópicos a seguir.

Tópicos

- [Configuração de CORS](#)
- [Configurar o compartilhamento de recursos de origem cruzada \(CORS\)](#)

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 \(CORS\)](#).

 Important

No novo console do S3, a configuração CORS deve ser JSON.

Tópicos

- [Exemplo 1](#)
- [Exemplo 2](#)
- [Elemento AllowedMethod](#)
- [Elemento AllowedOrigin](#)
- [Elemento AllowedHeader](#)
- [Elemento ExposeHeader](#)
- [Elemento MaxAgeSeconds](#)

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 [Tutorial: Configurar um site estático usando um domínio personalizado registrado no Route 53](#).

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": []
  }
]
```

```

    }
  ]

```

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-*` 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](#).

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](#).

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](#).

6. Selecione Save changes.

ℹ 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, consulte [Nomes do recurso da Amazon \(ARNs\) e namespaces do serviço da AWS](#) noReferência geral da Amazon Web Services.

Uso dos 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 \(CORS\)](#).

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](#).

```
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](#).

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

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

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            CORSConfigTestAsync().Wait();
        }
        private static async Task CORSConfigTestAsync()
        {
            try
            {
                // Create a new configuration request and add two rules
                CORSConfiguration configuration = new CORSConfiguration
                {
                    Rules = new System.Collections.Generic.List<CORSRule>
```

```

        {
            new CORSRule
            {
                Id = "CORSRule1",
                AllowedMethods = new List<string> {"PUT", "POST",
"DELETE"},
                AllowedOrigins = new List<string> {"http://
*.example.com"}
            },
            new CORSRule
            {
                Id = "CORSRule2",
                AllowedMethods = new List<string> {"GET"},
                AllowedOrigins = new List<string> {"*"},
                MaxAgeSeconds = 3000,
                ExposeHeaders = new List<string> {"x-amz-server-side-
encryption"}
            }
        }
    };

    // Add the configuration to the bucket.
    await PutCORSConfigurationAsync(configuration);

    // Retrieve an existing configuration.
    configuration = await RetrieveCORSConfigurationAsync();

    // Add a new rule.
    configuration.Rules.Add(new CORSRule
    {
        Id = "CORSRule3",
        AllowedMethods = new List<string> { "HEAD" },
        AllowedOrigins = new List<string> { "http://www.example.com" }
    });

    // Add the configuration to the bucket.
    await PutCORSConfigurationAsync(configuration);

    // Verify that there are now three rules.
    configuration = await RetrieveCORSConfigurationAsync();
    Console.WriteLine();
    Console.WriteLine("Expected # of rulest=3; found:{0}",
configuration.Rules.Count);
    Console.WriteLine();

```

```
        Console.WriteLine("Pause before configuration delete. To continue,
click Enter...");
        Console.ReadKey();

        // Delete the configuration.
        await DeleteCORSConfigurationAsync();

        // Retrieve a nonexistent configuration.
        configuration = await RetrieveCORSConfigurationAsync();
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
}

static async Task PutCORSConfigurationAsync(CORSConfiguration configuration)
{
    PutCORSConfigurationRequest request = new PutCORSConfigurationRequest
    {
        BucketName = bucketName,
        Configuration = configuration
    };

    var response = await s3Client.PutCORSConfigurationAsync(request);
}

static async Task<CORSConfiguration> RetrieveCORSConfigurationAsync()
{
    GetCORSConfigurationRequest request = new GetCORSConfigurationRequest
    {
        BucketName = bucketName
    };

    var response = await s3Client.GetCORSConfigurationAsync(request);
    var configuration = response.Configuration;
    PrintCORSRules(configuration);
}
```

```
        return configuration;
    }

    static async Task DeleteCORSConfigurationAsync()
    {
        DeleteCORSConfigurationRequest request = new
DeleteCORSConfigurationRequest
        {
            BucketName = bucketName
        };
        await s3Client.DeleteCORSConfigurationAsync(request);
    }

    static void PrintCORSRules(CORSConfiguration configuration)
    {
        Console.WriteLine();

        if (configuration == null)
        {
            Console.WriteLine("\nConfiguration is null");
            return;
        }

        Console.WriteLine("Configuration has {0} rules:",
configuration.Rules.Count);
        foreach (CORSRule rule in configuration.Rules)
        {
            Console.WriteLine("Rule ID: {0}", rule.Id);
            Console.WriteLine("MaxAgeSeconds: {0}", rule.MaxAgeSeconds);
            Console.WriteLine("AllowedMethod: {0}", string.Join(", ",
rule.AllowedMethods.ToArray()));
            Console.WriteLine("AllowedOrigins: {0}", string.Join(", ",
rule.AllowedOrigins.ToArray()));
            Console.WriteLine("AllowedHeaders: {0}", string.Join(", ",
rule.AllowedHeaders.ToArray()));
            Console.WriteLine("ExposeHeader: {0}", string.Join(", ",
rule.ExposeHeaders.ToArray()));
        }
    }
}
}
```

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](#)

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](#).

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 proíba 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.

Important

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 nível determinado de acesso público aos buckets ou objetos, por exemplo, para hospedar um site estático como descrito em [Hospedagem de um site estático usando o Amazon S3](#), é possível personalizar as configurações individualmente para atender aos seus casos de uso de armazenamento. Habilitar o Bloqueio de Acesso Público ajuda a proteger seus recursos, impedindo que o acesso público seja concedido por meio de políticas de recursos ou listas de controle de acesso (ACLs) diretamente vinculadas aos recursos do S3. Além de habilitar o Bloqueio de Acesso Público, inspecione cuidadosamente as seguintes políticas para garantir que elas não concedam acesso público:

- Políticas baseadas em identidade vinculadas a entidades principais da AWS associadas (por exemplo, perfis do IAM)
- Políticas baseadas em recursos anexadas a recursos da AWS associados [por exemplo, chaves do AWS Key Management Service (KMS)]

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](#)
- [Executar operações de bloqueio de acesso público em um ponto de acesso](#)
- [O significado de "público"](#)
- [Usar o IAM Access Analyzer para S3 para revisar buckets públicos](#)
- [Permissões](#)
- [Configurar o bloqueio de acesso público](#)
- [Configurar o bloqueio de acesso público para sua conta](#)
- [Configurar o bloqueio de acesso público para seus buckets do S3](#)


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
BlockPublicAcls	A definição dessa opção como TRUE causa o seguinte comportamento:

Nome	Descrição
	<ul style="list-style-type: none">• As chamadas PUT Bucket acl e PUT Object falharão se a Access Control List (ACL – Lista de controle de acesso) especificada for pública.• As chamadas PUT Object falharão se a solicitação incluir uma ACL pública.• Se essa configuração for aplicada a uma conta, as chamadas PUT Bucket falharão se a solicitação incluir uma ACL pública. <p>Quando essa configuração for definida como TRUE, 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> <div data-bbox="428 989 1507 1449" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"><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></div>

Nome	Descrição
IgnorePublicAcls	<p>A definição dessa opção como TRUE faz o Amazon S3 ignorar todas as ACLs públicas em um bucket e todos os objetos contidos. Essa configuração permite bloquear com segurança acesso público concedido por ACLs ao mesmo tempo em que permite chamadas PUT Object que incluam uma ACL pública (ao contrário de BlockPublicAcls , que rejeita chamadas PUT Object que incluam uma ACL pública). A habilitação dessa configuração não afeta a persistência de ACLs existentes nem evita a definição de novas ACLs públicas.</p> <div data-bbox="428 638 1507 1096"><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></div>

Nome	Descrição
BlockPublicPolicy	<p>A definição dessa opção como TRUE para um bucket faz com que o Amazon S3 rejeite chamadas para a política PUT Bucket caso a política do bucket especificado permita acesso público. A definição dessa opção como TRUE para um bucket também faz com que o Amazon S3 rejeite chamadas para a política de acesso PUT para todos os pontos de acesso na mesma conta do bucket caso a política especificada permita acesso público.</p> <p>A definição dessa 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 Bucket que são feitas por meio do ponto de acesso, caso a política especificada (para o ponto de acesso ou para o bucket subjacente) permita acesso público.</p> <p>Use essa configuração para permitir 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> <div data-bbox="430 1035 1507 1633" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px;"><p> Important</p><p>Para usar essa configuração de maneira efetiva, recomendamos que aplique-a na 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></div>

Nome	Descrição
RestrictPublicBuckets	<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 entidades principais de serviço da AWS e a usuários autorizados dentro da conta do proprietário do bucket e da conta do proprietário do ponto de acesso. 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"

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](#).

Políticas de buckets


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](#)) para um ou mais dos seguintes itens:

- Um principal, um usuário, uma função ou um principal do produto da AWS (por exemplo, `aws:PrincipalOrgID`)
- 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.

Note

As políticas de bucket que concedem acesso condicionado à chave de condição `aws:SourceIp` com intervalos de IPs muito amplos (por exemplo, `0.0.0.0/1`) são avaliadas como “públicas”. Isso inclui valores maiores que `/8` para IPv4 e `/32` para IPv6 (excluindo intervalos privados RFC1918). Bloquear o acesso público rejeitará essas políticas “públicas” e impedirá o acesso entre contas a buckets que já usam essas políticas “públicas”.

- `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 ter mais informações, consulte [Pontos de acesso](#).

- `s3:DataAccessPointAccount`

Para obter mais informações sobre políticas de bucket, consulte [Políticas de bucket para o Amazon S3](#).

Example : Políticas públicas de buckets

Nessas regras, as políticas de exemplo a seguir são consideradas públicas.

```
{
  "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"}}
}
```

Como o Amazon S3 avalia uma política de bucket que contém concessões de acesso público e não público

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 IAM Access Analyzer para S3 para revisar buckets públicos

É possível usar o IAM Access Analyzer para S3 para revisar buckets com ACLs de bucket, políticas de bucket ou políticas de ponto de acesso que concedem acesso público. O IAM Access Analyzer para 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 IAM Access Analyzer para S3, é possível 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 IAM Access Analyzer para S3 pode relatar nenhuma descoberta para um bucket que uma análise do 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 IAM Access Analyzer 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 IAM Access Analyzer para S3, consulte [Revisar o acesso de bucket usando o IAM Access Analyzer para S3](#).

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

Operação	Permissões obrigatórias
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](#)
- [Configurar o bloqueio de acesso público para seus buckets do S3](#)

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 ter mais informações, consulte [Bloquear o acesso público ao armazenamento do Amazon S3](#).

Note

As configurações no nível da conta substituem as configurações de objetos individuais. Configurar a conta para bloquear o acesso público substituirá todas as configurações de acesso público feitas a objetos individuais na conta.

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](#). Para obter mais informações sobre pontos de acesso, consulte [Executar operações de bloqueio de acesso público em um ponto de acesso](#).

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](#).

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.

Usando a 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 dos 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](#).

```
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](#).

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 ter mais informações, consulte [Bloquear o acesso público ao armazenamento do Amazon S3](#).

É possível usar o console do S3, a AWS CLI, os AWS SDKs e a API REST para conceder acesso público a um ou mais buckets. Também é possível bloquear o acesso público a buckets que já são públicos. Para obter mais informações, consulte as seções abaixo.

Para definir as configurações do Bloqueio de Acesso Público para todos os buckets em sua conta, consulte [Configurar o bloqueio de acesso público para sua conta](#). 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](#).

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](#).

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](#).

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](#).

Usar a AWS CLI

Para bloquear o acesso público em um bucket ou excluir o Bloqueio de Acesso Público, use o `s3api` do serviço da AWS CLI. 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 dos AWS SDKs

Java

```
AmazonS3 client = AmazonS3ClientBuilder.standard()
    .withCredentials(<credentials>)
    .build();

client.setPublicAccessBlock(new SetPublicAccessBlockRequest()
    .withBucketName(<bucket-name>)
    .withPublicAccessBlockConfiguration(new PublicAccessBlockConfiguration()
        .withBlockPublicAcls(<value>)
        .withIgnorePublicAcls(<value>)
        .withBlockPublicPolicy(<value>)
        .withRestrictPublicBuckets(<value>)));
```


⚠ Important

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](#).

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 IAM Access Analyzer para S3

O IAM 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 IAM Access Analyzer 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 multirregional ou uma política de ponto de acesso. Com essas descobertas, você pode tomar medidas corretivas imediatas e precisas para restaurar o acesso ao bucket da forma desejada.

Ao revisar um bucket em risco no IAM Access Analyzer para S3, é possível bloquear todo o acesso público ao bucket com um único clique. Recomendamos que você bloqueie 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](#).

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 IAM Access Analyzer para S3 está disponível sem custo adicional no console do Amazon S3. O IAM Access Analyzer para S3 usa tecnologia do AWS Identity and Access Management (IAM) Access Analyzer. Para usar o IAM Access Analyzer para S3 no console do Amazon S3, é necessário acessar o console do IAM e habilitar o IAM Access Analyzer por região.

Para obter mais informações sobre o IAM Access Analyzer, consulte [O que é o IAM Access Analyzer?](#) no Guia do usuário do IAM. Para obter mais informações sobre o IAM Access Analyzer para S3, consulte as seções a seguir.

Important

- O IAM Access Analyzer para S3 requer um analisador de nível de conta. Para usar o IAM Access Analyzer para S3, é necessário acessar o IAM Access Analyzer e criar um analisador que tenha uma conta como a zona de confiança. Para obter mais informações, consulte [Habilitar o IAM Access Analyzer](#) no Guia do usuário do IAM.
- O IAM Access Analyzer para S3 não analisa a política de ponto de acesso vinculada a pontos de acesso entre contas. Esse comportamento ocorre porque o ponto de acesso e sua política estão fora da zona de confiança, ou seja, fora da conta. Os buckets que delegam acesso a um ponto de acesso entre contas estão listados em Buckets with public access (Buckets com acesso público) caso você não tenha aplicado a configuração de bloqueio de acesso público `RestrictPublicBuckets` ao bucket ou à conta. Quando você aplica a configuração de bloqueio de acesso público `RestrictPublicBuckets`, o bucket é relatado em Buckets com acesso de outras Contas da AWS, inclusive Contas da AWS de terceiros.

- Quando uma política de bucket ou a ACL do bucket é adicionada ou modificada, o IAM Access Analyzer gera e atualiza descobertas com base na alteração em até 30 minutos. As descobertas relacionadas às configurações de bloqueio de acesso público na conta podem não ser geradas nem atualizadas por até 6 horas após a alteração das configurações. As descobertas relacionadas a pontos de acesso multirregionais podem não ser geradas nem atualizadas até seis horas após o ponto de acesso multirregional ser criado, excluído ou você alterar sua política.

Tópicos

- [Quais informações o IAM Access Analyzer para S3 fornece?](#)
- [Habilitar o IAM Access Analyzer para S3](#)
- [Bloquear todo o acesso público](#)
- [Revisar e alterar o acesso ao bucket](#)
- [Arquivar descobertas de bucket](#)
- [Ativar uma descoberta de bucket arquivada](#)
- [Visualizar detalhes de descobertas](#)
- [Baixar um relatório do IAM Access Analyzer para S3](#)

Quais informações o IAM Access Analyzer para S3 fornece?

O IAM 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 IAM Access Analyzer 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 com acesso de outras Contas da AWS, inclusive Contas da AWS de terceiros são compartilhados condicionalmente com outras Contas da AWS, inclusive com contas fora de sua organização.

Para cada bucket, o IAM Access Analyzer para S3 fornece as seguintes informações:

- Bucket name
- Descoberto pelo Access Analyzer: quando o IAM Access Analyzer 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 e pontos de acesso entre contas 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 IAM Access Analyzer 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 IAM Access Analyzer para S3

Para usar o IAM Access Analyzer para S3, é necessário concluir as etapas de pré-requisito a seguir.

1. Conceda as permissões necessárias aos usuários.

Para obter mais informações, consulte [Permissões necessárias para usar o IAM Access Analyzer](#) 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 IAM Access Analyzer.

O IAM Access Analyzer para S3 requer um analisador de nível de conta. Para usar o IAM Access Analyzer para S3, é necessário criar um analisador que tenha uma conta como a zona de confiança. Para obter mais informações, consulte [Habilitar o IAM Access Analyzer](#) 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 Bloquear todo o acesso público no IAM Access Analyzer para S3. Quando você bloquear todo o acesso público a um bucket, nenhum acesso público será concedido. Recomendamos que você bloqueie 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 ter mais informações, consulte [Bloquear o acesso público ao armazenamento do Amazon S3](#).

Em eventos raros, o IAM Access Analyzer para S3 pode relatar nenhuma descoberta para um bucket que uma análise do 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 IAM Access Analyzer 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 IAM Access Analyzer 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 IAM Access Analyzer 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 resolvido e o bucket desaparece da listagem do IAM Access Analyzer para S3. Se você quiser revisar os buckets resolvidos, abra o IAM Access Analyzer 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 e pontos de acesso entre contas 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 (Analisador 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](#).

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](#).

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](#).

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](#).

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 IAM Access Analyzer para S3, mas é possível visualizá-las no IAM Access Analyzer.

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 IAM Access Analyzer para S3 para que você sempre saiba quais buckets são públicos ou compartilhados.

Como arquivar descobertas de bucket no IAM Access Analyzer 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 IAM Access Analyzer 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 revisita-las e alterar o status novamente para ativo, indicando que o bucket requer outra análise.

Como ativar uma descoberta de bucket arquivado no IAM Access Analyzer 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 IAM Access Analyzer no [console do IAM](#).

Para visualizar detalhes de descoberta no IAM Access Analyzer 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 IAM Access Analyzer para S3, escolha um bucket.
4. Escolha View details (Exibir detalhes).

Os detalhes de descobertas são abertos no IAM Access Analyzer no [console do IAM](#).

Baixar um relatório do IAM Access Analyzer 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 IAM Access Analyzer 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.

O IAM Access Analyzer para S3 é atualizado para mostrar 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](#)
- [Verificar um proprietário do bucket](#)
- [Exemplos](#)
- [Restrições e limitações](#)

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](#).

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 proprietária 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>
APIs REST do Amazon S3	<code>x-amz-expected-bucket-owner</code> Cabeçalho	<code>x-amz-source-expected-bucket-owner</code> Cabeçalho	<code>x-amz-expected-bucket-owner</code> Cabeçalho

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
<i>DOC-EXAMPLE-BUCKET1</i>	111122223333
<i>DOC-EXAMPLE-BUCKET2</i>	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 (valor numérico 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

A Propriedade de objeto do S3 é uma configuração no nível do bucket do Amazon S3 que você pode usar para controlar a propriedade de objetos carregados no bucket e desabilitar ou habilitar as [listas de controle de acesso \(ACLs\)](#). Por padrão, a Propriedade de objeto está definida com a configuração [Imposto pelo proprietário do bucket](#) e todas as ACLs estão desabilitadas. Quando as ACLs são desabilitadas, o proprietário do bucket possui todos os objetos do bucket e gerencia o acesso aos dados usando políticas de gerenciamento de acesso.

A maioria dos casos de uso modernos no Amazon S3 não requer mais o uso de ACLs e, portanto, recomendamos manter as ACLs desabilitadas exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com as ACLs desativadas, é possível usar políticas para controlar mais facilmente o acesso a cada objeto no bucket, independentemente de quem carregou os objetos no bucket.

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

- **Imposto pelo proprietário do bucket (padrão):** 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):** 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 manter as ACLs desabilitadas escolhendo a configuração imposta pelo proprietário do bucket e usar sua política de bucket para compartilhar dados com usuários fora da conta, conforme necessário. Essa abordagem simplifica o gerenciamento de permissões. É possível desabilitar as ACLs em buckets recém-criados e já existentes. Para buckets recém-criados, as ACLs são desabilitadas por padrão. 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 abranja todas as maneiras pelas quais você pretende conceder acesso ao bucket fora da conta. 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 predefinida `bucket-owner-full-control` 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, [adicione uma política de bucket](#) que permita somente o carregamento de objetos que usam essa ACL.

Para ver quais configurações da Propriedade de Objeto são aplicadas aos seus buckets, você pode usar as métricas da Lente de Armazenamento do Amazon S3. A Lente de Armazenamento do S3 é um recurso de análise de armazenamento em nuvem que você pode usar para obter visibilidade em toda a organização sobre o uso e a atividade do armazenamento de objetos. Para obter mais informações, consulte [Como usar a Lente de Armazenamento do S3 para encontrar configurações da Propriedade de Objeto](#).

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Configurações do Object Ownership

Esta tabela mostra como cada configuração do Object Ownership afeta ACLs, objetos, propriedade de objetos e carregamentos de objetos.

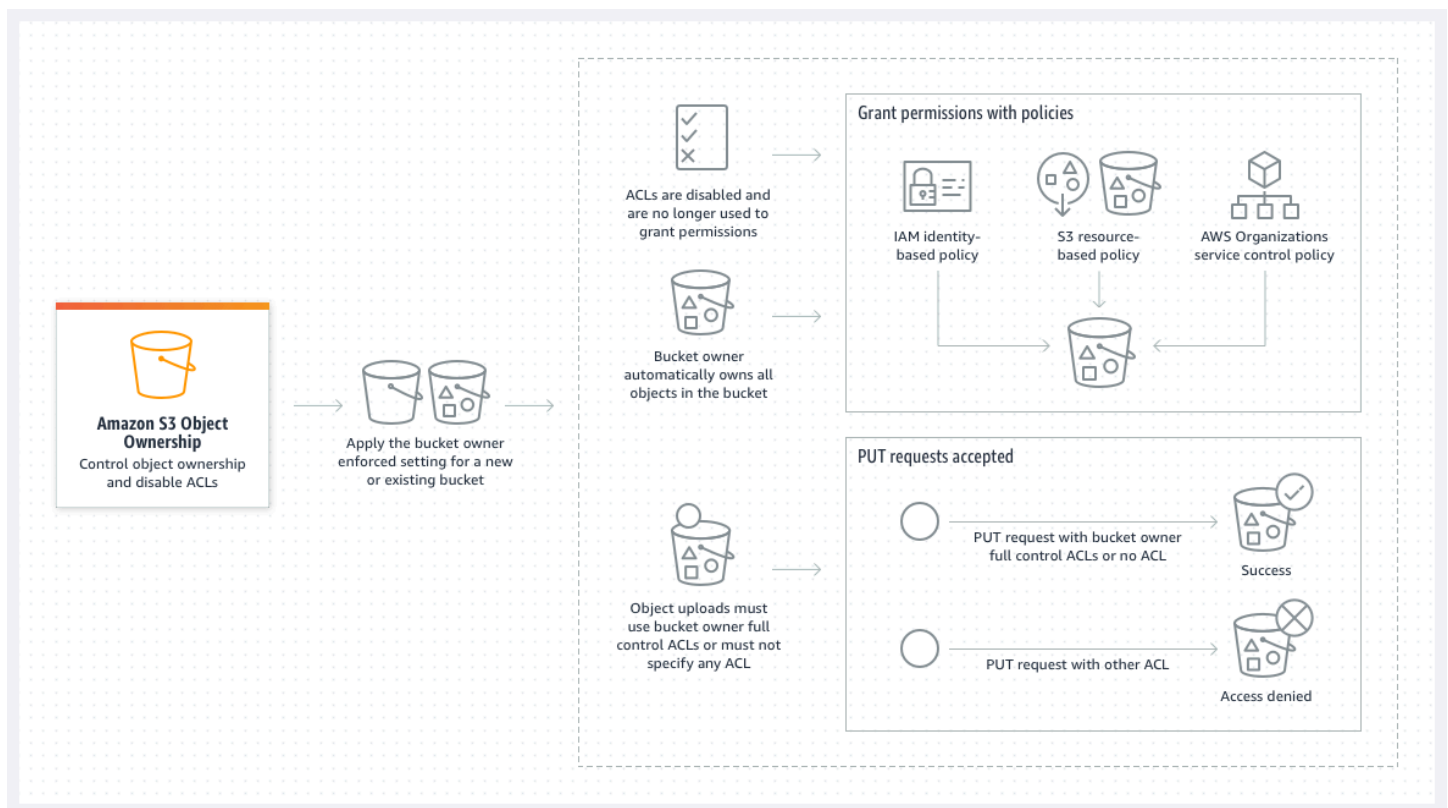
Configuração	Aplica-se a	Efeito na propriedade de objetos	Efeito nas ACLs	Carregamentos aceitos
Imposto pelo proprietário do bucket (padrão)	Objetos novos e existentes	O proprietário do bucket possui todos os objetos.	<p>As ACLs estão desabilitadas e não afetam mais as permissões de acesso ao bucket. As solicitações para definir ou atualizar ACLs falham. Porém, há suporte para solicitações de leitura de ACLs.</p> <p>O proprietário do bucket tem total propriedade e controle.</p> <p>O gravador de objetos não tem mais total propriedade e controle.</p>	Carregamentos com ACLs de controle total do proprietário do bucket ou carregamentos que não especificam uma ACL
Preferencial do proprietário do bucket	Novos objetos	Se um carregamento de objeto incluir a ACL <code>bucket-owner-full-control</code> pré-configurada, o proprietário do	<p>As ACLs podem ser atualizadas e podem conceder permissões.</p> <p>Se um carregamento de objeto incluir a</p>	Todos os carregamentos

Configuração	Aplica-se a	Efeito na propriedade de objetos	Efeito nas ACLs	Carregamentos aceitos
		<p>bucket terá o objeto.</p> <p>Os objetos carregados com outras ACLs são de propriedade da conta de gravação.</p>	ACL bucket-owner-full-control pré-configurada, o proprietário do bucket terá acesso de controle total, e o gravador de objetos não terá mais acesso de controle total.	
Autor do objeto	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

Quando a configuração *Imposto pelo proprietário do bucket* para a *Propriedade de objeto* é utilizada, as ACLs são desabilitadas e você automaticamente se torna o proprietário e assume controle total sobre todos os objetos do bucket sem realizar nenhuma outra ação. A configuração *Imposto pelo proprietário do bucket* é a configuração padrão para todos os buckets recém-criados. Ao adotar a configuração *Imposto pelo proprietário do bucket*, 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 Imposto pelo proprietário do bucket não adiciona uma nova versão de um 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](#).

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 que usam políticas para gerenciar o acesso a recursos do Amazon S3](#).

Habilitar ACLs novamente

É possível habilitar novamente as ACLs alterando a configuração Imposto pelo proprietário do bucket para outra configuração de Propriedade de objeto a qualquer momento. Se você usou ACLs de objeto para gerenciamento de permissões antes de adotar a configuração Imposto pelo proprietário do bucket e não migrou essas permissões de ACL de objeto para a política de bucket, essas permissões serão restauradas quando as ACLs forem habilitadas novamente. Além disso, os objetos gravados no bucket enquanto a configuração Imposto pelo proprietário do bucket foi usada ainda pertencem ao proprietário do bucket.

Por exemplo, ao mudar da configuração Imposto pelo proprietário do bucket para a configuração Autor do objeto, você, como proprietário do bucket, 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 Imposto pelo proprietário do bucket foi usada. Esses objetos não pertencem ao gravador de objetos, mesmo que você habilite as ACLs novamente.

Para obter instruções sobre como habilitar e gerenciar ACLs usando o AWS Management Console, a AWS Command Line Interface (CLI), a API REST ou AWS SDKs, consulte [Configurar ACLs](#).

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 Imposto 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 Imposto pelo proprietário do bucket falhará e retornará o código de erro [InvalidBucketAclWithObjectOwnership](#).

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 ter mais informações, consulte [Conceder acesso ao grupo de entrega de logs do S3 para registro em log de acesso ao servidor](#).

Para que o autor do objeto mantenha o controle total do objeto que ele carrega, Autor do objeto é a melhor configuração da Propriedade de Objeto 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](#).

Identificar as solicitações que exigiram uma ACL para autorização

Para identificar solicitações do Amazon S3 que exigiam ACLs para autorização, você pode usar o valor `aclRequired` nos logs de acesso do servidor do Amazon S3 ou do AWS CloudTrail. Se a solicitação exigiu uma ACL para autorização ou se você tem solicitações PUT que especificam uma ACL, a string é `Yes`. Se nenhuma ACL for necessária, ou se você estiver configurando uma ACL `bucket-owner-full-control` predefinida, ou se as solicitações forem permitidas pela política

de bucket, a string de valor `aclRequired` será “-” nos logs de acesso ao servidor do Amazon S3 e estará ausente no CloudTrail. Para obter mais informações sobre valores `aclRequired` esperados, consulte [Valores `aclRequired` para solicitações comuns do Amazon S3](#).

Se você tiver solicitações `PutBucketAcl` ou `PutObjectAcl` com cabeçalhos que concedam permissões baseadas em ACL, com exceção da ACL `bucket-owner-full-control` predefinida, será necessário remover esses cabeçalhos antes de poder desabilitar as ACLs. Caso contrário, as solicitações falharão.

Para todas as outras solicitações que exigiram uma ACL para autorização, migre essas permissões de ACL para políticas de bucket. Depois, remova todas as ACLs do bucket antes de habilitar a configuração imposta pelo proprietário do bucket.

Note

Não remova ACLs de objetos. Caso contrário, as aplicações que dependem de ACLs de objetos para obter permissões perderão o acesso.

Se você perceber que nenhuma solicitação exigiu uma ACL para autorização, será possível continuar desabilitando as ACLs. Para obter mais informações sobre como identificar solicitações, consulte [Usar logs de acesso ao servidor do Amazon S3 para identificar solicitações](#) e [Identificar solicitações do Amazon S3 usando o CloudTrail](#).

Revisar e atualizar políticas de bucket que usam chaves de condição relacionadas à ACL

Depois de adotar a configuração imposta 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](#) e [Visualizar a configuração do Object Ownership de um bucket do Amazon S3](#).

Desabilitar ACLs para todos os novos buckets

Por padrão, todos os novos buckets são criados usando a configuração Imposto pelo proprietário do bucket e as ACLs são desabilitadas. Recomendamos manter as ACLs desabilitadas. 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. As políticas são uma opção de controle de acesso simplificada e mais flexível. Com políticas de bucket e políticas de ponto de acesso, é possível definir regras que se aplicam amplamente a todas as solicitações para seus recursos do Amazon S3.

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 Imposto pelo proprietário do bucket em Propriedade de objeto) 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 Imposto 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](#).

Configurar o Object Ownership

É possível aplicar uma configuração de Propriedade de objeto usando o console do Amazon 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	put-bucket-ownership-controls	Cria ou modifica a configuração do Object Ownership para um bucket do S3 existente.
CreateBucket	create-bucket	Cria um bucket usando o cabeçalho de solicitação <code>x-amz-object-ownership</code> para especificar a configuração do Object Ownership.
GetBucketOwnershipControls	get-bucket-ownership-controls	Recupera a configuração do Object Ownership de um bucket do Amazon S3.
DeleteBucketOwnershipControls	delete-bucket-ownership-controls	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](#)
- [Configurar Object Ownership ao criar um bucket](#)
- [Configurar Object Ownership em um bucket existente](#)
- [Visualizar a configuração do Object Ownership de um bucket do Amazon S3](#)
- [Desabilitar ACLs de todos os novos buckets e aplicar Object Ownership](#)
- [Solução de problemas](#)

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 Imposto pelo proprietário do bucket para desabilitar ACLs falhará e retornará o código de erro [InvalidBucketAclWithObjectOwnership](#). 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](#).

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](#).

Para revisar e migrar permissões de ACL para políticas de bucket, consulte os tópicos a seguir.

Tópicos

- [Exemplos de políticas de bucket](#)
- [Usar o console do S3 para revisar e migrar permissões de ACL](#)
- [Usar a AWS CLI para revisar e migrar permissões de ACL](#)
- [Demonstrações de exemplo](#)

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 permissão à Conta da AWS **111122223333** para listar o conteúdo do bucket, você poderá escrever uma política de bucket que conceda as permissões s3:ListBucket, s3:ListBucketVersions e 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 no bucket tiver uma ACL READ que conceda acesso à Conta da AWS **111122223333**, você poderá escrever uma política de bucket que conceda as 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 **WRITE** que concede permissões para gravar objetos em um bucket

Se o bucket tiver uma ACL **WRITE** que conceda permissão à Conta da AWS **111122223333** para gravar objetos no bucket, você poderá escrever 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

Como revisar as permissões de ACL de 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 (Buckets), escolha o nome do bucket.

3. Escolha a aba Permissões.
4. Em Access control list (ACL) (Lista de controle de acesso [ACL]), revise suas permissões de ACL de bucket.

Como revisar as permissões de ACL 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.
4. Escolha a aba Permissões.
5. 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](#) e [Demonstrações de exemplo](#).

5. Escolha Salvar alterações.
6. [Atualize sua ACL de bucket](#) para remover concessões de ACL para outros grupos ou Contas da AWS.
7. [Adote a configuração Imposto pelo proprietário do bucket](#) em Propriedade de objeto.

Usar a AWS CLI para revisar e migrar permissões de ACL

1. Para retornar a ACL para o 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](#). Para adotar a configuração Imposto pelo proprietário do bucket e desabilitar ACLs, é necessário migrar essas permissões à 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](#).

2. Migre 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](#) e [Demonstrações de exemplo](#).

3. Para retornar a ACL para 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. [Adote a configuração Imposto pelo proprietário do bucket](#) em Propriedade de objeto.

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 registro em log de acesso ao servidor](#)
- [Conceder acesso público de leitura aos objetos do bucket](#)
- [Conceder acesso do Amazon ElastiCache for Redis ao bucket do Amazon S3](#)

Conceder acesso ao grupo de entrega de logs do S3 para registro em log de acesso ao servidor

Para adotar a configuração Imposto pelo proprietário do bucket a fim de desabilitar ACLs para um bucket de destino de logs de acesso ao servidor (também conhecido como bucket-alvo), é necessário migrar permissões de ACL de bucket para o grupo de entrega de logs do S3 para a entidade principal de serviço do registro em 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](#).

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",
        "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 do bucket de destino para a ACL padrão.

```
aws s3api put-bucket-acl --bucket DOC-EXAMPLE-BUCKET --acl private
```

3. [Aplique a configuração Imposto pelo proprietário do bucket](#) para a Propriedade de Objetos no 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/*"
```

2. [Adote a configuração Imposto pelo proprietário do bucket](#) em Propriedade de objeto.

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, será necessário migrar essas permissões para uma política de bucket antes de adotar a configuração Imposto 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

1. 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. [Adote a configuração Imposto pelo proprietário do bucket](#) em Propriedade de objeto.

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](#).

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, a Propriedade de Objetos do S3 é definida como a configuração Imposto pelo proprietário do bucket e as ACLs são desabilitadas para novos buckets. Com as ACLs desabilitadas, o proprietário do bucket possui cada objeto no bucket e gerencia o acesso aos dados exclusivamente usando políticas de gerenciamento de acesso. Recomendamos manter as ACLs desabilitadas, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente.

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

- Imposto pelo proprietário do bucket (padrão): 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): 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 aplicar a configuração **Bucket owner enforced** (Imposto pelo proprietário do bucket) ou **Bucket owner preferred** (Proprietário do bucket preferido), é necessário ter as seguintes permissões: `s3:CreateBucket` e `s3:PutBucketOwnershipControls`. Nenhuma permissão adicional é necessária ao criar um bucket com a configuração **Object writer** (Autor do objeto). Consulte mais informações sobre as permissões do Amazon S3 em [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço.

Important

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.

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 barra de navegação na parte superior da página, escolha o nome da Região da AWS exibida no momento. Em seguida, escolha a região na qual você deseja criar um bucket.

 Note

Para reduzir a latência e os custos e atender aos requisitos regulatórios, selecione uma região perto de você. Os objetos armazenados em uma região nunca saem dessa região, a menos que você os transfira para outra região. Para obter uma lista de Regiões da AWS do Amazon S3, consulte [Endpoints de AWS service \(Serviço da AWS\)](#) no Referência geral da Amazon Web Services.

3. No painel de navegação à esquerda, escolha Buckets.
4. Selecione Create bucket (Criar bucket).

A página Create bucket (Criar bucket) é aberta.

5. Em Configuração geral, visualize a Região da AWS onde o bucket será criado.
6. Em Tipo de bucket, escolha Propósito geral.
7. Para Nome do bucket, insira um nome para o bucket.

O nome do bucket deve:


- Seja exclusivo em 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) Regions).
- Ter entre 3 e 63 caracteres.
- Consistem apenas em letras minúsculas, números, pontos (.) e hífen (-). 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.
- Inicie e termine com uma letra ou um número.

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

 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.

8. O AWS Management Console permite que você copie as configurações de um bucket existente para seu novo bucket. Prossiga para a próxima etapa se você não quiser copiar as configurações de um bucket existente.

 Note

Essa opção:

- Não está disponível na AWS CLI, mas apenas no console.
- Não está disponível para buckets de diretório.
- Não copia a política do bucket existente para o novo bucket.

Para copiar as configurações de um bucket existente, em Copiar configurações do bucket existente, selecione Escolher bucket. A janela Escolher bucket é aberta. Encontre o bucket com as configurações que você gostaria de copiar e selecione Escolher bucket. A janela Escolher bucket é fechada e a janela Criar bucket é reaberta.

Agora, em Copiar configurações do bucket existente, você verá o nome do bucket selecionado. Você também verá uma opção Restaurar padrões, que pode ser usada para remover as configurações do bucket copiado. Revise as configurações restantes do bucket na página Criar bucket. Você verá que agora elas correspondem às configurações do bucket que você selecionou. Você pode pular para a etapa final.

9. 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

- Imposto pelo proprietário do bucket (padrão): 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 de acesso aos dados no bucket do S3. O bucket usa políticas exclusivamente para definir o controle de acesso.

Por padrão, as ACLs estão desabilitadas. A maioria dos casos de uso modernos no Amazon S3 não exige mais o uso de ACLs. Recomendamos manter as ACLs desabilitadas, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

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 Proprietário do bucket preferido para exigir que todos os uploads do Amazon S3 incluam a ACL `bucket-owner-full-control` pré-configurada, será possível [adicionar uma política de bucket](#) que só permita uploads de objetos que usem 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

A configuração padrão é imposto pelo proprietário do Bucket. Para aplicar a configuração padrão e manter as ACLs desabilitadas, somente a permissão `s3:CreateBucket` é necessária. Para habilitar ACLs, é necessário ter a permissão `s3:PutBucketOwnershipControls`.

10. Em Configurações de bloqueio de acesso público para este bucket, selecione as configurações de bloqueio de acesso público que deseja aplicar ao bucket.

Por padrão, todas as quatro configurações do Bloqueio de Acesso Público são habilitadas. Recomendamos que você mantenha todas as configurações habilitadas, a menos que precise desativar uma ou mais delas para seu caso de uso específico. Para obter mais informações sobre como bloquear o acesso público, consulte [Bloquear o acesso público ao armazenamento do Amazon S3](#).

Note

Para habilitar todas as configurações do Bloqueio de Acesso Público, somente a permissão `s3:CreateBucket` é necessária. Para desabilitar qualquer

configuração do Bloqueio de Acesso Público, é necessário ter a permissão `s3:PutBucketPublicAccessBlock`.

11. (Opcional) Em Bucket Versioning (Versionamento do bucket), você pode escolher se deseja manter variantes de objetos no bucket. Para obter mais informações sobre versionamento, consulte [Usando o versionamento em buckets do S3](#).

Para desabilitar ou habilitar o versionamento em seu bucket, escolha Disable (Desabilitar) ou Enable (Habilitar).

12. (Opcional) Em Tags (Etiquetas), você pode optar por adicionar etiquetas ao seu bucket. As etiquetas são pares de chave-valor usados para categorizar o armazenamento.

Para adicionar uma etiqueta, escolha Add Tag (Adicionar etiqueta) e insira uma chave e um valor para cada etiqueta.

13. Em Default encryption (Criptografia padrão), escolha Edit (Editar).

14. Para configurar a criptografia padrão, em Tipo de criptografia, selecione uma das seguintes opções:

- Chave gerenciada pelo Amazon S3 (SSE-S3)
- Chave do AWS Key Management Service (SSE-KMS)

Important

Se você usar a opção SSE-KMS em sua configuração de criptografia padrão, estará sujeito à cota de solicitações por segundo (RPS) do AWS KMS. Para obter mais informações sobre as cotas do AWS KMS e como solicitar um aumento de cota, consulte [Cotas](#) no Guia do desenvolvedor do AWS Key Management Service.

Os buckets e novos objetos são criptografados com criptografia do lado do servidor com uma chave gerenciada do Amazon S3 como nível básico da configuração de criptografia. Para obter mais informações sobre criptografia padrão, consulte [Definir o comportamento padrão da criptografia para os buckets do Amazon S3](#).


Para obter mais informações sobre como usar a criptografia no lado do servidor do Amazon S3 para criptografar seus dados, consulte [Usar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#).

15. Se você selecionou Chave do AWS Key Management Service (SSE-KMS), faça o seguinte:
- a. Em Chave do AWS KMS, especifique sua chave do KMS de uma das seguintes maneiras:
 - Para escolher entre uma lista de chaves do KMS disponíveis, selecione Escolher entre suas AWS KMS keys e escolha sua chave do KMS na lista de chaves disponíveis.

As chaves Chave gerenciada pela AWS (aws/s3) e as chaves gerenciadas pelo cliente são exibidas nessa lista. Para ter mais informações sobre chaves gerenciadas pelo cliente, consulte [Chaves de clientes e chaves da AWS](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para inserir o ARN da chave do KMS, escolha Inserir ARN da AWS KMS key e insira o ARN da chave do KMS no campo exibido.
- Para criar uma chave gerenciada pelo cliente no console do AWS KMS, selecione Criar uma chave do KMS.

Para ter mais informações sobre como criar uma AWS KMS key, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

 **Important**

Você só pode usar chaves do KMS disponíveis na mesma Região da AWS que o bucket. O console do Amazon S3 lista somente as primeiras 100 chaves do KMS na mesma região que o bucket. Para usar uma chave do KMS que não esteja listada, você deve inserir o ARN da chave do KMS. Se quiser usar uma chave do KMS que seja de propriedade de outra conta, primeiro você deverá ter permissão para usar a chave e, depois, inserir o ARN da chave do KMS. Para obter mais informações sobre permissões entre contas para chaves do KMS, consulte [Criar chaves do KMS que outras contas podem usar](#) no Guia do desenvolvedor do AWS Key Management Service. Para obter mais informações sobre a SSE-KMS, consulte [Especificando criptografia no lado do servidor com o AWS KMS \(SSE-KMS\)](#).

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 ter mais informações, consulte [Identificar](#)

[chaves do KMS 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 [Usar criptografia do lado do servidor com o AWS KMS \(SSE-KMS\)](#).

- b. Quando você configura seu bucket para usar a criptografia padrão com o SSE-KMS, também pode habilitar as chaves de bucket do S3. As chaves de bucket do S3 diminuem o custo de criptografia reduzindo o tráfego de solicitações do Amazon S3 para o AWS KMS. Para ter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3](#).

Para usar Chaves de bucket do S3, em Bucket Key (Chave de bucket), escolha Enable (Habilitar).

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

- a. Selecione Advanced settings (Configurações avançadas).

 Important

A ativação do bloqueio de objetos também permite o versionamento para o bucket. Depois de habilitar, é necessário definir as configurações de retenção padrão e retenção legal do bloqueio de objetos para proteger novos objetos contra exclusão ou substituição.

- b. Se quiser habilitar o bloqueio de objetos, escolha Enable (Habilitar), leia o aviso que aparece e confirme-o.

Para ter mais informações, consulte [Usar o bloqueio de objetos do S3](#).

 Note

Para criar um bucket com bloqueio de objetos habilitado, você deve ter as seguintes permissões: `s3:CreateBucket`, `s3:PutBucketVersioning` e `s3:PutBucketObjectLockConfiguration`.


17. Selecione Criar bucket.

Usando a 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 Imposto 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
```

 Important

Se você não definir a Propriedade de objeto ao criar um bucket usando a AWS CLI, a configuração padrão será `ObjectWriter` (ACLs habilitadas).

Usar o AWS SDK para Java

Este exemplo define a configuração Imposto pelo proprietário do bucket para um novo bucket usando o AWS SDK for Java:

```
// Build the ObjectOwnership for CreateBucket
CreateBucketRequest createBucketRequest = CreateBucketRequest.builder()
    .bucket(bucketName)
    .objectOwnership(ObjectOwnership.BucketOwnerEnforced)
    .build()

// Send the request to Amazon S3
s3client.createBucket(createBucketRequest);
```

Usar o AWS CloudFormation

Para usar o recurso `AWS::S3::Bucket` do AWS CloudFormation para definir a Propriedade de Objeto ao criar um bucket, consulte [OwnershipControls em AWS::S3::Bucket](#) no Guia do usuário do AWS CloudFormation.

Uso da API REST

Para adotar a configuração Imposto pelo proprietário do bucket para Propriedade de objeto do S3, use a operação de API `CreateBucket` com o cabeçalho da solicitação `x-amz-object-ownership` definido como `BucketOwnerEnforced`. Para obter informações e exemplos, consulte [CreateBucket](#) na Referência da API do Amazon Simple Storage Service.

Próximas etapas: depois de adotar as configurações Imposto pelo proprietário do bucket ou Proprietário do bucket preferido para a Propriedade de Objetos, você poderá seguir estas etapas:

- [Bucket owner enforced](#) (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](#) (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](#).

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, a Propriedade de Objetos do S3 é definida como a configuração Imposto pelo proprietário do bucket e as ACLs são desabilitadas para novos buckets. Com as ACLs desabilitadas, o proprietário do bucket possui cada objeto no bucket e gerencia o acesso aos dados exclusivamente usando políticas de gerenciamento de acesso. Recomendamos manter as ACLs desabilitadas, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente.

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

- Imposto pelo proprietário do bucket (padrão): 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): 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 Imposto 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](#).

Permissions (Permissões): para usar essa operação, é necessário ter a permissão `s3:PutBucketOwnershipControls`. Consulte mais informações sobre as permissões do Amazon S3 em [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço.

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 aba 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 \(aplicada pelo proprietário do bucket\)](#).

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](#) 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 Salvar.

Usando a 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`. Os valores válidos de propriedade são `BucketOwnerEnforced`, `BucketOwnerPreferred` ou `ObjectWriter`.

Este exemplo aplica a configuração Imposto 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}]"
```

Para obter informações sobre `put-bucket-ownership-controls`, consulte [put-bucket-ownership-controls](#) no Guia do usuário do AWS Command Line Interface.

Usar o AWS SDK para Java

Este exemplo aplica a configuração `BucketOwnerEnforced` à Propriedade de objeto em um bucket existente usando o 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 Amazon S3
s3client.putBucketOwnershipControls(putBucketOwnershipControlsRequest);
```

Usar o AWS CloudFormation

Para usar o AWS CloudFormation para aplicar uma configuração de Propriedade de Objeto a um bucket existente, consulte [AWS::S3::Bucket OwnershipControls](#) no Guia do usuário do AWS CloudFormation.

Uso da API REST

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 Imposto pelo proprietário do bucket ou Proprietário do bucket preferido para a Propriedade de Objetos, você poderá seguir estas etapas:

- [Bucket owner enforced](#) (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](#) (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 \(ACLs\)](#) e assumir a propriedade de cada objeto em seu bucket, simplificando o gerenciamento de acesso para dados armazenados no Amazon S3. Por padrão, a Propriedade de Objetos do S3 é definida como a configuração Imposto pelo proprietário do bucket e as ACLs são desabilitadas para novos buckets. Com as ACLs desabilitadas, o proprietário do bucket possui cada objeto no bucket e gerencia o acesso aos dados exclusivamente usando políticas de gerenciamento de acesso. Recomendamos manter as ACLs desabilitadas, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente.

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

- Imposto pelo proprietário do bucket (padrão): 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): 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](#). Para definir o Object Ownership para um bucket existente, consulte [Configurar Object Ownership em um bucket existente](#).

Permissões (Permissões): para usar essa operação, é necessário ter a permissão `s3:GetBucketOwnershipControls`. Consulte mais informações sobre as permissões do Amazon S3 em [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço.

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 aba Permissões.
4. Em Object Ownership, você pode ver as configurações de Object Ownership do bucket.

Usando a AWS CLI

Para recuperar a configuração da Propriedade de objeto do S3 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 da API REST

Para recuperar a configuração de Object Ownership para um bucket do S3, use a operação de API `GetBucketOwnershipControls`. Para ter 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 Imposto pelo proprietário do bucket para Propriedade de objeto do S3. 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. Para 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 serviços (SCPs) do AWS Organizations, conforme descrito na próxima seção.

Para aplicar a propriedade do objeto a novos objetos sem desabilitar ACLs, aplique a configuração preferencial do proprietário do bucket. Quando você aplica essa configuração, é altamente recomendável atualizar a política de bucket para exigir a ACL `bucket-owner-full-control` pré-configurada para todas as solicitações PUT para o bucket. Atualize também os clientes para enviar a ACL `bucket-owner-full-control` pré-configurada de outras contas para o bucket.

Tópicos

- [Desabilitar ACLs para todos os novos buckets \(aplicada pelo proprietário do bucket\)](#)
- [Exigir a ACL pré-configurada `bucket-owner-full-control` para operações PUT do Amazon S3 \(preferencial do proprietário do bucket\)](#)

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 um perfil do IAM específico, a menos que a configuração Imposto pelo proprietário do bucket seja adotada em Propriedade de objeto. 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 Imposto pelo proprietário do bucket para Propriedade de objeto 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 Propriedade de objeto, 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 Imposto pelo proprietário do bucket, como proprietário do bucket, você automaticamente terá propriedade e controle total sobre todos os objetos do 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 pré-configurada `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 pré-configurada `bucket-owner-full-control`, a operação falhará, e quem carregar 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 que usam políticas para gerenciar o acesso a recursos do Amazon S3](#).

Solução de problemas

Ao adotar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos do S3, as listas de controle de acesso (ACLs) serão desabilitadas e, como proprietário do bucket, você assumirá a propriedade de todos os objetos no 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 a ACL predefinida `bucket-owner-full-control` ou não especificar uma ACL, senão essas solicitações falharão. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

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

Ao adotar a configuração Imposto pelo proprietário do bucket em Propriedade de objeto, 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. Caso contrário, as operações PUT falharão.

O exemplo de comando `put-object` da AWS CLI a seguir inclui a ACL `public-read` predefinida.

```
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 Imposto pelo proprietário do bucket para desabilitar ACLs, essa operação falhará, e a pessoa que carregar 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 Imposto 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, a 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 Imposto pelo proprietário do bucket para Propriedade de objeto enquanto essas permissões de ACL não forem migradas para uma política de bucket e sua ACL de bucket não for redefinida para a ACL privada padrão. Para obter mais informações, consulte [Pré-requisitos para desabilitar ACLs](#).

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"
    }
  ]
}
```

O exemplo a seguir do comando put-bucket-ownership-controls da AWS CLI adota a configuração Imposto pelo proprietário do bucket para Propriedade de objeto:

```
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 ter mais informações, consulte [Monitorar o Amazon S3](#).

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

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](#).

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 ter mais informações, consulte [Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail](#).

Amazon GuardDuty

O [Amazon GuardDuty](#) é um serviço de detecção de ameaças que monitora continuamente contas, contêineres, workloads e dados no ambiente da AWS para identificar possíveis ameaças ou riscos à segurança nos buckets do S3. O GuardDuty também fornece um contexto detalhado sobre as ameaças detectadas. O GuardDuty monitora os logs de gerenciamento do AWS

CloudTrail em busca de ameaças e mostra informações relevantes de segurança. Por exemplo, o GuardDuty inclui fatores de uma solicitação de API, como o usuário que fez a solicitação, o local de onde ela foi feita e a API específica solicitada, que podem ser incomuns em seu ambiente. A [Proteção do S3 no GuardDuty](#) monitora os eventos de dados do S3 coletados pelo CloudTrail e identifica comportamentos possivelmente anômalos e mal-intencionados em todos os buckets do S3 do ambiente.

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](#).

AWS Trusted Advisor

O Trusted Advisor conta com as práticas recomendadas aprendidas com o atendimento a centenas de milhões de clientes da AWS. O Trusted Advisor inspeciona seu ambiente da AWS e faz recomendações quando há oportunidades para economizar dinheiro, melhorar a performance e a disponibilidade do sistema e ajuda a corrigir falhas de segurança. Todos os clientes da AWS têm acesso a cinco verificações do Trusted Advisor. Os clientes com um plano de suporte Business ou Enterprise podem ver todas as verificações do Trusted Advisor.

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](#)
- [Habilitar AWS Config](#)
- [Enable Amazon S3 server access logging](#)

- [Use CloudTrail](#)
- [Monitor Amazon Web Services security advisories](#)

Validação de conformidade para o Amazon S3

Audidores 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 download por meio do AWS Artifact. Para obter mais informações, consulte [Baixando 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](#) 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](#) 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](#), 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 as 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, archive-os ou exclua-os. Para obter mais informações, consulte [Gerenciando seu ciclo de vida de armazenamento](#).

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](#).

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](#).

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](#).

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](#).

Para obter mais informações sobre versionamento e bloqueio de objetos, consulte os seguintes tópicos: [Usando o versionamento em buckets do S3](#) [Usar o bloqueio de objetos do S3](#)

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 ser compatíveis com o Transport Layer Security (TLS) 1.2. Recomendamos também o suporte ao TLS 1.3. (Para obter mais informações sobre essa recomendação, consulte [Conexões de nuvem AWS mais rápidas com TLS 1.3](#) no Blog de segurança da AWS.) 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](#).

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](#)
- [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](#)
- [Habilitar AWS Config](#)

Práticas recomendadas 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 para o seu ambiente, trate-as como recomendações úteis em vez de requisitos.

Tópicos

- [Práticas recomendadas de segurança para o Amazon S3](#)
- [Práticas recomendadas de auditoria e monitoramento do Amazon S3](#)

Práticas recomendadas de segurança para o Amazon S3

As seguintes práticas recomendadas do Amazon S3 podem ajudar a evitar incidentes de segurança.

Desativar listas de controle de acesso (ACLs)

A Propriedade de objeto do S3 é uma configuração no nível do bucket do Amazon S3 que você pode usar para controlar a propriedade de objetos carregados no bucket e para desabilitar ou habilitar as ACLs. Por padrão, a Propriedade de objeto está definida com a configuração Imposto pelo proprietário do bucket e todas as ACLs estão desabilitadas. Quando as ACLs são desabilitadas, o proprietário do bucket possui todos os objetos do bucket e gerencia o acesso aos dados usando políticas de gerenciamento de acesso.

A maioria dos casos de uso no Amazon S3 não exige mais o uso de [listas de controle de acesso \(ACLs\)](#). Recomendamos desabilitar as ACLs, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Para desativar as ACLs e assumir a propriedade de cada objeto de seu bucket, aplique a configuração Imposto pelo proprietário do bucket para a Propriedade de objeto do S3. Ao desabilitar ACLs, você pode facilmente manter um bucket com objetos carregados por diferentes Contas da AWS.

Quando as ACLs são desabilitadas, o controle de acesso para os dados é baseado em políticas, como as seguintes:

- Políticas de usuário do AWS Identity and Access Management (IAM)
- Políticas de bucket do S3
- Políticas de endpoint de nuvem privada virtual (VPC)

- Políticas de controle de serviço (SCPs) do AWS Organizations

A desativação das ACLs simplifica o gerenciamento e a auditoria de permissões. Por padrão, as ACLs estão desabilitadas para novos buckets. Também é possível desabilitar ACLs para buckets existentes. Se você tiver um bucket existente que já contenha objetos, depois de desabilitar ACLs, as ACLs de objeto e bucket não farão mais parte do processo de avaliação de acesso. Em vez disso, o acesso é concedido ou negado com base nas políticas.

Antes de desabilitar as ACLs, faça o seguinte:

- Revise sua política de bucket para garantir que ela abranja todas as maneiras pelas quais você pretende conceder acesso ao bucket fora da conta.
- Redefina a ACL do bucket para o padrão (controle total para o proprietário do bucket).

Ao desabilitar as ACLs, os seguintes comportamentos ocorrem:

- O bucket aceita somente solicitações PUT que não especificam uma ACL ou solicitações PUT com ACLs de controle total do proprietário do bucket. Essas ACLs incluem a ACL predefinida `bucket-owner-full-control` ou formas equivalentes dessa ACL que são expressas 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 código de status de HTTP de 400 (Bad Request) com o código de erro `AccessControlListNotSupported`.

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

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 gravar no seu bucket do S3, ele não deve ser público. Veja abaixo algumas etapas que você pode seguir para bloquear o acesso público:

- Use o Bloqueio de Acesso Público do S3. Com o Bloqueio de Acesso Público do S3, é possível configurar facilmente os controles centralizados impostos para limitar o acesso público aos recursos do Amazon S3. Esses controles centralizados são aplicados independentemente de

como os recursos são criados. Para ter mais informações, consulte [Bloquear o acesso público ao armazenamento do Amazon S3](#).

- Identifique as políticas de bucket do Amazon S3 que permitam uma identidade curinga, como "Principal": "*" (que efetivamente significa "qualquer pessoa"). Procure também políticas que permitem uma ação curinga "*" (que efetivamente permite que o usuário realize qualquer ação no bucket do Amazon S3).
- Da mesma forma, procure 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 operação de API `ListBuckets` para verificar todos os buckets do Amazon S3. Depois, use `GetBucketAcl`, `GetBucketWebsite` e `GetBucketPolicy` para determinar se cada bucket tem controles de acesso de conformidade e uma configuração de conformidade.
- Use o [AWS Trusted Advisor](#) para inspecionar sua implementação do Amazon S3.
- Considere a implementação de controles contínuos de detecção usando o Regras do AWS Config gerenciado [s3-bucket-public-read-prohibited](#) e [s3-bucket-public-write-prohibited](#).

Para ter mais informações, consulte [Gerenciamento de identidade e acesso para o Amazon S3](#).

Identificar possíveis ameaças aos buckets do Amazon S3 usando o Amazon GuardDuty

O [Amazon GuardDuty](#) é um serviço de detecção de ameaças que identifica possíveis ameaças a contas, contêineres, workloads e dados no ambiente da AWS. Usando modelos de machine learning (ML) e recursos de detecção de anomalias e ameaças, o Amazon GuardDuty monitora continuamente diferentes fontes de dados para identificar e priorizar possíveis riscos de segurança e atividades mal-intencionadas no seu ambiente. Ao habilitar o GuardDuty, ele oferece a detecção de ameaças para fontes de dados fundamentais, incluindo [eventos de gerenciamento do AWS CloudTrail](#), logs de fluxo da VPC e logs de DNS. Para estender a detecção de ameaças aos eventos do plano de dados em buckets do S3, você pode habilitar o recurso [Proteção do S3 no GuardDuty](#). Esse recurso detecta ameaças como exfiltração de dados e acesso suspeito a buckets S3 por meio de nós Tor. O GuardDuty também estabelece um padrão normal de referência no ambiente e, quando identifica um comportamento possivelmente anômalo, fornece informações contextuais para ajudar a corrigir o bucket do S3 ou as credenciais da AWS possivelmente comprometidos. Consulte mais informações em [GuardDuty](#).

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,

recomendamos conceder apenas as permissões necessárias para executar uma tarefa. A implementação do acesso de privilégio 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 de políticas para o Amazon S3](#) e [Limites de permissões para entidades do IAM](#)
- [Como o Amazon S3 funciona com o IAM](#)
- [Visão geral da lista de controle de acesso \(ACL\)](#)
- [Políticas de controle de serviço](#)

Para obter orientação sobre o que considerar ao escolher um ou mais mecanismos precedentes, consulte [Gerenciamento de identidade e acesso para o Amazon S3](#).

Usar perfis do IAM para aplicações e Serviços da AWS que exigem acesso ao Amazon S3

Para que aplicações em execução no Amazon EC2 ou outros Serviços da AWS acessem recursos do Amazon S3, eles devem incluir credenciais válidas da AWS em suas solicitações de API da AWS. Recomendamos não 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, use um perfil do IAM para gerenciar credenciais temporárias para aplicações ou serviços que precisam de acesso ao Amazon S3. Quando você usa um perfil, 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 um AWS service (Serviço 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 do lado do servidor: todos os buckets do Amazon S3 têm criptografia configurada por padrão e todos os novos objetos que são carregados em um bucket do S3 são

automaticamente criptografados em repouso. A criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) é a configuração de criptografia padrão para todos os buckets no Amazon S3. Para usar um tipo diferente de criptografia, você pode especificar a criptografia do lado do servidor a ser usada nas solicitações PUT do S3 ou definir a configuração de criptografia padrão no bucket de destino.

O Amazon S3 também fornece as seguintes opções de criptografia do lado do servidor:

- Criptografia no lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS)
- Criptografia de camada dupla do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (DSSE-KMS)
- Criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C)

Para ter mais informações, consulte [Proteger os dados usando criptografia do lado do servidor](#).

- Criptografia do lado do cliente: criptografe dados do lado do cliente e carregue os 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 do lado do cliente](#).

Aplicar a criptografia de dados em trânsito

É possível usar HTTPS (TLS) para ajudar a evitar que invasores espionem ou manipulem tráfego de rede usando ataques person-in-the-middle (intermediários) ou similares.

Recomendamos permitir apenas conexões criptografadas por HTTPS (TLS) usando a condição [aws:SecureTransport](#) nas políticas de bucket do Amazon S3.

Important

Recomendamos que sua aplicação não fixe certificados TLS do Amazon S3, pois a AWS não permite a fixação de certificados publicamente confiáveis. O S3 renova automaticamente os certificados, o que pode ocorrer a qualquer momento antes da expiração do certificado. Renovar um certificado gera um novo par de chaves pública-privada. Se você fixou um certificado do S3 que foi renovado recentemente com uma

nova chave pública, não poderá se conectar ao S3 enquanto a aplicação não usar o novo certificado.

Considere também implementar controles de detecção contínuos usando a regra gerenciada [s3-bucket-ssl-requests-only](#) de AWS Config.

Considerar o uso do Bloqueio de objetos do S3

Com o Bloqueio de objetos do S3, é possível armazenar objetos usando um modelo “gravação única e várias leituras” (WORM). 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 os logs do AWS CloudTrail.

Para ter mais informações, consulte [Usar o bloqueio de objetos do S3](#).

Habilitar o versionamento do S3

O versionamento do S3 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 . 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 contínuos usando a regra gerenciada [s3-bucket-versioning-enabled](#) de AWS Config.

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

Considerar o uso da Replicação do S3 Entre Regiões Diferentes

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. Com a Replicação do S3 Entre Regiões Diferentes (CRR), é possível replicar dados entre Regiões da AWS distantes para ajudar a cumprir esses requisitos. A CRR permite a cópia assíncrona automática de objetos em buckets, em diferentes Regiões da AWS. Para ter mais informações, consulte [Replicação de objetos](#).

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 contínuos usando a regra gerenciada [s3-bucket-replication-enabled](#) de AWS Config.

Considerar o uso dos endpoints da VPC para acesso ao Amazon S3

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. Os endpoints da VPC podem ajudar a impedir que o tráfego atravesse a 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 endpoint da VPC específico usando políticas de bucket do S3.
- 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 ter mais informações, consulte [Controlar o acesso a partir de VPC endpoints com políticas de bucket](#).

Use serviços de segurança gerenciados da AWS para monitorar a segurança dos dados

Vários serviços gerenciados de segurança da AWS podem ajudar a identificar, avaliar e monitorar os riscos de segurança e conformidade dos dados do Amazon S3. Esses serviços também podem ajudar você a proteger seus dados contra esses riscos. Esses serviços incluem recursos automatizados de detecção, monitoramento e proteção que são projetados para escalar desde recursos do Amazon S3 para uma única Conta da AWS até recursos para organizações que abrangem milhares de contas.

Para ter mais informações, consulte [Monitorar a segurança dos dados com serviços de segurança gerenciados da AWS](#).

Práticas recomendadas 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. Para auditar seus recursos, recomendamos fazer o seguinte:

- Use o Editor de tags para identificar e marcar recursos sensíveis quanto a segurança ou auditoria; depois, use essas tags quando precisar procurar esses recursos. Para obter mais informações, consulte [Pesquisa de recursos para marcar](#) no Guia do usuário de recursos da AWS.
- Use o S3 Inventory para auditar e gerar relatórios sobre o status da replicação e criptografia de seus objetos para necessidades empresariais, regulatórias e de conformidade. Para ter mais informações, consulte [Inventário do Amazon S3](#).
- Crie grupos de recursos para seus recursos do Amazon S3. Para obter mais informações, consulte [O que são grupos de atributos?](#) no Guia do usuário do AWS Resource Groups.

Implementar o monitoramento usando ferramentas de monitoramento da AWS

O monitoramento é uma parte importante para manter a confiabilidade, segurança, disponibilidade e 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, é possível monitorar métricas do Amazon CloudWatch para o Amazon S3, particularmente as métricas PutRequests, GetRequests, 4xxErrors e DeleteRequests. Para obter mais informações, consulte [Monitoramento de métricas com o Amazon CloudWatch e Monitorar o Amazon S3](#).

Para obter um segundo exemplo, consulte, [Exemplo: atividade do bucket do Amazon S3](#). Este exemplo descreve como criar um alarme do CloudWatch que é ativado quando uma chamada de API do Amazon S3 é feita para PUT ou DELETE uma política do bucket, um ciclo de vida do bucket ou uma configuração de replicação do bucket ou para PUT uma 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](#).

Considere também implementar controles de detecção contínuos usando a regra gerenciada [s3-bucket-logging-enabled](#).

Usar o AWS CloudTrail

O AWS CloudTrail fornece um registro das ações executadas por um usuário, uma função ou um AWS service (Serviço da AWS) no Amazon S3. É possível usar as informações coletadas pelo CloudTrail para determinar o seguinte:

- A solicitação feita ao Amazon S3
- O endereço IP do qual a solicitação foi feita.
- Quem fez a solicitação.
- Quando a solicitação foi feita
- Detalhes adicionais sobre a solicitação

Por exemplo, é possível identificar entradas do CloudTrail para ações PUT que afetam o acesso a dados, em particular PutBucketAc1, PutObjectAc1, 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 [Registrar eventos de dados](#), 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 é compatível com 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](#). 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](#).

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 ajuda a avaliar, fazer auditoria e analisar as configurações de seus recursos da AWS. O AWS Config monitora as configurações de recurso, para que seja possível avaliar as configurações gravadas com relação às configurações seguras desejadas. Com o AWS Config, você pode fazer o seguinte:

- Revisar as alterações nas configurações e nas relações entre recursos da AWS
- Investigar históricos detalhados de configurações de recursos
- Determinar sua conformidade geral com relação às configurações especificadas nas diretrizes internas.

O uso do AWS Config 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.

Important

As regras gerenciadas do AWS Config só oferecem suporte a buckets de uso geral ao avaliar os recursos do Amazon S3. O AWS Config não registra alterações de configuração em buckets de diretório. Para obter mais informações, consulte [AWS Config Managed Rules](#) e [List of AWS Config Managed Rules](#) no Guia do desenvolvedor do AWS Config.

Para obter um exemplo de como usar o AWS Config, 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.

Descobrir dados sigilosos usando o Amazon Macie

O Amazon Macie é um serviço de segurança que descobre dados sigilosos usando machine learning e correspondência entre padrões. O Macie fornece visibilidade sobre os riscos de segurança de dados e permite a proteção automatizada contra esses riscos. Com o Macie, é possível automatizar a descoberta e a emissão de relatórios de dados sigilosos em seu

patrimônio de dados do Amazon S3 para compreender melhor os dados armazenados pela organização no S3.

Para detectar dados sigilosos com o Macie, é possível usar técnicas e critérios incorporados projetados para detectar uma lista grande e crescente de tipos de dados sigilosos para muitos países e regiões. Esses tipos de dados sigilosos incluem vários tipos de informações de identificação pessoal (PII), dados financeiros e dados de credenciais. Também é possível usar critérios personalizados definidos por você: expressões regulares que definem padrões de texto com os quais deve haver correspondência e, opcionalmente, sequências de caracteres e regras de proximidade que refinam os resultados.

Se o Macie detectar dados sigilosos em um objeto do S3, ele gerará uma descoberta de segurança para notificar você. Essa descoberta fornece informações sobre o objeto afetado, os tipos e o número de ocorrências dos dados sigilosos encontrados pelo Macie e detalhes adicionais para ajudar você a investigar o bucket e o objeto do S3 afetados. Para obter mais informações, consulte o [Guia do usuário da Amazon Macie](#).

Uso da Lente de Armazenamento do S3

A Lente de Armazenamento do S3 é um recurso de análise de armazenamento em nuvem que você pode usar para obter visibilidade em toda a organização sobre o uso e a atividade do armazenamento de objetos. A Lente de Armazenamento do S3 também analisa as métricas para fornecer recomendações contextuais que você pode usar para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados.

Com a Lente de Armazenamento do S3, é possível usar métricas para gerar insights resumidos, como descobrir quanto armazenamento você tem em toda a organização ou quais são os buckets e prefixos que mais crescem. Você também pode usar as métricas da Lente de Armazenamento do S3 para identificar oportunidades de otimização de custos, implementar práticas recomendadas de proteção de dados e gerenciamento de acesso, e melhorar a performance das workloads de aplicações.

Por exemplo, você pode identificar buckets que não têm regras de Ciclo de Vida do S3 para anular multipart uploads que estão incompletos há mais de 7 dias. Você também pode identificar buckets que não estão seguindo as práticas recomendadas de proteção de dados, como usar a Replicação do S3 ou o Versionamento do S3. Para obter mais informações, consulte [Noções básicas do Amazon S3 Storage Lens](#).

Monitorar as recomendações de segurança da AWS

Recomendamos verificar regularmente as recomendações de segurança publicadas no Trusted Advisor para sua Conta da AWS. Especificamente, procure avisos sobre buckets do Amazon S3 com “permissões de acesso em aberto”. É possível fazer isso de maneira programática 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 usa esse endereço de e-mail para entrar em contato sobre problemas de segurança emergentes que poderiam afetar você.

Problemas operacionais da AWS com grande impacto são publicados no [AWS Health Dashboard Service Health](#). Eles também são publicados em contas individuais por meio do AWS Health Dashboard. Para obter mais informações, consulte a [documentação do AWS Health](#).

Monitorar a segurança dos dados com serviços de segurança gerenciados da AWS

Vários serviços gerenciados de segurança da AWS podem ajudar a identificar, avaliar e monitorar os riscos de segurança e conformidade dos dados do Amazon S3. Esses serviços também podem ajudar você a proteger seus dados contra esses riscos. Esses serviços incluem recursos automatizados de detecção, monitoramento e proteção que são projetados para escalar desde recursos do Amazon S3 para uma única Conta da AWS até recursos para organizações que abrangem milhares de Contas da AWS.

Os serviços de detecção e resposta da AWS podem ajudar a identificar possíveis configurações incorretas de segurança, ameaças ou comportamentos inesperados, para que você possa responder rapidamente a atividades possivelmente não autorizadas ou mal-intencionadas em seu ambiente. Os serviços de proteção de dados da AWS podem ajudar a monitorar e proteger seus dados, contas e workloads contra o acesso não autorizado. Eles também podem ajudar a descobrir dados sigilosos, como informações de identificação pessoal (PII), em seu conjunto de dados do Amazon S3.

Para ajudar a identificar e avaliar riscos de segurança e conformidade de dados, os serviços de segurança gerenciados da AWS geram descobertas para notificar você sobre possíveis eventos ou problemas de segurança com os dados do Amazon S3. As descobertas fornecem detalhes relevantes que você pode usar para investigar, avaliar e tomar medidas com relação a esses riscos de acordo com seus fluxos de trabalho e políticas de resposta a incidentes. É possível acessar os dados das descobertas diretamente usando cada serviço. Também é possível enviar os dados para outras aplicações, serviços e sistemas, como o sistema de gerenciamento de incidentes e eventos de segurança (SIEM).

Para monitorar a segurança dos dados do Amazon S3, considere usar estes serviços de segurança gerenciados da AWS.

Amazon GuardDuty

O Amazon GuardDuty é um serviço de detecção de ameaças que monitora continuamente suas Contas da AWS e workloads em busca de atividades mal-intencionadas e fornece descobertas de segurança detalhadas para visibilidade e correção.

Com o recurso de proteção do S3 no GuardDuty, é possível configurar o GuardDuty para analisar eventos de gerenciamento e de dados do AWS CloudTrail para os recursos do Amazon S3. Depois, o GuardDuty monitora esses eventos em busca de atividades mal-intencionadas e

suspeitas. Para fundamentar a análise e identificar possíveis riscos de segurança, o GuardDuty usa feeds de inteligência de ameaças e machine learning.

O GuardDuty pode monitorar diferentes tipos de atividade para os recursos do Amazon S3. Por exemplo, os eventos de gerenciamento do CloudTrail para o Amazon S3 incluem operações em nível de bucket, como `ListBuckets`, `DeleteBucket` e `PutBucketReplication`. Os eventos de dados do CloudTrail para o Amazon S3 incluem operações em nível de objeto, como `GetObject`, `ListObjects` e `PutObject`. Se o GuardDuty detectar atividades anômalas ou possivelmente mal-intencionadas, ele gerará uma descoberta para notificar você.

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

Amazon Detective

O Amazon Detective simplifica o processo investigativo e ajuda você a conduzir investigações de segurança mais rápidas e eficazes. O Detective fornece agregações de dados, resumos e contexto predefinidos que podem ajudar a analisar e avaliar a natureza e a extensão dos possíveis problemas de segurança.

O Detective extrai automaticamente eventos baseados em tempo, como chamadas de API do AWS CloudTrail e logs de fluxo da Amazon VPC para os recursos da AWS. Ele também ingere descobertas geradas pelo Amazon GuardDuty. Depois, o Detective usa machine learning, análise estatística e teoria de grafos para gerar visualizações que ajudam você a conduzir investigações de segurança eficazes com maior rapidez.

Essas visualizações fornecem uma visão unificada e interativa dos comportamentos dos recursos e das interações entre eles ao longo do tempo. É possível explorar esse grafo de comportamento para examinar ações possivelmente mal-intencionadas, como tentativas malsucedidas de login ou chamadas de API suspeitas. Também é possível ver como essas ações afetam recursos, como buckets e objetos do S3.

Para obter mais informações, consulte o [Guia de administração do Amazon Detective](#).

IAM Access Analyzer

O AWS Identity and Access Management Access Analyzer (IAM Access Analyzer) pode ajudar a identificar os recursos compartilhados com uma entidade externa. Também é possível usar o IAM Access Analyzer para validar as políticas do IAM em relação à gramática de políticas e às práticas recomendadas, além de gerar políticas do IAM com base na atividade de acesso nos logs do AWS CloudTrail.

O IAM Access Analyzer usa raciocínio baseado em lógica para analisar políticas de recursos em seu ambiente da AWS, como políticas de bucket. Com o IAM Access Analyzer para S3, você recebe alertas quando um bucket do S3 é configurado para permitir o acesso a qualquer pessoa na internet ou a outras Contas da AWS, incluindo contas fora da organização. Por exemplo, o IAM Access Analyzer para S3 pode relatar 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 multirregional ou uma política de ponto de acesso. Para cada bucket público ou compartilhado, você recebe descobertas que indicam a origem e o nível de acesso público ou compartilhado. Com essas descobertas, você pode tomar medidas corretivas imediatas e precisas para restaurar o acesso ao bucket da forma desejada.

Para ter mais informações, consulte [Revisar o acesso de bucket usando o IAM Access Analyzer para S3](#).

Amazon Macie

O Amazon Macie é um serviço de segurança de dados que descobre dados sigilosos usando machine learning e correspondência de padrões, fornece visibilidade dos riscos de segurança de dados e permite proteção automatizada contra esses riscos.

Com o Macie, é possível automatizar a descoberta e a emissão de relatórios de dados sigilosos nos buckets do S3 para compreender melhor os dados armazenados pela organização no Amazon S3. Para detectar dados sigilosos, é possível usar critérios e técnicas incorporados que o Macie fornece, critérios personalizados que você define ou uma combinação dos dois. Se o Macie detectar dados sigilosos em um objeto do S3, ele gerará uma descoberta para notificar você. Essa descoberta fornece informações sobre o bucket e o objeto afetados, os tipos e o número de ocorrências dos dados sigilosos encontrados pelo Macie e detalhes adicionais para facilitar sua investigação.

O Macie também fornece estatísticas e outros dados que oferecem uma visão sobre o procedimento de segurança dos dados do Amazon S3 e avalia e monitora automaticamente os buckets do S3 quanto à segurança e ao controle de acesso. Se o Macie detectar um possível problema com a segurança ou a privacidade dos dados, como um bucket que se torna acessível ao público, ele gerará uma descoberta para você revisar e corrigir conforme necessário.

Para obter mais informações, consulte o [Guia do usuário da Amazon Macie](#).

AWS Security Hub

O AWS Security Hub é um serviço de gerenciamento de procedimento de segurança que realiza verificações de práticas recomendadas de segurança, agrega alertas e descobertas de várias fontes em um único formato e permite a correção automatizada.

O Security Hub coleta e fornece dados de descobertas de segurança de Serviços da AWS e soluções de segurança do AWS Partner Network integradas, como Amazon Detective, Amazon GuardDuty, IAM Access Analyzer e Amazon Macie. Ele também gera suas próprias descobertas executando verificações de segurança automatizadas e contínuas com base nas práticas recomendadas da AWS e em padrões compatíveis do setor.

Depois, o Security Hub correlaciona e consolida as descobertas nos provedores para ajudar a priorizar as mais significativas. Ele também aceita ações personalizadas, que você pode usar para invocar respostas ou ações de correção para classes específicas de descobertas.

Com o Security Hub, é possível avaliar o status de segurança e conformidade dos recursos do Amazon S3 e fazer isso como parte de uma análise mais ampla do procedimento de segurança da organização em Regiões da AWS individuais e em várias regiões. Isso inclui analisar as tendências de segurança e identificar os problemas de segurança de maior prioridade. Também é possível agregar descobertas de várias Regiões da AWS e monitorar e processar dados de descobertas agregadas de uma única região.

Para obter mais informações, acesse [Controles do Amazon Simple Storage Service](#) no Guia do usuário do AWS Security Hub.

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.

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Tópicos

- [Usando o versionamento em buckets do S3](#)
- [Usar o AWS Backup para o Amazon S3](#)
- [Trabalhando com objetos arquivados](#)
- [Usar o bloqueio de objetos do S3](#)
- [Uso de classes de armazenamento do Amazon S3](#)
- [Amazon S3 Intelligent-Tiering](#)
- [Gerenciando seu ciclo de vida de armazenamento](#)
- [Inventário do Amazon S3](#)
- [Replicação de objetos](#)
- [Categorizando seu armazenamento usando tags](#)
- [Usar tags de alocação de custos para buckets do S3](#)
- [Relatórios de uso e faturamento do Amazon S3](#)
- [Filtragem e recuperação de dados usando o Amazon S3 Select](#)
- [Executar operações em lote de grande escala em objetos do Amazon S3](#)

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](#).

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](#).

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](#).
- 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](#).

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 ter mais informações, consulte [Gerenciando seu ciclo de vida de armazenamento](#). Para ter informações sobre como criar configurações de ciclo de vida do S3 usando o AWS Management Console, a AWS CLI, os AWS SDKs ou a API REST, consulte [Definir uma configuração de ciclo de vida em um bucket](#).

Important

Se você tem uma configuração de ciclo de vida de expiração do objeto em seu bucket sem versionamento e quiser manter o mesmo comportamento de exclusão permanente quando ativar o versionamento, precisará adicionar uma configuração de expiração de versão não corrente. A política de expiração do ciclo de vida não corrente gerencia as exclusões de versões de objeto não correntes no bucket habilitado para versão. (Um bucket habilitado para versionamento mantém uma versão atual e zero ou mais versões desatualizadas de

objetos.) Para obter mais informações, consulte [Definir uma configuração de ciclo de vida em um bucket](#).

Para obter informações sobre como trabalhar com o Versionamento do S3, consulte os tópicos a seguir.

Tópicos

- [Como funciona o Versionamento do S3](#)
- [Habilitar o versionamento em buckets](#)
- [Configurando a exclusão de MFA](#)
- [Trabalhar com objetos em um bucket com versionamento habilitado](#)
- [Trabalhar com objetos em um bucket com versionamento suspenso](#)

Como funciona o Versionamento do S3

Você pode usar o versionamento do S3 para manter várias versões de um objeto em um bucket com o intuito de restaurar objetos excluídos ou substituídos acidentalmente. Por exemplo, se você aplicar o versionamento do S3 a um bucket, as seguintes alterações ocorrerão:

- 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 ter mais informações, consulte [Excluir versões de objetos de um bucket com versionamento habilitado](#).
- Se você substituir um objeto, o Amazon S3 adicionará uma nova versão do objeto no bucket. A versão anterior permanecerá no bucket e se tornará uma versão desatualizada. Você poderá 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, e não 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](#).) 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 `Enabled`.

```
<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

Quando você habilitar o versionamento em um bucket pela primeira vez, poderá levar um curto período para que a alteração seja totalmente propagada. Recomendamos que você aguarde 15 minutos depois de habilitar o versionamento para emitir operações de gravação (PUT ou DELETE) em objetos no bucket.

O proprietário do bucket e todos os usuários autorizados do AWS Identity and Access Management (IAM) podem habilitar o versionamento. O proprietário do bucket é a Conta da AWS que criou o bucket. Para obter mais informações sobre permissões, consulte [Gerenciamento de identidade e acesso para o Amazon S3](#).

Para obter mais informações sobre como habilitar e desabilitar o versionamento do S3 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"](#).

Tópicos

- [IDs de versão](#)
- [Fluxos de trabalho de versionamento](#)

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. Por exemplo, em um bucket, você pode ter dois objetos com a mesma chave (nome do objeto), mas diferentes IDs de versão, como `photo.gif` (versão 111111) e `photo.gif` (versão 121212).

Diagrama que retrata um bucket com versionamento habilitado que tem dois objetos com a mesma chave e IDs de versão diferentes.

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 (`null`), o conteúdo e as permissões continuarão os mesmos. Depois que habilitar o versionamento do S3, 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+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo
```

Note

Por simplicidade, os outros exemplos neste tópico usam IDs muito mais curtos.

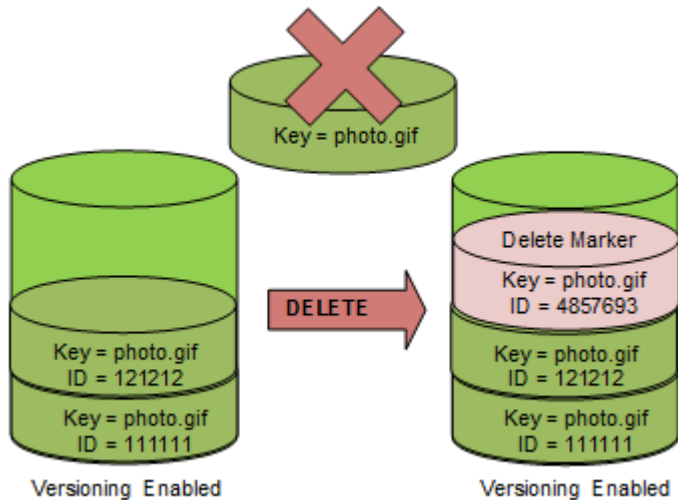
Fluxos de trabalho de versionamento

Quando você PUT um objeto em um bucket com versionamento ativado, a versão desatualizada não é substituída. Conforme mostrado na seguinte figura, quando uma nova versão de `photo.gif` é PUT em um bucket que já contém um objeto com o mesmo nome, o seguinte comportamento ocorre:

- O objeto original (ID = 111111) permanece no bucket.
- O Amazon S3 gera um novo ID de versão (121212) e adiciona essa versão mais recente do objeto ao bucket.

Com essa funcionalidade, você poderá recuperar uma versão anterior de um objeto se um objeto tiver sido sobrescrito ou excluído acidentalmente.

Quando você aplica DELETE em 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 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 ter mais informações, consulte [Recuperando versões de objeto de um bucket habilitado para versionamento](#).

Você pode excluir permanentemente um objeto, especificando a versão que deseja excluir. Somente o proprietário de um bucket do Amazon S3 ou um usuário do IAM autorizado pode excluir uma versão permanentemente. Se a operação DELETE especificar o `versionId`, essa versão do objeto será excluída permanentemente e o Amazon S3 não inserirá um marcador de exclusão.

Você pode obter segurança adicional configurando um bucket para habilitar a exclusão com autenticação multifator (MFA). Quando você habilita a exclusão com MFA para um bucket, 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 ter mais informações, consulte [Configurando a exclusão de MFA](#).

Quando novas versões são criadas para um objeto?

Novas versões de objetos são criadas somente quando você aplica PUT em um novo objeto. Esteja ciente de que certas ações como CopyObject funcionam implementando uma operação PUT.

Algumas ações que modificam o objeto atual não criam uma versão porque elas não aplicam PUT em 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 (Serviço indisponível) recebidas para solicitações PUT ou DELETE de objetos do Amazon S3 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 ter mais informações, consulte a seção Versionamento do S3 de [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 habilitar o versionamento em um bucket pela primeira vez, poderá levar até 15 minutos para que a alteração seja totalmente propagada. Recomendamos que você aguarde 15 minutos depois de habilitar o versionamento para emitir operações de gravação (PUT ou DELETE) em objetos no bucket. As operações de gravação emitidas antes da conclusão dessa conversão podem ser aplicadas a objetos sem versões.

Para obter mais informações sobre o S3 Versioning, consulte [Usando o versionamento em buckets do S3](#). 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](#).

Para saber mais sobre como usar o Versionamento do S3 para proteger dados, consulte [Tutorial: Protecting data on Amazon S3 against accidental deletion or application bugs using S3 Versioning, S3 Object Lock, and S3 Replication](#) (Tutorial: Proteção de dados no Amazon S3 contra exclusão acidental ou bugs de aplicações usando o Versionamento do S3, o Bloqueio de Objeto do S3 e a Replicação do S3).

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](#).) 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 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 [Gerenciamento de identidade e acesso para o Amazon S3](#).

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 (Suspend) 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](#).

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](#).

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](#).

```
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](#).

```
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 \(Boto\)](#).

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.

Para identificar buckets que têm exclusão de MFA habilitada, você pode usar as métricas da Lente de Armazenamento do Amazon S3. A Lente de Armazenamento do S3 é um recurso de análise de armazenamento em nuvem que você pode usar para obter visibilidade em toda a organização sobre o uso e a atividade do armazenamento de objetos. Para obter mais informações, consulte [Avaliar a atividade e o uso do armazenamento com o S3 Storage Lens](#). Para obter uma lista completa de métricas, consulte o [Glossário de métricas da Lente de Armazenamento do S3](#).

O proprietário do bucket, a Conta da AWS que criou o bucket (conta raiz) e todos os usuários autorizados 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](#).

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](#).

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 [Exigir MFA](#).

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

Usando 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 da API REST

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 ter mais informações, consulte [Gerenciando seu ciclo de vida de armazenamento](#).

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](#).

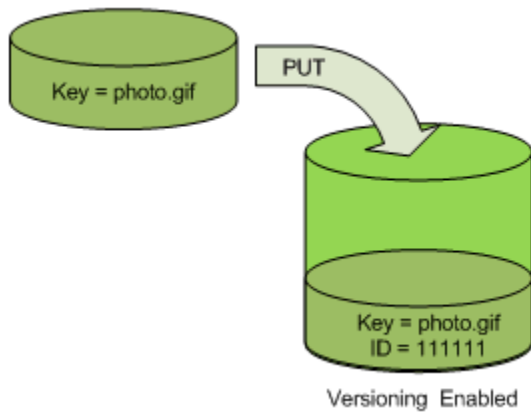
Tópicos

- [Adicionar objetos a buckets com versionamento habilitado](#)
- [Listar objetos em um bucket com versionamento habilitado](#)
- [Recuperando versões de objeto de um bucket habilitado para versionamento](#)
- [Excluir versões de objetos de um bucket com versionamento habilitado](#)
- [Configurando permissões de objeto com versão](#)

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 CopyObject) 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](#). 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](#).

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](#). 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 `PutBucketVersioning`.

Para obter mais informações, consulte [PutBucketVersioning](#) na Referência da API do Amazon Simple Storage Service.

2. Envie uma solicitação `PUT`, `POST` ou `CopyObject` 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](#). Para listar os objetos em um bucket do 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 ter mais informações, consulte [Exibir uma visão geral do objeto no console do Amazon S3](#).

Note

Para acessar versões de objetos anteriores a 300 versões, use a AWS CLI ou o URL do objeto.

⚠ 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](#).

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.

i 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](#).

```
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
```

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListVersionsRequest;
import com.amazonaws.services.s3.model.S3VersionSummary;
import com.amazonaws.services.s3.model.VersionListing;

public class ListKeysVersioningEnabledBucket {

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

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

            // Retrieve the list of versions. If the bucket contains more versions
            // than the specified maximum number of results, Amazon S3 returns
            // one page of results per request.
            ListVersionsRequest request = new ListVersionsRequest()
                .withBucketName(bucketName)
                .withMaxResults(2);
            VersionListing versionListing = s3Client.listVersions(request);
            int numVersions = 0, numPages = 0;
            while (true) {
                numPages++;
                for (S3VersionSummary objectSummary :
versionListing.getVersionSummaries()) {
                    System.out.printf("Retrieved object %s, version %s\n",
                        objectSummary.getKey(),
                        objectSummary.getVersionId());
                    numVersions++;
                }
                // Check whether there are more pages of versions to retrieve. If
                // there are, retrieve them. Otherwise, exit the loop.
                if (versionListing.isTruncated()) {
                    versionListing =
s3Client.listNextBatchOfVersions(versionListing);
                } else {
                    break;
                }
            }
        }
    }
}
```

```
    }
    System.out.println(numVersions + " object versions retrieved in " +
numPages + " pages");
  } catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
  } catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
  }
}
}
```

.NET

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

```
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 ter 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=
```

Usando a AWS CLI

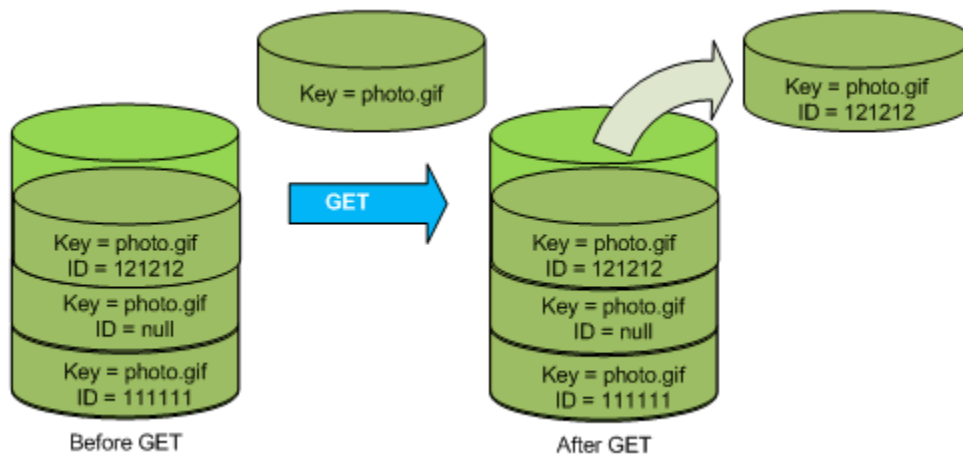
O comando a seguir retorna os metadados de todas as versões dos objetos em um bucket.

```
aws s3api list-object-versions --bucket DOC-EXAMPLE-BUCKET1
```

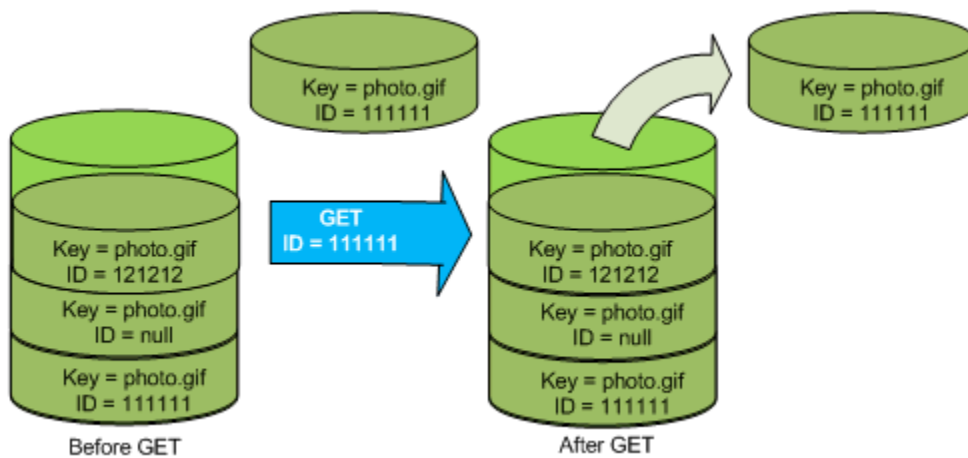
Para obter mais informações sobre `list-object-versions`, consulte [list-object-versions](#) na Referência de comandos de AWS CLI.

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.

Note

Para acessar versões de objetos anteriores a 300 versões, use a AWS CLI ou o URL do objeto.

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](#).

⚠ 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](#).

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](#).

Para obter exemplos de como listar a versão de objetos usando AWS SDKs para .NET e Rust, consulte [Listar a versão dos objetos em um bucket do Amazon S3](#).

Uso da API REST

Para recuperar uma versão específica do objeto

1. Defina `versionId` como o ID da versão do objeto que você deseja recuperar.
2. Envie uma solicitação `GET Object versionId`.

Example — 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”](#).

Para obter informações sobre como restaurar uma versão de objeto anterior, consulte [the section called “Restaurar versões anteriores”](#).

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: ef8yU9AS1ed40pIszj7UDNEHGran
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:

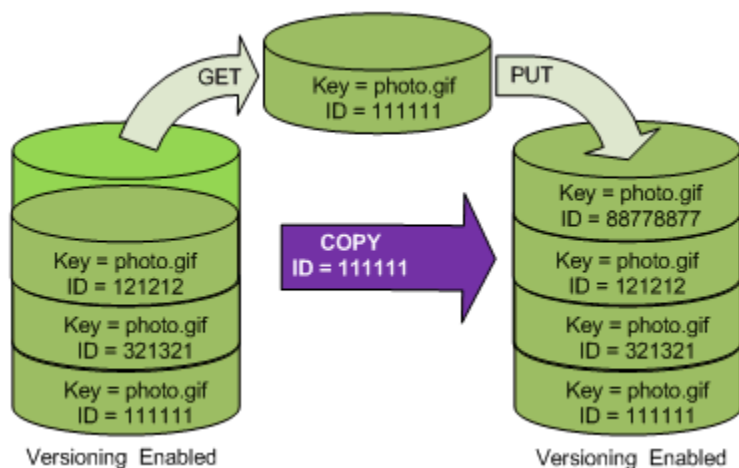
- Copie uma versão anterior do objeto para o mesmo bucket.

O objeto copiado torna-se a versão atual desse objeto e todas as versões são preservadas.

- Exclua permanentemente a versão atual do objeto.

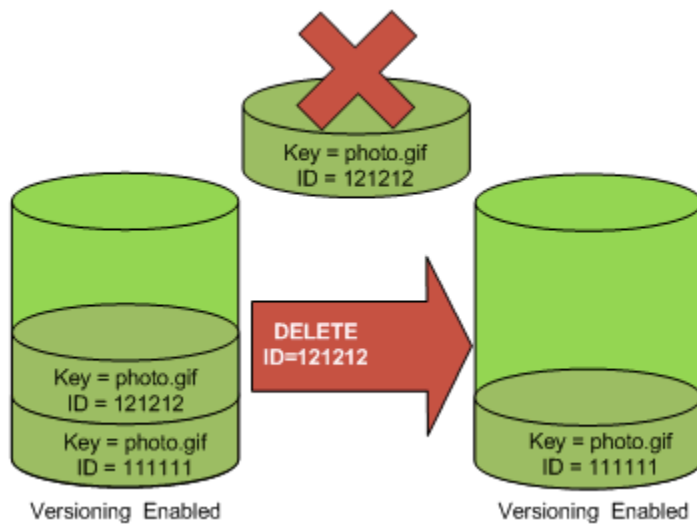
Quando você exclui a versão atual do objeto, torna a versão anterior a versão atual desse objeto.

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.

Note

Para restaurar versões de objetos em lotes, [use a operação CopyObject](#). A operação CopyObject copia cada objeto especificado no manifesto. No entanto, esteja ciente de que os objetos não são necessariamente copiados na mesma ordem em que estão listados no manifesto. Para buckets versionados, se preservar a ordem de versão atual/não atual for importante, você deve copiar todas as versões não atuais primeiro. Em seguida, após a conclusão do primeiro trabalho, copie as versões atuais em um trabalho subsequente.

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 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](#).

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](#).

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 \(Boto\)](#).

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 sobre o Ciclo de Vida do S3, consulte [Gerenciando seu ciclo de vida de armazenamento](#) e [Exemplos de configuração do S3 Lifecycle](#).

Para ver quantas versões de objetos atuais e desatualizadas seus buckets têm, você pode usar as métricas da Lente de Armazenamento do Amazon S3. A Lente de Armazenamento do S3 é um recurso de análise de armazenamento em nuvem que você pode usar para obter visibilidade em toda a organização sobre o uso e a atividade do armazenamento de objetos. Para obter mais informações, consulte [Usar a Lente de Armazenamento do S3 para otimizar os custos do armazenamento](#). Para obter uma lista completa de métricas, consulte o [Glossário de métricas da Lente de Armazenamento do S3](#).

Note

As taxas normais do Amazon S3 se aplicam a cada versão de um objeto armazenado e transferido, incluindo versões desatualizadas de objetos. Para obter mais informações, consulte [Preço do Amazon S3](#).

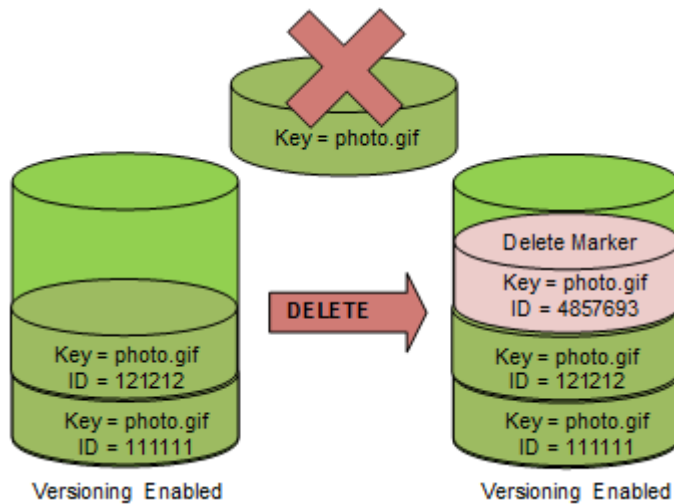
Excluir casos de uso de solicitação

Uma solicitação DELETE tem os seguintes casos de uso:

- Quando o versionamento está habilitado, um DELETE simples não pode excluir permanentemente um objeto. (Uma solicitação DELETE simples é uma solicitação que não especifica um ID de versão.) 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.

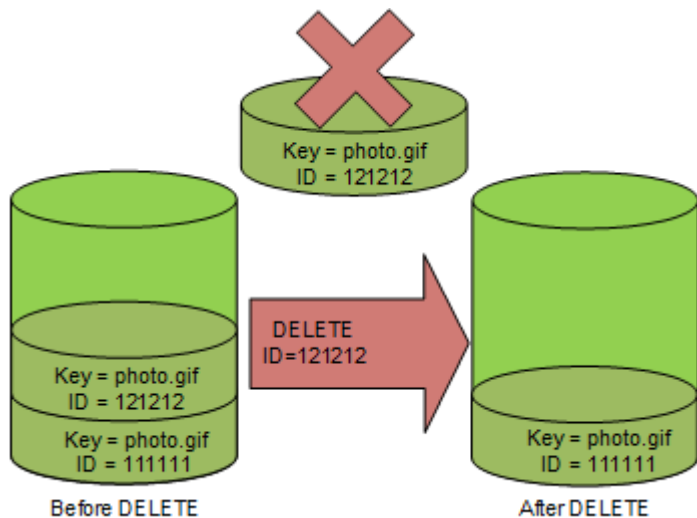
Quando você tenta um GET de um objeto cuja versão atual é um marcador de exclusão, o Amazon S3 se comporta como se o objeto tivesse sido excluído (mesmo que não tenha sido apagado) e retorna um erro 404. Para obter mais informações, consulte [Trabalhar com marcadores de exclusão](#).

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.
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 dos AWS SDKs

Para ver exemplos de exclusão de objetos usando os AWS SDKs para Java, o .NET e o PHP, consulte [Excluir objetos do Amazon S3](#). 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 \(Boto\)](#).

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 da API REST

Como 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 UI0RUnfnd89493jJFJ do `photo.gif`.

```
DELETE /photo.gif?versionId=UI0RUnfnd89493jJFJ 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
```

Usando a AWS CLI

O comando a seguir exclui um objeto chamado `test.txt` de um bucket chamado `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](#)
- [Como remover marcadores de exclusão para tornar uma versão mais antiga atual](#)
- [Excluir um objeto de um bucket com exclusão de MFA habilitada](#)

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 especificado em uma solicitação DELETE simples. Uma solicitação DELETE simples é uma solicitação que não especifica um ID de versão. 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 o objeto tivesse sido excluído. É possível usar a chamada de API DELETE do Amazon S3 em um marcador de exclusão. Para isso, será necessário fazer a solicitação DELETE usando um usuário ou um perfil do AWS Identity and Access Management (IAM) com as permissões adequadas.

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:

- Um marcador de exclusão não tem dados associados a ele.
- Um marcador de exclusão não está associado a um valor da lista de controle de acesso (ACL).
- Se você emitir uma solicitação GET para um marcador de exclusão, a solicitação GET não recuperará nada porque um marcador de exclusão não tem dados. Especificamente, quando a solicitação GET não especifica um `versionId`, você recebe um erro 404 (não encontrado).

Os marcadores de exclusão acumulam uma cobrança mínima 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 para o nome da chave adiciona de 1 a 4 bytes de armazenamento ao bucket para cada caractere no nome. Os marcadores de exclusão são armazenados na classe de armazenamento S3 Standard.

Se você quiser descobrir quantos marcadores de exclusão você tem e em qual classe de armazenamento eles estão armazenados, use a Lente de Armazenamento do Amazon S3. Para obter mais informações, consulte [Avaliação de sua atividade de armazenamento e uso com a Lente de Armazenamento do Amazon S3](#) e [Glossário de métricas de lente de armazenamento do Amazon S3](#).

Para obter mais informações sobre nomes de chave, consulte [Criar nomes de chave de objeto](#). Para obter mais informações sobre exclusão de um marcador de exclusão, consulte [Gerenciamento de marcadores de exclusão](#).

Somente o Amazon S3 pode criar e excluir um marcador de exclusão, e ele faz isso sempre que você envia uma solicitação `DeleteObject` em um objeto em um bucket com versionamento habilitado ou suspenso. O objeto especificado 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.

Ao obter um objeto sem especificar um `versionId` na solicitação, se a versão atual for um marcador de exclusão, o Amazon S3 responderá com o seguinte:

- Um erro 404 (não encontrado)
- Um cabeçalho de resposta, `x-amz-delete-marker: true`

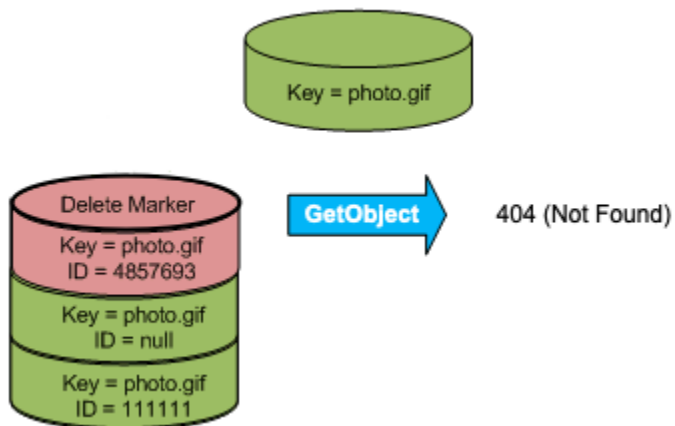
Ao obter um objeto especificando um `versionId` na solicitação, se a versão especificada for um marcador de exclusão, o Amazon S3 responderá com o seguinte:

- Um erro 405 (Método Não Permitido)
- Um cabeçalho de resposta, `x-amz-delete-marker: true`
- Um cabeçalho de resposta, `Last-Modified: timestamp` (somente ao usar as operações de API [HeadObject](#) ou [GetObject](#))

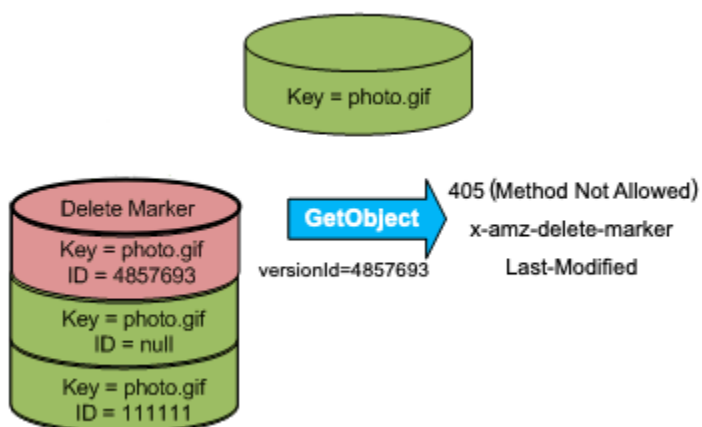
O cabeçalho de resposta `x-amz-delete-marker: true` mostra que o objeto acessado era um marcador de exclusão. Esse cabeçalho de resposta nunca retorna `false`, porque quando o valor é `false`, a versão atual ou especificada do objeto não é um marcador de exclusão.

O cabeçalho de resposta `Last-Modified` fornece a hora de criação dos marcadores de exclusão.

A figura a seguir mostra como uma chamada de API `GetObject` em um objeto, cuja versão atual é um marcador de exclusão, responde com um erro 404 (não encontrado), e o cabeçalho da resposta inclui `x-amz-delete-marker: true`.

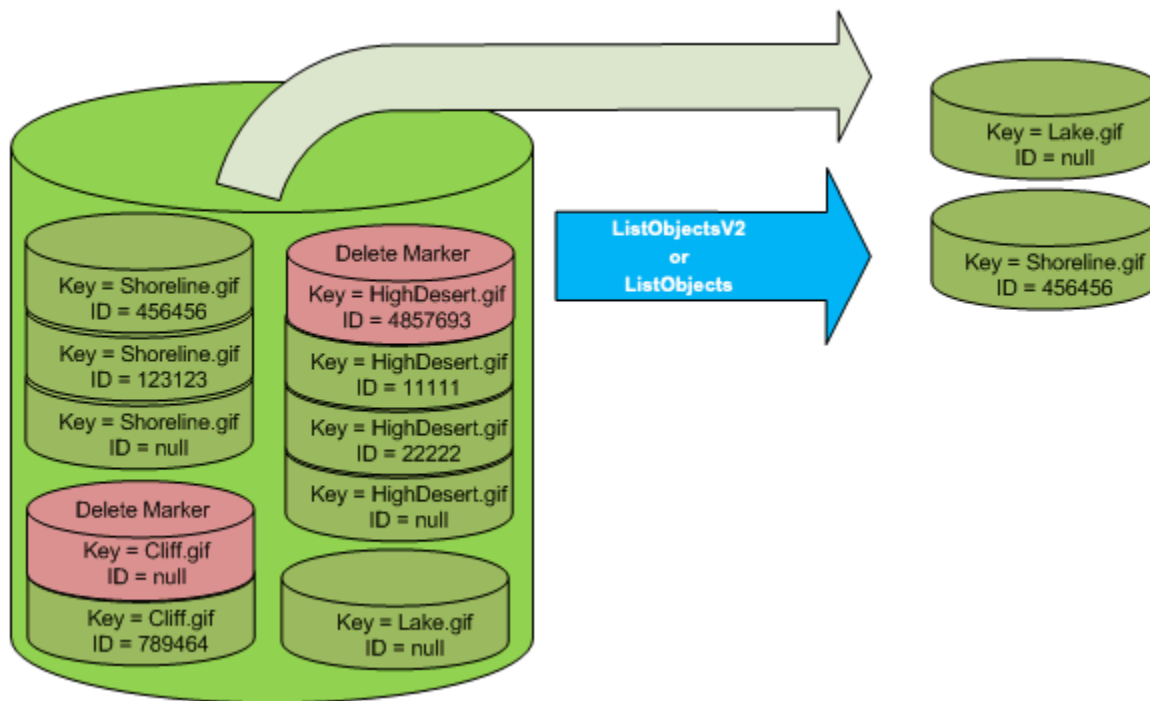


Se você fizer uma chamada `GetObject` em um objeto especificando um `versionId` na solicitação e se a versão especificada for um marcador de exclusão, o Amazon S3 responderá com um erro 405 (método não permitido) e os cabeçalhos de resposta incluirão `x-amz-delete-marker: true` e `Last-Modified: timestamp`.



O único modo de listar marcadores de exclusão (e outras versões de um objeto) é usando o `versions` sub-recurso em uma solicitação [ListObjectVersions](#). A figura a seguir mostra que uma

solicitação [ListObjectsV2](#) ou [ListObjects](#) não retorna objetos cuja versão atual é um marcador de exclusão.



Gerenciamento de marcadores de exclusão

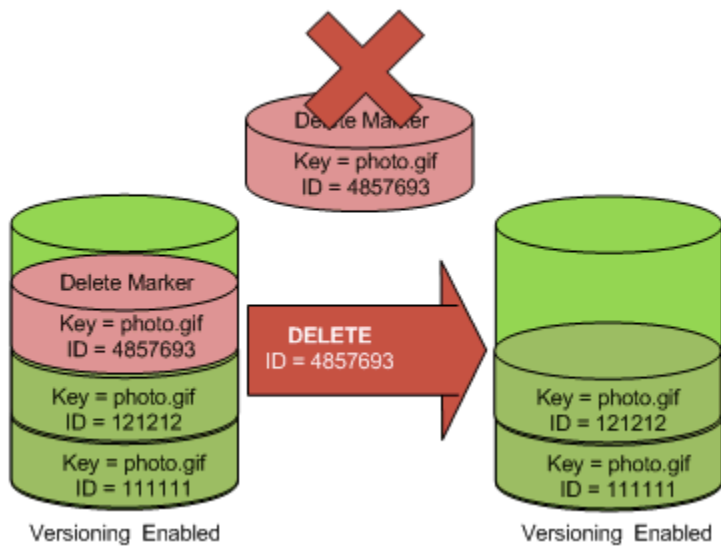
Como configurar o ciclo de vida para limpar marcadores de exclusão expirados automaticamente

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 configuração 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](#).

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](#).

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.

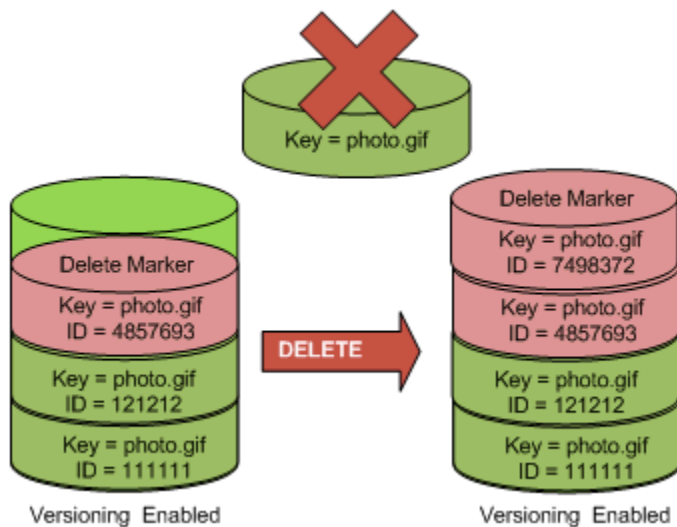


O efeito da remoção do marcador de exclusão é que uma simples solicitação GET não recuperará o ID da versão atual do objeto (121212).

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 dos AWS SDKs

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 \(Boto\)](#).

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=3HL4kqCxf3vjVBH40N1rjfkd 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](#).

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 [Gerenciamento de identidade e acesso para o Amazon S3](#).

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>
```

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, `DVBH40Nr8X8gUMLUo`.

```
GET /my-image.jpg?versionId=DVBH40Nr8X8gUMLUo&acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU
```

Para obter mais informações, consulte [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](#). Para obter mais informações sobre como recuperar as versões de um objeto, consulte [Recuperando versões de objeto de um bucket habilitado para versionamento](#).

Tópicos

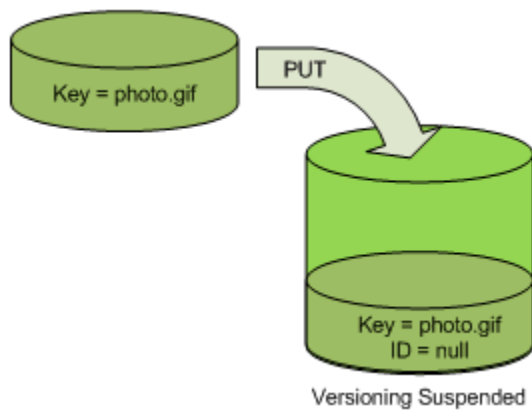
- [Adicionar objetos a buckets com versionamento suspenso](#)
- [Recuperar objetos de buckets com versionamento suspenso](#)
- [Excluir objetos de buckets com versionamento suspenso](#)

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 `CopyObject`) nesse bucket.

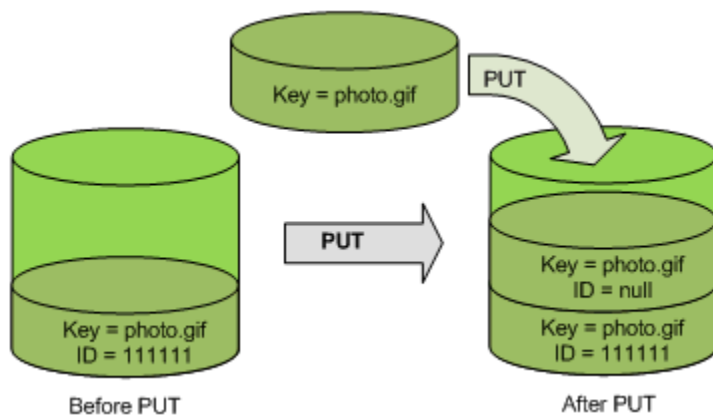
A figura a seguir mostra como o Amazon S3 adiciona um ID de versão `null` a cada objeto quando ele é adicionado a um bucket com versionamento suspenso.



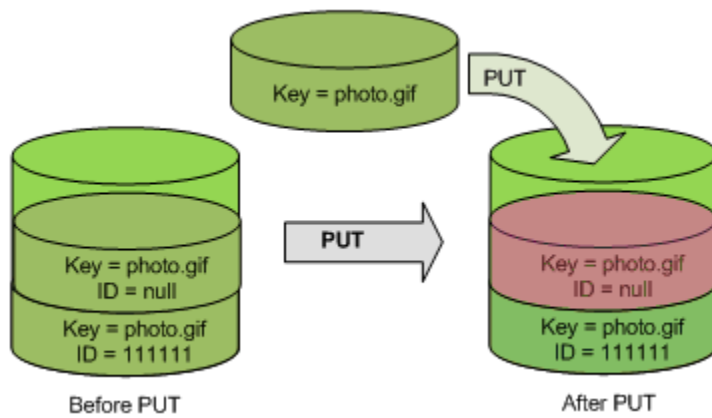
Se um versão nula já existir no bucket e você adicionar outro objeto com a mesma chave, o objeto adicionado substituirá a versão original nula.

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.



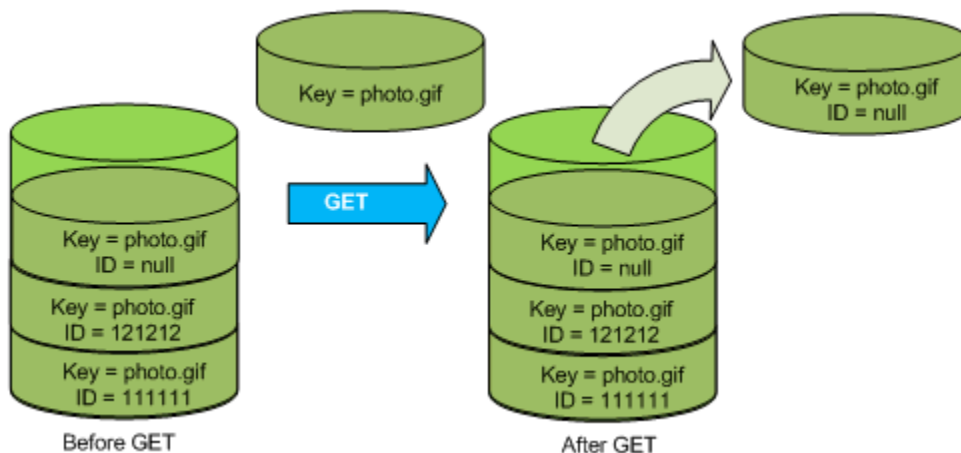
Se uma versão nula já existir em um bucket, a versão nula será substituída, como mostrado na figura a seguir.



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

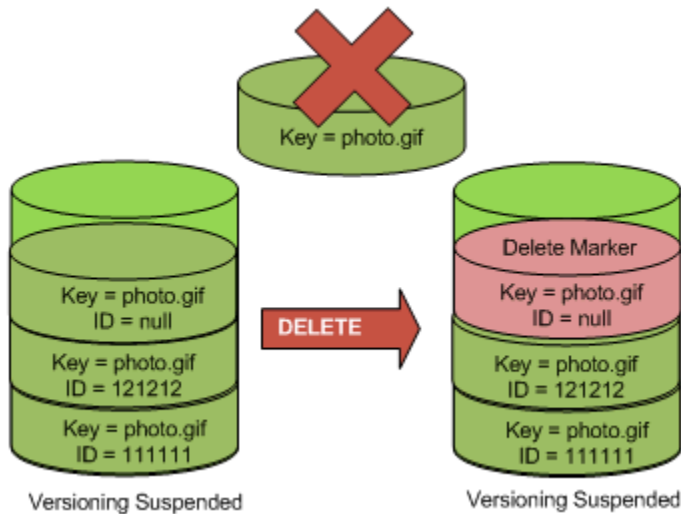
Você pode excluir objetos de buckets com versionamento suspenso para remover um objeto com um ID de versão nulo.

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.

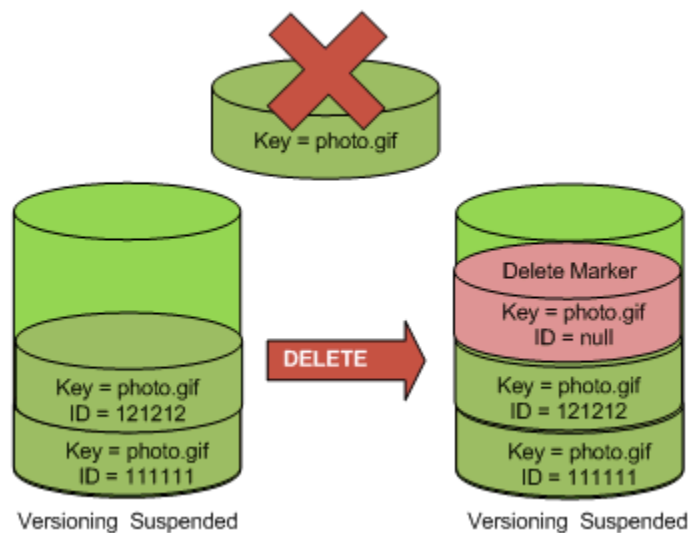
- Insere um marcador de exclusão no bucket.

A figura a seguir mostra como um simples DELETE remove uma versão nula. (Uma solicitação DELETE simples é uma solicitação que não especifica um ID de versão.) O Amazon S3 insere um marcador de exclusão em seu lugar com um ID de versão do null.



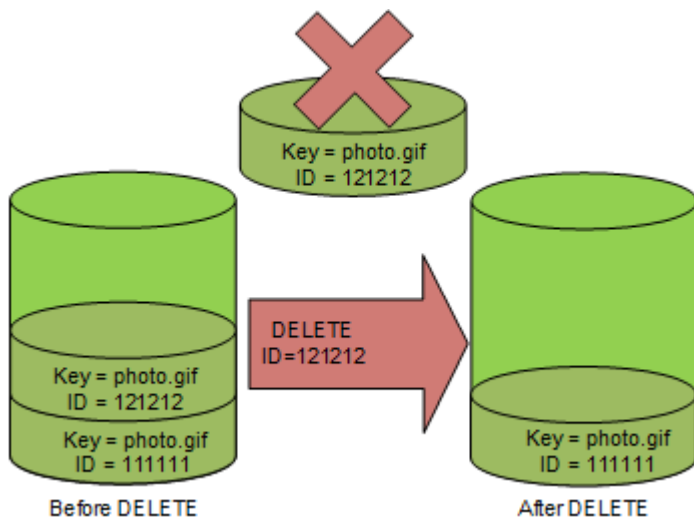
Lembre-se de que um marcador de exclusão não tem conteúdo, de modo que você perde o conteúdo da versão nula quando um marcador de exclusão a substitui.

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 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 [Creating Amazon S3 backups](#) (“Criar backups do Amazon S3”) no Guia do desenvolvedor do AWS Backup.

Trabalhando com objetos arquivados

Para reduzir os custos de armazenamento de objetos acessados com pouca frequência, você pode arquivar esses objetos. Quando você arquiva um objeto, ele é movido para um armazenamento de baixo custo, o que significa que não é possível acessá-lo em tempo real.

Embora os objetos arquivados não estejam acessíveis em tempo real, você pode restaurá-los em minutos ou horas, dependendo da classe de armazenamento. Você pode restaurar um objeto arquivado usando o console do Amazon S3, o Operações em Lote do S3, a API REST, os AWS SDKs e a AWS Command Line Interface (AWS CLI). Para obter instruções, consulte [Restaurar um objeto arquivado](#).

Os objetos do Amazon S3 nas seguintes classes ou níveis de armazenamento são arquivados e não podem ser acessados em tempo real:

- A classe de armazenamento S3 Glacier Flexible Retrieval
- A classe de armazenamento S3 Glacier Deep Archive
- O nível Acesso ao arquivamento do S3 Intelligent-Tiering
- O nível de acesso de arquivamento profundo do S3 Intelligent-Tiering

Para restaurar objetos arquivados, você deve fazer o seguinte:

- 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. Se for criada uma cópia temporária do objeto restaurado, a classe de armazenamento do objeto permanecerá a mesma. (Uma solicitação de operação de API [HeadObject](#) ou [GetObject](#) retornará S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive como a classe de armazenamento.)
- Para objetos nos níveis Acesso de arquivamento e Acesso de arquivamento profundo do S3 Intelligent-Tiering, você deve iniciar a solicitação de restauração e esperar até que o objeto seja movido para o nível Acesso frequente.

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](#). Para obter mais informações sobre o S3 Intelligent-Tiering, consulte [the section called “Como o S3 Intelligent-Tiering funciona”](#).

Restaurar objetos do S3 Glacier

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 validade de uma cópia restaurada reenviando uma solicitação de restauração. Nesse caso, o Amazon S3 atualiza o período de expiração relativo à hora atual.

Note

Ao restaurar um objeto arquivado do S3 Glacier Flexible Retrieval ou do S3 Glacier Deep Archive, você paga pelo objeto arquivado e pela cópia restaurada temporariamente. Para obter mais informações sobre preços, consulte [Preços do Amazon S3](#).

Restaurar objetos do S3 Intelligent-Tiering

Quando você está restaurando um objeto dos níveis S3 Intelligent-Tiering Archive Access ou S3 Intelligent-Tiering Deep Archive Access, o objeto volta para o nível S3 Intelligent-Tiering Frequent Access. Se o objeto não for acessado depois de 30 dias consecutivos, ele será movido automaticamente para o nível Acesso infrequente. Depois de um período mínimo de 90 dias

consecutivos sem acesso, o objeto é movido para o nível Acesso ao arquivamento do S3 Intelligent-Tiering. Se o objeto não for acessado depois de um período mínimo de 180 dias consecutivos, ele será movido para o nível Acesso profundo ao arquivamento.

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.

Usar o recurso Operações em Lote do S3 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 operação de 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.

Tempo de restauração

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, o Amazon S3 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, à 0h UTC, hora em que o Amazon S3 vai excluir a cópia do objeto.

O tempo necessário para concluir um trabalho de restauração depende da classe ou do nível de armazenamento de arquivo usado e da opção de recuperação especificada: expressa (disponível apenas para o S3 Glacier Flexible Retrieval e o S3 Intelligent-Tiering Archive Access), padrão ou em massa. Para ter mais informações, consulte [Opções de recuperação de arquivamento](#).

Você pode ser notificado quando sua restauração for concluída usando as Notificações de Eventos do Amazon S3. Para ter mais informações, consulte [Notificações de eventos do Amazon S3](#).

Tópicos

- [Opções de recuperação de arquivamento](#)

- [Restaurar um objeto arquivado](#)

Opções de recuperação de arquivamento

Veja a seguir as opções de recuperação disponíveis ao restaurar um objeto arquivado no Amazon S3:

- **Expressa:** acesse rapidamente os dados armazenados na classe de armazenamento S3 Glacier Flexible Retrieval ou no nível S3 Intelligent-Tiering Archive Access. É possível usar essa opção quando solicitações urgentes ocasionais para um subconjunto de arquivos for solicitado. 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.

Note

As recuperações expressas são um recurso premium e são cobradas de acordo com a taxa de solicitação e recuperação expressas.

Para receber informações sobre a definição de preço do Amazon S3, consulte [Definição de preço do Amazon S3](#).

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 ter mais informações, consulte [Capacidade provisionada](#).

- **Padrão:** acesse qualquer um dos objetos arquivados em algumas horas. Padrão é a opção assumida 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, essas recuperações terminam em até 12 horas para objetos armazenados na classe de armazenamento S3 Glacier Deep Archive ou no nível S3 Intelligent-Tiering Deep Archive Access. As recuperações padrão são gratuitas para objetos armazenados na classe S3 Intelligent-Tiering.

Note

- Para objetos armazenados na classe de armazenamento S3 Glacier Flexible Retrieval ou no nível S3 Intelligent-Tiering Archive Access, as recuperações padrão iniciadas por

meio da operação de restauração de Operações em Lote do S3 normalmente se iniciam em minutos e terminam em três a cinco horas.

- Para objetos na classe de armazenamento S3 Glacier Deep Archive ou no nível S3 Intelligent-Tiering Deep Archive Access, as recuperações padrão iniciadas por meio da operação de restauração de Operações em Lote normalmente se iniciam em nove horas e terminam em doze horas.

- Em massa: acesse os dados usando a opção de recuperação de menor custo do Amazon S3 Glacier. Com as recuperações em massa, você pode recuperar grandes quantidades de dados por um baixo custo, até mesmo petabytes.

Para objetos armazenados na classe de armazenamento S3 Glacier Flexible Retrieval ou no nível S3 Intelligent-Tiering Archive Access, as recuperações em massa normalmente terminam em cinco a doze horas. Para objetos armazenados na classe de armazenamento S3 Glacier Deep Archive ou no nível S3 Intelligent-Tiering Deep Archive Access, essas recuperações normalmente terminam em 48 horas.

As recuperações em massa são gratuitas para objetos armazenados na classe de armazenamento S3 Glacier Flexible Retrieval ou S3 Intelligent-Tiering.

A tabela a seguir resume as opções de recuperação de arquivos. Para obter mais informações sobre preços, consulte [Preços do Amazon S3](#).

Para fazer uma recuperação de Expedited, Standard ou Bulk, defina o elemento `Tier` da solicitação na solicitação da operação de API REST de [RestoreObject](#) que você deseja, ou o equivalente na AWS Command Line Interface (AWS CLI) ou em AWS SDKs. Se você adquiriu a capacidade provisionada, todas as recuperações expressas serão automaticamente fornecidas por meio de sua capacidade provisionada.

Capacidade provisionada

A capacidade provisionada ajuda a garantir que 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 megabytes por segundo (MB/s) de throughput de recuperação.

Se a workload exigir acesso altamente confiável e previsível a um subconjunto de dados em minutos, considere 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 circunstâncias, recomendamos que você compre a capacidade de recuperação provisionada.

As unidades de capacidade provisionada são alocadas a uma Conta da AWS. Portanto, o solicitante da recuperação rápida de dados deve comprar a unidade de capacidade provisionada, não o proprietário do bucket.

É possível comprar a capacidade provisionada usando o console do Amazon S3, o console do Amazon S3 Glacier, a operação [Purchase Provisioned Capacity](#) da API REST, 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](#).

Taxas de solicitações de iniciação de restauração do S3 Glacier

Quando você inicia solicitações de restauração de objetos que estão armazenados nas classes de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, uma cota de solicitações de recuperação é aplicada para sua Conta da AWS. O S3 Glacier comporta solicitações de restauração em uma taxa de até 1.000 transações por segundo. Se essa taxa for excedida, as solicitações consideradas válidas em outros aspectos terão a utilização controlada ou serão rejeitadas, e o Amazon S3 retornará um erro `ThrottlingException`.

Opcionalmente, você também pode usar o recurso Operações em Lote do S3 para recuperar um grande número de objetos armazenados nas classes S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive com uma única solicitação. Para ter mais informações, consulte [Executar operações em lote de grande escala em objetos do Amazon S3](#).

Restaurar um objeto arquivado

Os objetos do Amazon S3 nas seguintes classes ou níveis de armazenamento são arquivados e não podem ser acessados em tempo real:

- A classe de armazenamento S3 Glacier Flexible Retrieval
- A classe de armazenamento S3 Glacier Deep Archive
- O nível Acesso ao arquivamento do S3 Intelligent-Tiering
- O nível de acesso de arquivamento profundo do S3 Intelligent-Tiering

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 para uma duração especificada (número de dias). Se você quiser uma cópia permanente do objeto, restaure o objeto e crie uma cópia dele no bucket do Amazon S3. O console do Amazon S3 não comporta a cópia de objetos restaurados. Para esse tipo de operação de cópia, use a AWS Command Line Interface (AWS CLI), AWS SDKs ou a API REST. A menos que você faça uma cópia e altere a classe de armazenamento, o objeto ainda será armazenado nas classes de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive. Para receber informações sobre como usar essas classes de armazenamento, consulte [Classes de armazenamento para arquivamento de objetos](#).

Para acessar objetos nos níveis S3 Intelligent-Tiering Archive Access e Deep Archive Access, você deve iniciar a solicitação de restauração e esperar até que o objeto seja movido para o nível Frequent Access para acessá-lo. 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. Para receber informações sobre como usar essas classes de armazenamento, consulte [Classe de armazenamento para otimizar automaticamente dados com padrões de acesso alterados ou desconhecidos](#).

Para receber informações gerais sobre objetos arquivados, consulte [Trabalhando com objetos arquivados](#).

Note

- Ao restaurar um objeto arquivado das classes de armazenamento S3 Glacier Flexible Retrieval ou do S3 Glacier Deep Archive, você paga pelo objeto arquivado e pela cópia restaurada temporariamente.
- Quando você restaura um objeto do S3 Intelligent-Tiering, não há cobranças para recuperações padrão ou em massa.
- As solicitações de restauração subsequentes chamadas em objetos arquivados que já estão sendo restaurados são cobradas como uma solicitação GET. Para obter mais informações sobre preços, consulte [Preços do Amazon S3](#).

Restaurar um objeto arquivado

Você pode restaurar um objeto arquivado usando o console do Amazon S3, a API REST do Amazon S3, os AWS SDKs, a AWS Command Line Interface (AWS CLI) ou o Operações em Lote do S3.

Uso do console do S3

Restaurar objetos usando o console do Amazon S3

Use o procedimento a seguir para restaurar um objeto que foi arquivado na classe de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive ou no nível de armazenamento S3 Intelligent-Tiering Archive Access ou Deep Archive Access.

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. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, escolha o nome do bucket que contém os objetos que você deseja restaurar.
4. Na lista Objects (Objetos), selecione o objeto ou os objetos que você deseja restaurar, selecione Actions (Ações) e escolha Initiate restore (Iniciar restauração).
5. Se você estiver restaurando do S3 Glacier Flexible Retrieval ou do S3 Glacier Deep Archive, insira o número de dias em que deseja que seus dados arquivados fiquem acessíveis na caixa Número de dias em que a cópia restaurada fica disponível.
6. Em Nível de recuperação, siga um destes procedimentos:
 - Selecione Recuperação em massa ou Recuperação padrão e, depois, escolha Iniciar restauração.
 - Escolha Expedited retrieval (Recuperação acelerada) (disponível somente para o S3 Glacier Flexible Retrieval ou S3 Intelligent-Tiering Archive Access). Se você estiver restaurando um objeto no S3 Glacier Flexible Retrieval, poderá escolher se deseja comprar capacidade provisionada para recuperação expressa. Se quiser comprar capacidade provisionada, vá para a próxima etapa. Caso contrário, selecione Iniciar restauração.
7. (Opcional) Se você estiver restaurando um objeto no S3 Glacier Flexible Retrieval e escolher Recuperação expressa, poderá escolher se deseja comprar capacidade provisionada. A capacidade provisionada só está disponível para objetos no S3 Glacier Flexible Retrieval. Se

Note

Os objetos dos níveis S3 Intelligent-Tiering Archive Access e Deep Archive Access são restaurados automaticamente para o nível Frequent Access.

Se você tiver a capacidade provisionada, escolha Iniciar restauração para iniciar uma recuperação provisionada.

Se você tiver a capacidade provisionada, todas as recuperações expressas serão atendidas pela capacidade provisionada. Para ter mais informações, consulte [Capacidade provisionada](#).

- Se você não tiver a capacidade provisionada e não desejar comprá-la, selecione Iniciar restauração.
- Se você não tiver capacidade provisionada mas quiser comprar unidades de capacidade provisionada (PCUs), selecione Comprar PCUs. Na caixa de diálogo Comprar PCUs, escolha quantas PCUs você deseja comprar, confirme a compra e selecione Comprar PCUs. Ao receber a mensagem Compra realizada, selecione Iniciar restauração para iniciar a recuperação provisionada.

Usando a AWS CLI

Restaurar objetos do S3 Glacier Flexible Retrieval ou do S3 Glacier Deep Archive

O exemplo a seguir usa o comando `restore-object` para restaurar o objeto `dir1/example.obj` no bucket `DOC-EXAMPLE-BUCKET` por 25 dias.

```
aws s3api restore-object --bucket DOC-EXAMPLE-BUCKET --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"}
```

Restaurar objetos do S3 Intelligent-Tiering Archive Access e Deep Archive Access

O exemplo a seguir usa o comando `restore-object` para restaurar o objeto `dir1/example.obj` no bucket `DOC-EXAMPLE-BUCKET` no nível Frequent Access.

```
aws s3api restore-object --bucket DOC-EXAMPLE-BUCKET --key dir1/example.obj --restore-request '{}'
```

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.

Monitorar o status da restauração

Para monitorar o status da solicitação `restore-object`, use o seguinte comando `head-object`:

```
aws s3api head-object --bucket DOC-EXAMPLE-BUCKET --key dir1/example.obj
```

Para obter mais informações, consulte [restore-object](#) na Referência de comandos da AWS CLI.

Uso da API REST

O Amazon S3 fornece uma operação de API para que você inicie a restauração de um objeto arquivado. Para obter mais informações, consulte [RestoreObject](#) na Referência da API do Amazon Simple Storage Service.

Uso dos AWS SDKs

Para ver exemplos de como restaurar objetos arquivados no S3 Glacier Flexible Retrieval ou no S3 Glacier Deep Archive com os AWS SDKs, consulte [Usar RestoreObject com um SDK da AWS ou uma ferramenta de linha de comandos](#).

Usar o Operações em Lote do S3

Para restaurar mais de um objeto arquivado com uma única solicitação, você pode usar o 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 operação de 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.

Para criar um trabalho de operações em lote, você deve ter um manifesto que contenha somente os objetos a serem restaurados. Você pode criar um manifesto usando o Inventário S3 ou fornecer um arquivo CSV com as informações necessárias. Para ter mais informações, consulte [the section called “Especificar um manifesto”](#).

Antes de criar e executar trabalhos de operações em lote no S3, você deve conceder permissões ao Amazon S3 para realizar essas operações em seu nome. Para ver as permissões necessárias, consulte [the section called “Conceder permissões”](#).

Note

Os trabalhos de operações em lote podem operar em objetos das classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive ou em objetos dos níveis S3 Intelligent-Tiering Archive Access e Deep Archive Access. As operações em lote não podem operar nos dois tipos de objetos arquivados no mesmo trabalho. Para restaurar objetos de ambos os tipos, você deve criar trabalhos de Operações em lote separados.

Para receber mais informações sobre o uso de operações em lote para restauração de objetos de arquivo, consulte [the section called “Restaurar objetos”](#).

Como criar um trabalho de operações em lote de iniciar restauração no 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 Criar trabalho.
4. Para Região da AWS, escolha a região onde você deseja criar o trabalho.
5. Em Formato do manifesto, selecione o tipo de manifesto a ser usado.
 - Se você escolher Relatório de inventário do S3, insira o caminho para o objeto `manifest.json` que o Amazon S3 gerou como parte do relatório de inventário em formato CSV. Se você quiser usar uma versão de manifesto que não seja a mais recente, insira o ID da versão do objeto `manifest.json`.
 - 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. Se você quiser usar outra versão que não seja a mais recente, poderá incluir o ID da versão do objeto de manifesto.
6. Escolha Próximo.
7. Na seção Operação, escolha Restaurar.
8. Na seção Restaurar, em Restaurar origem, selecione Glacier Flexible Retrieval ou Glacier Deep Archive ou Nível Intelligent-Tiering Archive Access ou Deep Archive Access.

Se você escolheu Glacier Flexible Retrieval ou Glacier Deep Archive, insira um número para Número de dias em que a cópia restaurada fica disponível.

Em Nível de recuperação, selecione o nível que você deseja usar.

9. Escolha Próximo.

10.

Na página Configurar opções adicionais, preencha as seguintes seções:

- Na seção Opções adicionais, forneça uma descrição para o trabalho e especifique um número de prioridade para ele. Números maiores indicam uma prioridade mais alta. Para ter mais informações, consulte [the section called “Atribuir prioridade aos trabalhos”](#).
- Na seção Relatório de conclusão, selecione se as operações em lote devem criar um relatório de conclusão. Para receber mais informações sobre relatórios de conclusão, consulte [the section called “Relatórios de conclusão”](#).
- Na seção Permissões, você deve conceder permissões ao Amazon S3 para realizar operações em lote em seu nome. Para ver as permissões necessárias, consulte [the section called “Conceder permissões”](#).
- (Opcional) Na seção Tags de trabalho, adicione tags em pares de chave-valor. Para ter mais informações, consulte [the section called “Usar tags”](#).

Quando terminar, escolha Próximo.

11. Na página Review (Examinar), verifique as configurações. Se precisar fazer alterações, escolha Previous (Anterior). Caso contrário, selecione Criar trabalho.

Para receber mais informações sobre operações em lote, consulte [Restaurar objetos com operações em lote](#) e [Criar um trabalho de operações em lote do S3](#).

Verificar o status de restauração e a data de expiração

Você pode verificar o status de uma solicitação de restauração ou a data de expiração usando o console do Amazon S3, as Notificações de eventos do Amazon S3, a AWS CLI ou a API REST do Amazon S3.

Note

Os objetos restaurados das classes de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive são armazenados apenas pelo número de dias que você especificar.

Os procedimentos abaixo retornarão a data de expiração dessas cópias.

Os objetos restaurados dos níveis de armazenamento S3 Intelligent-Tiering Archive Access e Deep Archive Access não têm datas de expiração e, em vez disso, são movidos para o nível Frequent Access.

Uso do console do S3

Como conferir o status de restauração e a data de validade de um objeto no console do Amazon S3

1. 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, selecione o nome do bucket com os objetos que você está restaurando.
4. Na lista Objetos, selecione o objeto que você está restaurando. A página de detalhes do objeto é exibida.
 - Se a restauração não estiver concluída, você verá uma seção que indica Restauração em andamento na parte superior da página.
 - Se a restauração estiver concluída, você verá uma seção que indica Restauração concluída na parte superior da página. Se você estiver restaurando por meio do S3 Glacier Flexible Retrieval ou do S3 Glacier Deep Archive, essa seção também exibirá a Data de expiração da restauração. O Amazon S3 removerá a cópia restaurada do objeto arquivado nessa data.

Usar Notificações de eventos do Amazon S3

Você pode receber notificações sobre a conclusão da restauração de um objeto usando a ação `s3:ObjectRestore:Completed` com o recurso Notificações de eventos do Amazon S3. Para obter mais informações sobre como habilitar notificações de eventos, consulte [Usar o Amazon SQS, o Amazon SNS e o AWS Lambda](#). Para obter mais informações sobre os vários tipos de evento `ObjectRestore`, consulte [the section called “Tipos de evento compatíveis com SQS, SNS e Lambda”](#).

Usando a AWS CLI

Confira o status de restauração e a data de validade de um objeto com a AWS CLI.

O exemplo a seguir usa o comando `head-object` para visualizar metadados referentes ao objeto `dir1/example.obj` no bucket `DOC-EXAMPLE-BUCKET`. Quando você executa esse comando em um objeto que está sendo restaurado, o Amazon S3 retorna se a restauração está em andamento e (se aplicável) a data de validade.

```
aws s3api head-object --bucket DOC-EXAMPLE-BUCKET --key dir1/example.obj
```

Saída esperada (restauração em andamento):

```
{
  "Restore": "ongoing-request=\"true\"",
  "LastModified": "2020-06-16T21:55:22+00:00",
  "ContentLength": 405,
  "ETag": "\"b662d79adeb7c8d787ea7eafb9ef6207\"",
  "VersionId": "wbYaE2vt0V0iIBXr0qGAJt3fP1cHB8Wi",
  "ContentType": "binary/octet-stream",
  "ServerSideEncryption": "AES256",
  "Metadata": {},
  "StorageClass": "GLACIER"
}
```

Saída esperada (restauração concluída):

```
{
  "Restore": "ongoing-request=\"false\", expiry-date=\"Wed, 12 Aug 2020 00:00:00 GMT\"",
  "LastModified": "2020-06-16T21:55:22+00:00",
  "ContentLength": 405,
  "ETag": "\"b662d79adeb7c8d787ea7eafb9ef6207\"",
  "VersionId": "wbYaE2vt0V0iIBXr0qGAJt3fP1cHB8Wi",
  "ContentType": "binary/octet-stream",
  "ServerSideEncryption": "AES256",
  "Metadata": {},
  "StorageClass": "GLACIER"
}
```

Para obter mais informações sobre `head-object`, consulte [head-object](#) na Referência de comandos da AWS CLI.

Uso da API REST

O Amazon S3 fornece uma operação de API para você recuperar metadados do objeto. Para conferir o status da restauração e a data de validade de um objeto arquivado usando a API REST, consulte [HeadObject](#) na Referência da API do Amazon Simple Storage Service.

Atualizar a velocidade de uma restauração em andamento

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. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, escolha o nome do bucket que contém os objetos que você deseja restaurar.
4. Na lista Objetos, selecione o objeto que você está restaurando. A página de detalhes do objeto é exibida. Na página de detalhes do objeto, selecione Fazer upgrade do nível de recuperação. Para obter informações sobre como verificar o status de restauração de um objeto, consulte [Verificar o status de restauração e a data de expiração](#).
5. Escolha o nível para o qual você deseja atualizar e selecione Iniciar restauração.

Usar o bloqueio de objetos do S3

A funcionalidade Bloqueio de Objetos do S3 pode ajudar a evitar que os objetos do Amazon S3 sejam excluídos ou substituídos por um período fixo ou indefinidamente. A funcionalidade Bloqueio de Objetos usa um modelo de gravação única e várias leituras (WORM). Você pode usar a funcionalidade Bloqueio de Objetos para ajudar a atender aos requisitos regulamentares que exigem armazenamento WORM ou para adicionar outra camada de proteção contra alterações ou exclusão de objetos.

Note

A funcionalidade Bloqueio de Objetos do S3 foi avaliada pela Cohasset Associates para uso em ambientes sujeitos aos regulamentos SEC 17a-4, CFTC e FINRA. Para obter mais informações sobre como a funcionalidade Bloqueio de Objetos está relacionada a essas regulamentações, consulte o documento [Cohasset Associates Compliance Assessment](#).

O bloqueio de objetos fornece duas maneiras de gerenciar a retenção de objetos: períodos de retenção e retenções legais. Uma versão do objeto pode ter um período de retenção, uma retenção jurídica ou ambos.

- **Período de retenção:** um período de retenção especifica um período fixo durante o qual um objeto permanece bloqueado. Você pode definir um período de retenção exclusivo para objetos individuais. Além disso, você pode definir um período de retenção padrão em um bucket do S3. Também é possível restringir os períodos mínimo e máximo de retenção permitidos com a chave de condição `s3:object-lock-remaining-retention-days` na política do bucket. Isso ajuda você a estabelecer uma faixa de períodos de retenção e a restringir períodos de retenção que podem ser menores ou maiores do que essa faixa.
- **Retenção jurídica:** uma retenção jurídica 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 jurídicas independem dos períodos de retenção e são colocadas em versões de objetos individuais.

A funcionalidade Bloqueio de Objetos só funciona somente em buckets com a funcionalidade Versionamento do S3 habilitada. 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 jurídica em um objeto só protege a versão especificada na solicitação. Os períodos de retenção e as retenções jurídicas não impedem a criação de versões do objeto nem a adição de marcadores de exclusão no objeto. Para obter informações sobre o Versionamento do S3, consulte [Usando o versionamento em buckets do S3](#).

Se você colocar um objeto em um bucket que já contenha um objeto protegido com o mesmo nome de chave de objeto, o Amazon S3 criará outra versão desse objeto. A versão protegida existente do objeto permanece bloqueada de acordo com a configuração da retenção.

Como o bloqueio de objetos do S3 funciona

Tópicos

- [Períodos de retenção](#)
- [Modos de retenção](#)
- [Retenções legais](#)
- [Práticas recomendadas para usar a funcionalidade Bloqueio de Objetos do S3](#)
- [Permissões obrigatórias](#)

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.

Você pode colocar um período de retenção explicitamente em uma versão de objeto individual ou nas propriedades de um bucket para que ele se aplique automaticamente a todos os objetos no 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 essa data nos metadados da versão do objeto.

Você também pode definir um período de retenção nas propriedades de um bucket. Ao definir um período de retenção em um bucket, especifique uma duração, em dias ou anos, para proteger cada versão do objeto colocada no bucket. Quando você coloca um objeto no bucket, o Amazon S3 calcula um Prazo de retenção para a versão do objeto, adicionando a duração especificada ao timestamp da criação da versão do objeto. A versão do objeto é protegida exatamente como se você tivesse colocado explicitamente um bloqueio individual com esse período de retenção na versão do objeto.

Note

Quando você usa PUT em uma versão de objeto que tem um modo e um período de retenção individual explícitos em um bucket, as configurações individuais da funcionalidade Bloqueio de Objetos da versão do objeto substituem qualquer configuração de retenção de propriedades do bucket.

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, a solicitação PUT será bem-sucedida e o Amazon S3 criará outra 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.

Depois que você tiver aplicado uma configuração de retenção a uma versão do objeto, poderá estender o período de retenção. Para fazer isso, envie uma nova solicitação da funcionalidade Bloqueio de Objetos para a versão do objeto com um Prazo de retenção que seja posterior ao configurado 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 estender um período de retenção para uma versão do objeto. Para definir um período de retenção, você deve ter a permissão `s3:PutObjectRetention`.

Ao definir um período de retenção em um objeto ou bucket do S3, é necessário selecionar um dos dois modos de retenção: conformidade ou governança.

Modos de retenção

A funcionalidade Bloqueio de Objetos do S3 oferece dois modos de retenção que aplicam níveis diferentes de proteção aos objetos:

- Modo de conformidade
- Modo de governança

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 única maneira de excluir um objeto no modo de conformidade antes da data de retenção é excluir a Conta da AWS associada.

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 objetos, caso necessário. Também é possível usar o modo de governança para testar as configurações do período de retenção antes de criar um período de retenção do modo de conformidade.

Para substituir ou remover as configurações de retenção do modo de governança, você precisa 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 de governança e tiver a permissão `s3:BypassGovernanceRetention`, a operação será bem-sucedida.

Retenções legais

Com a funcionalidade Bloqueio de Objetos, você também pode colocar uma retenção jurídica 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 jurídica não tem uma quantidade fixa de tempo associada e permanecerá em vigor até ser removida. As retenções legais podem ser feitas e removidas livremente por qualquer usuário com a permissão `s3:PutObjectLegalHold`.

As retenções legais independem dos períodos de retenção. 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 jurídica em uma versão do objeto e que 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 jurídica 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.

Práticas recomendadas para usar a funcionalidade Bloqueio de Objetos do S3

Considere usar o Modo de governança se quiser proteger objetos de serem excluídos pela maioria dos usuários durante um período de retenção predefinido, mas ao mesmo tempo quiser que alguns usuários com permissões especiais tenham a flexibilidade de alterar as configurações de retenção ou excluir os objetos.

Considere usar o Modo de conformidade se você nunca quiser que nenhum usuário, incluindo o usuário raiz em sua Conta da AWS, possa excluir os objetos durante um período de retenção predefinido. Você pode usar esse modo caso precise armazenar dados de conformidade.

Você pode usar a Retenção jurídica quando não tiver certeza de quanto tempo deseja que os objetos permaneçam imutáveis. Isso pode ser devido a uma futura auditoria externa dos dados, em que você deseja manter os objetos imutáveis até que a auditoria seja concluída. Como alternativa, você pode ter um projeto em andamento que utiliza um conjunto de dados que deseja manter imutável até que o projeto seja concluído.

Permissões obrigatórias

As operações do bloqueio de objetos exigem permissões específicas. Dependendo da operação exata que você está tentando realizar, algumas das seguintes permissões poderão ser necessárias:

- `s3:BypassGovernanceRetention`
- `s3:GetBucketObjectLockConfiguration`
- `s3:GetObjectLegalHold`
- `s3:GetObjectRetention`
- `s3:PutBucketObjectLockConfiguration`
- `s3:PutObjectLegalHold`
- `s3:PutObjectRetention`

Consulte uma lista completa de permissões do Amazon S3 com descrições em [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço.

Para obter informações sobre como usar condições com permissões, consulte [Exemplos de políticas de bucket usando chaves de condição](#).

Considerações sobre a funcionalidade Bloqueio de Objetos

A funcionalidade Bloqueio de Objetos do Amazon S3 pode ajudar a evitar que os objetos sejam excluídos ou substituídos por um período fixo ou indefinidamente.

É possível usar o console do Amazon S3, a AWS Command Line Interface (AWS CLI), os AWS SDKs ou a API REST do Amazon S3 para visualizar ou definir informações da funcionalidade Bloqueio de Objetos. Para obter informações gerais sobre os recursos da funcionalidade Bloqueio de Objetos do S3, consulte [Usar o bloqueio de objetos do S3](#).

Important

- Depois de habilitar a funcionalidade Bloqueio de Objetos em um bucket, não será possível desabilitá-la nem suspender o versionamento nesse bucket.
- Os buckets do S3 com funcionalidade Bloqueio de Objetos não podem ser usados como buckets de destino para logs de acesso ao servidor. Para ter mais informações, consulte [the section called “Registrando acesso ao servidor”](#).

Tópicos

- [Permissões para visualizar as informações de bloqueio](#)
- [Ignorar modo de governança](#)
- [Usar a funcionalidade Bloqueio de Objetos com a funcionalidade Replicação do S3](#)
- [Usar a funcionalidade Bloqueio de Objetos com a funcionalidade Inventário Amazon S3](#)
- [Gerenciar políticas da funcionalidade Ciclo de Vida do S3 com a funcionalidade Bloqueio de Objetos](#)
- [Gerenciar marcadores de exclusão com a funcionalidade Bloqueio de Objetos](#)
- [Usar a funcionalidade Lente de Armazenamento do S3 com a funcionalidade Bloqueio de Objetos](#)
- [Fazer upload de objetos em um bucket habilitado para a funcionalidade Bloqueio de Objetos](#)
- [Configurar eventos e notificações](#)
- [Definir limites para os períodos de retenção com uma política de bucket](#)

Permissões para visualizar as informações de bloqueio

É possível visualizar de forma programática o status da funcionalidade Bloqueio de Objetos de uma versão de objeto do Amazon S3 usando as operações [HeadObject](#) ou [GetObject](#). Ambas as operações retornam o modo de retenção, a data de retenção e o status de retenção jurídica para a versão do objeto especificada. Além disso, é possível visualizar o status da funcionalidade Bloqueio de Objetos para vários objetos em um bucket do S3 usando a funcionalidade Inventário do S3.

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`. Para visualizar a configuração de retenção padrão de um bucket, é necessário ter a permissão `s3:GetBucketObjectLockConfiguration`.

Caso você faça uma solicitação de uma configuração da funcionalidade Bloqueio de Objetos em um bucket que não tenha a funcionalidade Bloqueio de Objetos do S3 habilitada, o Amazon S3 retornará um erro.

Ignorar modo de governança

Se você tiver a permissão `s3:BypassGovernanceRetention`, poderá realizar operações em versões de objeto bloqueadas no modo de governança como se elas estivessem desprotegidas. Essas operações incluem a exclusão de uma versão do objeto, a redução do período de retenção ou a remoção do período de retenção da funcionalidade Bloqueio de Objetos com a colocação de uma nova solicitação `PutObjectRetention` 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` na solicitação da operação de API `PutObjectRetention` ou use o parâmetro equivalente com solicitações feitas por meio da AWS CLI ou dos AWS SDKs. O console do S3 aplicará automaticamente esse cabeçalho para solicitações feitas por meio do console do S3 se você tiver a permissão `s3:BypassGovernanceRetention`.

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 jurídica habilitada, essa retenção permanecerá em vigor e evitará que solicitações substituam ou excluam a versão do objeto.

Usar a funcionalidade Bloqueio de Objetos com a funcionalidade Replicação do S3

Você pode usar a funcionalidade Bloqueio de Objetos com a funcionalidade Replicação do S3 para habilitar a cópia automática e assíncrona de objetos bloqueados e seus metadados de retenção em buckets do S3. Isso significa que, para objetos replicados, o Amazon S3 usa a configuração de bloqueio de objetos do bucket de origem. Em outras palavras, se o bucket de origem tiver o recurso Bloqueio de Objetos habilitado, os buckets de destino também deverão ter esse recurso habilitado. Se um objeto for carregado diretamente no bucket de destino (fora da replicação do S3), ele usará o Bloqueio de Objetos definido no bucket de destino. Ao usar a replicação, os objetos em um bucket de origem são replicados para um ou mais buckets de destino.

Para configurar a replicação em um bucket com a funcionalidade Bloqueio de Objetos habilitada, você pode usar o console do S3, a AWS CLI, a API REST do Amazon S3 ou os AWS SDKs.

Note

Para usar a funcionalidade Bloqueio de Objetos com replicação, é necessário conceder duas permissões adicionais no bucket de origem do S3 no perfil do AWS Identity and Access Management (IAM) usado para configurar a replicação. As duas permissões adicionais são `s3:GetObjectRetention` e `s3:GetObjectLegalHold`. Se o perfil tiver uma instrução de permissão `s3:Get*`, essa instrução satisfará o requisito. Para ter mais informações, consulte [Configuração de permissões](#).

Para obter informações gerais sobre a funcionalidade Replicação do S3, consulte [Replicação de objetos](#).

Para obter exemplos de configuração da funcionalidade Replicação do S3, consulte [Demonstrações: exemplos para configurar a replicação](#).

Usar a funcionalidade Bloqueio de Objetos com a funcionalidade Inventário Amazon S3

Você pode configurar a funcionalidade Inventário Amazon S3 para criar listas dos objetos em um bucket do S3 com base em uma programação definida. Você pode configurar a funcionalidade Inventário Amazon S3 para incluir os seguintes metadados da funcionalidade Bloqueio de Objetos para os objetos:

- O prazo de retenção
- O modo de retenção
- O status da retenção jurídica

Para ter mais informações, consulte [Inventário do Amazon S3](#).

Gerenciar políticas da funcionalidade Ciclo de Vida do S3 com a funcionalidade Bloqueio de Objetos

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, uma versão bloqueada de um objeto não pode ser excluída por uma política de expiração da funcionalidade Ciclo de Vida do S3. A funcionalidade Bloqueio de Objetos é mantida independentemente da classe de armazenamento em que o objeto reside, bem como durante as transições da funcionalidade Ciclo de Vida do S3 entre classes de armazenamento.

Para obter mais informações sobre como gerenciar ciclos de vida de objetos, consulte [Gerenciando seu ciclo de vida de armazenamento](#).

Gerenciar marcadores de exclusão com a funcionalidade Bloqueio 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](#).

Note

Os marcadores de exclusão não são protegidos por WORM, independentemente de qualquer período de retenção ou da retenção legal no objeto subjacente.

Usar a funcionalidade Lente de Armazenamento do S3 com a funcionalidade Bloqueio de Objetos

Para ver métricas de bytes de armazenamento e contagem de objetos com Bloqueio de Objeto habilitado, você pode usar a Lente de Armazenamento do Amazon S3. A Lente de Armazenamento do S3 é um recurso de análise de armazenamento em nuvem que você pode usar para obter visibilidade em toda a organização sobre o uso e a atividade do armazenamento de objetos.

Para ter mais informações, consulte [Usar a Lente de Armazenamento do S3 para proteger seus dados](#).

Para obter uma lista completa de métricas, consulte [Glossário de métricas de lente de armazenamento do Amazon S3](#).

Fazer upload de objetos em um bucket habilitado para a funcionalidade Bloqueio de Objetos

O cabeçalho Content-MD5 é necessário em todas as solicitações de upload de objeto com período de retenção configurado usando a funcionalidade Bloqueio de Objetos. O resumo MD5 é uma forma de verificar a integridade do objeto depois de fazer upload dele em um bucket. 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. O console do S3 adiciona automaticamente esse cabeçalho, mas você precisa especificá-lo ao usar a API [PutObject](#).

Para ter mais informações, consulte [Usar Content-MD5 ao carregar objetos](#).

Configurar eventos e notificações

Use a funcionalidade Notificações de Eventos do Amazon S3 para rastrear o acesso e as alterações feitas nas configurações da funcionalidade Bloqueio de Objetos e nos dados usando o AWS CloudTrail. Para obter informações sobre o CloudTrail, consulte [O que é AWS CloudTrail?](#) no Guia do Usuário do AWS CloudTrail.

Você também pode usar o Amazon CloudWatch para gerar alertas com base nesses dados. Para obter informações sobre o CloudWatch, consulte [O que é o Amazon CloudWatch?](#) no Guia do usuário do Amazon CloudWatch.

Definir limites para os períodos de retenção com uma política de bucket

É possível definir os períodos de retenção mínimo e máximo permitidos para um bucket usando uma política de bucket. 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:::DOC-EXAMPLE-BUCKET1/*",
      "Condition": {
        "NumericGreaterThan": {
          "s3:object-lock-remaining-retention-days": "10"
        }
      }
    }
  ]
}
```

```
]
}
```

Note

Se o seu bucket for o bucket de destino de uma configuração de replicação, você poderá definir períodos de retenção mínimo e máximo permitidos para as réplicas de objetos criadas usando a replicação. Para fazer isso, você deve permitir a ação `s3:ReplicateObject` na política de bucket. Para obter mais informações sobre as permissões de replicação, consulte [the section called “Configuração de permissões”](#).

Para obter mais informações sobre políticas de bucket, consulte os seguintes tópicos:

- [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço
- [Operações com objetos](#)
- [Exemplos de políticas de bucket usando chaves de condição](#)

Configurar a funcionalidade Bloqueio de Objetos do S3

Com a funcionalidade Bloqueio de Objetos do Amazon S3, é possível armazenar objetos no Amazon S3 usando um modelo de gravação única e várias leituras (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 informações gerais sobre os recursos da funcionalidade Bloqueio de Objetos, consulte [Usar o bloqueio de objetos do S3](#).

Antes de bloquear qualquer objeto, é necessário habilitar as funcionalidades Versionamento e Bloqueio de Objetos do S3 em um bucket. Depois, você pode definir um período de retenção, uma retenção jurídica ou ambos.

Para trabalhar com a funcionalidade Bloqueio de Objetos, é necessário ter determinadas permissões. Para obter uma lista das permissões relacionadas a várias operações da funcionalidade Bloqueio de Objetos, consulte [the section called “Permissões obrigatórias”](#).

Important

- Depois de habilitar a funcionalidade Bloqueio de Objetos em um bucket, não será possível desabilitá-la nem suspender o versionamento nesse bucket.

- Os buckets do S3 com funcionalidade Bloqueio de Objetos não podem ser usados como buckets de destino para logs de acesso ao servidor. Para ter mais informações, consulte [the section called “Registrando acesso ao servidor”](#).

Tópicos

- [Habilitar a funcionalidade Bloqueio de Objetos ao criar um bucket do S3](#)
- [Habilitar a funcionalidade Bloqueio de Objetos em um bucket do S3 existente](#)
- [Definir ou modificar uma retenção jurídica em um objeto do S3](#)
- [Definir ou modificar um período de retenção em um objeto do S3](#)
- [Definir ou modificar um período de retenção padrão em um bucket do S3](#)

Habilitar a funcionalidade Bloqueio de Objetos ao criar um bucket do S3

Você pode habilitar a funcionalidade Bloqueio de Objetos ao criar um bucket do S3 usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI), os AWS SDKs 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. No painel de navegação à esquerda, escolha Buckets.
3. Selecione Create bucket (Criar bucket).

A página Create bucket (Criar bucket) é aberta.

4. Para Nome do bucket, insira um nome para o bucket.

Note

Depois de criar um bucket, não será possível mudar o nome dele. Para obter mais informações sobre como nomear buckets, consulte [Regras de nomeação de bucket](#).

5. Em Region (Região), escolha a Região da AWS onde deseja que o bucket resida.
6. Em Propriedade de Objetos, escolha desabilitar ou habilitar as listas de controle de acesso (ACLs) e controlar a propriedade de objetos carregados no bucket.

7. Em Configurações de bloqueio de acesso público para este bucket, selecione as configurações de bloqueio de acesso público que deseja aplicar ao bucket.
8. Em Versionamento de bucket, escolha Habilitado.

A funcionalidade Bloqueio de Objetos só funciona em buckets com versionamento.

9. (Opcional) Em Tags (Etiquetas), você pode optar por adicionar etiquetas ao seu bucket. As etiquetas são pares de chave-valor usados para categorizar o armazenamento e alocar os custos.
10. Em Configurações avançadas, encontre Bloqueio de objeto e escolha Habilitar.

Você deve reconhecer que a ativação da funcionalidade Bloqueio de Objetos permitirá permanentemente que os objetos desse bucket sejam bloqueados.

11. Selecione Criar bucket.

Usando a AWS CLI

O seguinte exemplo `create-bucket` cria um bucket do S3 chamado *DOC-EXAMPLE-BUCKET1* com a funcionalidade Bloqueio de Objetos habilitada:

```
aws s3api create-bucket --bucket DOC-EXAMPLE-BUCKET1 --object-lock-enabled-for-bucket
```

Para obter mais informações e exemplos, consulte [create-bucket](#) na Referência de comandos da AWS CLI.

Note

Você pode executar comandos da AWS CLI pelo console usando o AWS CloudShell. O AWS CloudShell é um shell pré-autenticado e baseado em navegador que você pode inicializar diretamente do AWS Management Console. Para obter mais informações, consulte [What is CloudShell?](#) no Guia do usuário do AWS CloudShell.

Uso da API REST

É possível usar a API REST para criar um bucket do S3 com a funcionalidade Bloqueio de Objetos habilitada. Para obter mais informações, consulte [CreateBucket](#) na Referência da API do Amazon Simple Storage Service.

Uso dos AWS SDKs

Consulte exemplos de como habilitar o Bloqueio de Objetos ao criar um bucket do S3 com os SDKs da AWS em [Usar CreateBucket com um SDK da AWS ou uma ferramenta de linha de comandos](#).

Consulte exemplos de como obter a configuração atual do Bloqueio de Objetos com os SDKs da AWS em [Usar GetObjectLockConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#).

Para obter informações gerais sobre como usar AWS SDKs diferentes, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers](#).

Habilitar a funcionalidade Bloqueio de Objetos em um bucket do S3 existente

Você pode habilitar a funcionalidade Bloqueio de Objetos para um bucket do S3 existente usando o console do Amazon S3, a AWS CLI, os AWS SDKs ou a API REST do Amazon S3.

Uso do console do S3

Note

A funcionalidade Bloqueio de Objetos só funciona em buckets com versionamento.

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 para o qual você deseja habilitar a funcionalidade Bloqueio de Objetos.
4. Escolha a guia Properties (Propriedades).
5. Em Propriedades, role para baixo até a seção Bloqueio de objeto e escolha Editar.
6. Em Bloqueio de objeto, escolha Habilitar.

Você deve reconhecer que a ativação da funcionalidade Bloqueio de Objetos permitirá permanentemente que os objetos desse bucket sejam bloqueados.

7. Escolha Salvar alterações.

Usando a AWS CLI

O seguinte exemplo de comando `put-object-lock-configuration` define um período de retenção de 50 dias da funcionalidade Bloqueio de Objetos em um bucket chamado ***DOC-EXAMPLE-BUCKET1***:

```
aws s3api put-object-lock-configuration --bucket DOC-EXAMPLE-BUCKET1 --object-lock-configuration='{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 } } }'
```

Para obter mais informações e exemplos, consulte [put-object-lock-configuration](#) na Referência de comandos da AWS CLI.

Note

Você pode executar comandos da AWS CLI pelo console usando o AWS CloudShell. O AWS CloudShell é um shell pré-autenticado e baseado em navegador que você pode inicializar diretamente do AWS Management Console. Para obter mais informações, consulte [What is CloudShell?](#) no Guia do usuário do AWS CloudShell.

Uso da API REST

Você pode usar a API REST do Amazon S3 para habilitar a funcionalidade Bloqueio de Objetos em um bucket do S3 existente. Para obter mais informações, consulte [PutObjectLockConfiguration](#) na Referência da API do Amazon Simple Storage Service.

Uso dos AWS SDKs

Consulte exemplos de como habilitar o Bloqueio de Objetos para um bucket do S3 existente com os SDKs da AWS em [Usar PutObjectLockConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#).

Consulte exemplos de como obter a configuração atual do Bloqueio de Objetos com os SDKs da AWS em [Usar GetObjectLockConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#).

Para obter informações gerais sobre como usar AWS SDKs diferentes, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers](#).

Definir ou modificar uma retenção jurídica em um objeto do S3

Você pode definir ou remover uma retenção jurídica de um objeto do S3 usando o console do Amazon S3, a AWS CLI, os AWS SDKs ou a API REST do Amazon S3.

Important

- Se você quiser definir uma retenção jurídica em um objeto, o bucket do objeto já deverá ter a funcionalidade Bloqueio de Objetos habilitada.
- Quando você usa PUT em uma versão de objeto que tem um modo e um período de retenção individuais explícitos em um bucket, as configurações individuais da funcionalidade Bloqueio de Objetos da versão do objeto substituem qualquer configuração de retenção de propriedades do bucket.

Para ter mais informações, consulte [the section called “Retenções legais”](#).

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 Buckets.
3. Na lista Buckets, escolha o nome do bucket que contém o objeto no qual você deseja definir ou modificar uma retenção jurídica.
4. Na lista Objetos, selecione o objeto no qual você deseja definir ou modificar uma retenção jurídica.
5. Na página Propriedades do objeto, encontre a seção Retenção jurídica do Bloqueio de objeto e escolha Editar.
6. Escolha Habilitar para definir uma retenção jurídica ou Desabilitar para remover uma retenção jurídica.
7. Escolha Salvar alterações.

Usando a AWS CLI

O seguinte exemplo de `put-object-legal-hold` define uma retenção jurídica no objeto *my-image.fs* no bucket chamado *DOC-EXAMPLE-BUCKET1*:

```
aws s3api put-object-legal-hold --bucket DOC-EXAMPLE-BUCKET1 --key my-image.fs --legal-hold="Status=ON"
```

O seguinte exemplo de `put-object-legal-hold` remove uma retenção jurídica do objeto *my-image.fs* no bucket chamado *DOC-EXAMPLE-BUCKET1*:

```
aws s3api put-object-legal-hold --bucket DOC-EXAMPLE-BUCKET1 --key my-image.fs --legal-hold="Status=OFF"
```

Para obter mais informações e exemplos, consulte [put-object-legal-hold](#) na Referência de comandos da AWS CLI.

Note

Você pode executar comandos da AWS CLI pelo console usando o AWS CloudShell. O AWS CloudShell é um shell pré-autenticado e baseado em navegador que você pode inicializar diretamente do AWS Management Console. Para obter mais informações, consulte [What is CloudShell?](#) no Guia do usuário do AWS CloudShell.

Uso da API REST

É possível usar a API REST para definir ou modificar uma retenção jurídica em um objeto. Para obter mais informações, consulte [PutObjectLegalHold](#) na Referência da API do Amazon Simple Storage Service.

Uso dos AWS SDKs

Consulte exemplos de como definir uma retenção legal em um objeto com os SDKs da AWS em [Usar PutObjectLegalHold com um AWS SDK ou ferramenta de linha de comando](#).

Consulte exemplos de como obter o status atual de retenção legal com os SDKs da AWS em [Obter a configuração de retenção legal de um objeto do Amazon S3 usando um SDK da AWS](#).

Para obter informações gerais sobre como usar AWS SDKs diferentes, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers](#).

Definir ou modificar um período de retenção em um objeto do S3

Você pode definir ou modificar um período de retenção em um objeto do S3 usando o console do Amazon S3, a AWS CLI, os AWS SDKs ou a API REST do Amazon S3.

Important

- Se você quiser definir um período de retenção em um objeto, o bucket do objeto já deverá ter a funcionalidade Bloqueio de Objetos habilitada.
- Quando você usa PUT em uma versão de objeto que tem um modo e um período de retenção individual explícitos em um bucket, as configurações individuais da funcionalidade Bloqueio de Objetos da versão do objeto substituem qualquer configuração de retenção de propriedades do bucket.
- A única maneira de excluir um objeto no modo de conformidade antes da data de retenção é excluir a Conta da AWS associada.

Para ter mais informações, consulte [Períodos de retenção](#).

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 Buckets.
3. Na lista Buckets, escolha o nome do bucket que contém o objeto no qual você deseja definir ou modificar um período de retenção.
4. Na lista Objetos, selecione o objeto no qual você deseja definir ou modificar um período de retenção.
5. Na página Propriedades do objeto, encontre a seção Retenção de Bloqueio de objetos e escolha Editar.
6. Em Retenção, escolha Habilitar para definir um período de retenção ou Desabilitar para remover um período de retenção.
7. Se você escolher Habilitar, em Modo de retenção, escolha o Modo de governança ou o Modo de conformidade. Para ter mais informações, consulte [Modos de retenção](#).

8. Em Prazo de retenção, escolha a data em que deseja que o período de retenção termine. Durante esse período, o objeto será protegido por WORM e não poderá ser substituído nem excluído. Para ter mais informações, consulte [Períodos de retenção](#).
9. Selecione Save changes.

Usando a AWS CLI

O seguinte exemplo de `put-object-retention` define um período de retenção no objeto `my-image.fs` no bucket chamado `DOC-EXAMPLE-BUCKET1` até 1.º de janeiro de 2025:

```
aws s3api put-object-retention --bucket DOC-EXAMPLE-BUCKET1 --key my-image.fs --retention='{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

Para obter mais informações e exemplos, consulte [put-object-retention](#) na Referência de comandos da AWS CLI.

Note

Você pode executar comandos da AWS CLI pelo console usando o AWS CloudShell. O AWS CloudShell é um shell pré-autenticado e baseado em navegador que você pode inicializar diretamente do AWS Management Console. Para obter mais informações, consulte [What is CloudShell?](#) no Guia do usuário do AWS CloudShell.

Uso da API REST

É possível usar a API REST para definir um período de retenção em um objeto. Para obter mais informações, consulte [PutObjectRetention](#) na Referência da API do Amazon Simple Storage Service.

Uso dos AWS SDKs

Consulte exemplos de como definir um período de retenção em um objeto com os SDKs da AWS em [Usar PutObjectRetention com um AWS SDK ou ferramenta de linha de comando](#).

Consulte exemplos de como obter o período de retenção em um objeto com os SDKs da AWS em [Usar GetObjectRetention com um AWS SDK ou ferramenta de linha de comando](#).

Para obter informações gerais sobre como usar AWS SDKs diferentes, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers](#).

Definir ou modificar um período de retenção padrão em um bucket do S3

Você pode definir ou modificar um período de retenção padrão em um bucket do S3 usando o console do Amazon S3, a AWS CLI, os AWS SDKs ou a API REST do Amazon S3. Especifique uma duração, em dias ou anos, para proteger cada versão do objeto colocada no bucket.

Important

- Se você quiser definir um período de retenção padrão em um bucket, o bucket já deverá ter a funcionalidade Bloqueio de Objetos habilitada.
- Quando você usa PUT em uma versão de objeto que tem um modo e um período de retenção individual explícitos em um bucket, as configurações individuais da funcionalidade Bloqueio de Objetos da versão do objeto substituem qualquer configuração de retenção de propriedades do bucket.
- A única maneira de excluir um objeto no modo de conformidade antes da data de retenção é excluir a Conta da AWS associada.

Para ter mais informações, consulte [Períodos de retenção](#).

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 Buckets.
3. Na lista Buckets, escolha o nome do bucket no qual você deseja definir ou modificar um período de retenção padrão.
4. Escolha a guia Properties (Propriedades).
5. Em Propriedades, role para baixo até a seção Bloqueio de objeto e escolha Editar.
6. Em Retenção padrão, escolha Habilitar para definir uma retenção padrão ou Desabilitar para remover uma retenção padrão.
7. Se você escolher Habilitar, em Modo de retenção, escolha o Modo de governança ou o Modo de conformidade. Para ter mais informações, consulte [Modos de retenção](#).
8. Em Período de retenção padrão, escolha o número de dias ou anos para a duração do período de retenção. Os objetos colocados nesse bucket ficarão bloqueados durante esse número de dias ou anos. Para ter mais informações, consulte [Períodos de retenção](#).

9. Selecione Save changes.

Usando a AWS CLI

O seguinte exemplo de comando `put-object-lock-configuration` define um período de retenção de 50 dias da funcionalidade Bloqueio de Objetos no bucket chamado `DOC-EXAMPLE-BUCKET1` usando o modo de conformidade:

```
aws s3api put-object-lock-configuration --bucket DOC-EXAMPLE-BUCKET1 --object-lock-configuration='{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 } } }'
```

O seguinte exemplo de `put-object-lock-configuration` remove a configuração de retenção padrão de um bucket:

```
aws s3api put-object-lock-configuration --bucket DOC-EXAMPLE-BUCKET1 --object-lock-configuration='{ "ObjectLockEnabled": "Enabled" }'
```

Para obter mais informações e exemplos, consulte [put-object-lock-configuration](#) na Referência de comandos da AWS CLI.

Note

Você pode executar comandos da AWS CLI pelo console usando o AWS CloudShell. O AWS CloudShell é um shell pré-autenticado e baseado em navegador que você pode inicializar diretamente do AWS Management Console. Para obter mais informações, consulte [What is CloudShell?](#) no Guia do usuário do AWS CloudShell.

Uso da API REST

É possível usar a API REST para definir um período de retenção padrão em um bucket do S3 disponível. Para obter mais informações, consulte [PutObjectLockConfiguration](#) na Referência da API do Amazon Simple Storage Service.

Uso dos AWS SDKs

Consulte exemplos de como definir um período de retenção padrão em um bucket do S3 existente com os SDKs da AWS em [Usar PutObjectLockConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos.](#)

Para obter informações gerais sobre como usar AWS SDKs diferentes, consulte [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers](#).

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](#)
- [Classe de armazenamento para otimizar automaticamente dados com padrões de acesso alterados ou desconhecidos](#)
- [Classes de armazenamento de objetos acessados com pouca frequência](#)
- [Classes de armazenamento para arquivamento de objetos](#)
- [Classe de armazenamento para o Amazon S3 no Outposts](#)
- [Comparar as classes de armazenamento do Amazon S3](#)
- [Configurar a classe de armazenamento de um objeto](#)


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.
- S3 Express One Zone: a classe Amazon S3 Express One Zone é uma classe de armazenamento de zona única e alto desempenho do Amazon S3 desenvolvida com o propósito específico de fornecer acesso consistente aos dados e com latência inferior a dez milissegundos para as aplicações mais sensíveis à latência. A classe S3 Express One Zone é a classe de

armazenamento de objetos em nuvem de menor latência disponível atualmente, com velocidades de acesso aos dados até dez vezes mais rápidas e custos de solicitação 50% mais baixos do que a classe S3 Standard. Com a classe S3 Express One Zone, os dados são armazenados de forma redundante em vários dispositivos dentro de uma única zona de disponibilidade. Para ter mais informações, consulte [O que é a classe S3 Express One Zone?](#).

- Redundância reduzida: a classe Reduced Redundancy Storage (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 apresenta melhor custo-benefício.

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

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 nas despesas operacionais. O S3 Intelligent-Tiering é a única classe de armazenamento na nuvem que 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. O S3 Intelligent-Tiering é a classe de armazenamento ideal 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 custo mais baixo. A classe S3 Intelligent-Tiering oferece economia automática de custos de armazenamento em três níveis de acesso de baixa latência e alto throughput. 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. 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:

- **Acesso frequente:** os objetos que foram carregados ou transferidos para o S3 Intelligent-Tiering são automaticamente armazenados no nível Acesso frequente.
- **Acesso infrequente:** o S3 Intelligent-Tiering move os objetos que não foram acessados durante 30 dias consecutivos para o nível Acesso infrequente.
- **Acesso instantâneo ao arquivamento:** com o S3 Intelligent-Tiering, todos os objetos que não foram acessados durante 90 dias consecutivos são movidos automaticamente para o nível Acesso instantâneo ao arquivamento.

Além desses três níveis, o S3 Intelligent-Tiering oferece dois níveis opcionais de acesso ao arquivamento:

- **Acesso ao arquivamento:** o S3 Intelligent-Tiering oferece a opção de ativar o nível Acesso ao arquivamento 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.
- **Acesso profundo ao arquivamento:** o S3 Intelligent-Tiering oferece a opção de ativar o nível Acesso profundo ao arquivamento 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.

Note

- Ative apenas o nível Archive Access por 90 dias se você quiser ignorar o nível Archive Instant Access. O nível Acesso ao arquivamento oferece um custo de armazenamento um pouco menor com tempos de recuperação de minutos a horas. O nível Acesso instantâneo ao arquivamento oferece acesso de milissegundos e alta performance de throughput.
- 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 ao arquivamento ou Acesso profundo ao arquivamento, restaure o objeto usando `RestoreObject`.

Você pode [migrar os dados recém-criados para o S3 Intelligent-Tiering](#), definindo-o como a classe de armazenamento padrão. Você também pode optar por ativar um ou ambos os níveis de acesso ao arquivamento usando a operação de API [PutBucketIntelligentTieringConfiguration](#), a AWS CLI ou o console do Amazon S3. Para obter informações sobre como usar o S3 Intelligent-Tiering e ativar os níveis de acesso ao arquivamento, consulte [Como usar o S3 Intelligent-Tiering](#).

Para acessar objetos no nível Archive Access ou Deep Archive Access, primeiro você precisa restaurá-los. Para ter mais informações, consulte [Restaurar objetos dos níveis Archive Access e Deep Archive Access do S3 Intelligent-Tiering](#).

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](#).

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](#).

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. Os objetos que forem excluídos, sobrescritos ou transferidos para uma classe de armazenamento diferente antes de 30 dias incorrerão na cobrança normal de uso do armazenamento mais uma cobrança proporcional pelo restante do mínimo de 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 resiliência 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 S3 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](#) 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 recriar os dados em caso de falha da zona de disponibilidade, e para réplicas de objeto ao configurar a Replicação do S3 Entre Regiões Diferentes (S3 CRR).

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](#).

Note

Quando você usa as classes de armazenamento do S3 Glacier, os objetos permanecem no Amazon S3. Não será possível acessá-los diretamente por meio do serviço separado do Amazon S3 Glacier. Consulte informações sobre o serviço Amazon S3 Glacier no [Guia do desenvolvedor do Amazon S3 Glacier](#).

As diferenças entre as classes de armazenamento do S3 Glacier 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 arquivamentos 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 excluir, substituir ou fizer a transição de um objeto para outra classe de armazenamento antes do período mínimo de 90 dias, será cobrado pelos 90 dias. O Amazon S3 oferece suporte a solicitações de restauração a uma taxa de até 1.000 transações por segundo, por Conta da AWS, para S3 Glacier Flexible Retrieval.

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 excluir, substituir ou fizer a transição de um objeto para outra classe de armazenamento antes do período mínimo de 180 dias, será cobrado pelos 180 dias. O Amazon S3 oferece suporte a solicitações de restauração a uma taxa de até 1.000 transações por segundo, por Conta da AWS, para S3 Glacier Deep Archive.

Para obter informações sobre a definição de preços, consulte [Definição de preços do Amazon S3](#).

O 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](#). Porém, os objetos em S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive não estão disponíveis para acesso em tempo real. Primeiro, é necessário restaurar os objetos em S3 Glacier Flexible Retrieval e em S3 Glacier Deep Archive antes de acessá-los. (Os objetos em S3 Standard, Reduced Redundancy Storage (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](#).

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.

Conceitos básicos das classes de armazenamento do S3 Glacier

Para saber mais sobre como usar as classes de armazenamento do Amazon S3 Glacier, consulte o [Tutorial: Conceitos básicos do uso das classes de armazenamento 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 operações de API e recursos no AWS Outposts da mesma maneira que no Amazon S3, incluindo políticas de acesso, criptografia e marcação. 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 ter mais informações, consulte [Usar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#).

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 ter mais informações, consulte [Como usar criptografia do lado do servidor com chaves fornecidas pelo cliente \(SSE-C\)](#).

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).

Para obter mais informações sobre o S3 on Outposts, consulte [O que é o Amazon S3 on Outposts?](#)

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

* O S3 Glacier Flexible Retrieval requer 40 KB de metadados adicionais para cada objeto arquivado. Isso inclui 32 KB de metadados cobrados na taxa do S3 Glacier Flexible Retrieval (necessários para identificar e recuperar dados) e mais 8 KB de dados cobrados na taxa do S3 Standard. A taxa do S3 Standard é necessária para manter o nome e os metadados definidos pelo usuário para objetos arquivados no S3 Glacier Flexible Retrieval. Para obter mais informações sobre as classes de armazenamento, consulte [Classes de armazenamento do Amazon S3](#).

** A classe S3 Glacier Deep Archive requer 40 KB de metadados adicionais para cada objeto arquivado. Isso inclui 32 KB de metadados cobrados de acordo com a taxa do S3 Glacier Deep Archive (necessários para identificar e recuperar dados) e mais 8 KB de dados cobrados na taxa do S3 Standard. A taxa do S3 Standard é necessária para manter o nome e os metadados definidos pelo usuário para objetos arquivados no Amazon S3 Glacier Deep Archive. Para obter mais informações sobre as classes de armazenamento, consulte [Classes de armazenamento do Amazon S3](#).

Lembre-se de que todas as classes de armazenamento, exceto as classes S3 One Zone-IA e S3 Express One Zone, foram desenvolvidas para serem resistentes à perda física de uma zona de disponibilidade resultante de desastres. Considere também 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). Todas essas abordagens usam as operações de API do Amazon S3 para enviar solicitações ao Amazon S3.

As operações de API 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 operações de API [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, S3 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 operação de 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 especificando os cabeçalhos de solicitação da seguinte forma:

- Defina o cabeçalho `x-amz-metadata-directive` como `COPY`.
- Defina o cabeçalho `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 copia o objeto, o Amazon S3 fornece um novo ID de versão.

- Será possível alterar a classe de armazenamento de um objeto usando o console do Amazon S3 se o tamanho do objeto for menor que 160 GB. Se for maior, recomendamos adicionar a configuração do ciclo de vida do S3 para alterar a classe de armazenamento do objeto.
- 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](#).
- 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](#).

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 a permissão `s3:PutObject`, você pode restringir uploads de objetos a uma classe de armazenamento específica. Para visualizar um exemplo de política, consulte [Exemplo: restringir uploads aos objetos com uma classe de armazenamento específica](#).

Para obter mais informações sobre como usar condições em políticas e uma lista completa de chaves de condição do Amazon S3, consulte os seguintes tópicos:

- [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço
- [Exemplos de políticas de bucket usando chaves de condição](#)

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](#)
- [Como usar o S3 Intelligent-Tiering](#)
- [Como gerenciar o S3 Intelligent-Tiering](#)

Como o S3 Intelligent-Tiering funciona

A classe de armazenamento Amazon 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 para dados que levem de minutos a horas para serem acessados, ative recursos de arquivamento a fim de adicionar mais dois níveis de acesso. Você pode rebaixar os objetos para o nível Acesso ao arquivamento, para o nível Acesso profundo ao arquivamento ou para ambos. Com o Archive Access, o S3 Intelligent-Tiering move os objetos que não foram acessados por no mínimo 90 dias consecutivos para o nível Archive Access. Com o Deep Archive Access, o S3 Intelligent-Tiering move os objetos para o nível Deep Archive Access depois de no mínimo 180 dias consecutivos sem acesso. Para os dois níveis, é possível configurar o número de dias sem acesso com base nas suas necessidades.

As ações a seguir constituem um acesso que impede rebaixar os objetos para o nível Archive Access ou Deep Archive Access:

- Baixar ou copiar um objeto por meio do console do Amazon S3.
- Invocar [CopyObject](#), [UploadPartCopy](#) ou replicar objetos com a replicação em lote do S3. Nesses casos, os objetos de origem das operações de cópia ou replicação são hierarquizados.

- Invocar [GetObject](#), [PutObject](#), [RestoreObject](#), [CompleteMultipartUpload](#), [ListParts](#) ou [SelectObjectContent](#)

Por exemplo, se os objetos forem acessados por meio de `SelectObjectContent` antes do número especificado de dias sem acesso (por exemplo, 180 dias), essa ação redefinirá o timer. Os objetos não serão movidos para o nível `Archive Access` ou `Deep Archive Access` até que o tempo depois da última solicitação `SelectObjectContent` atinge o número de dias especificado.

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`.

As ações a seguir constituem um acesso que automaticamente move os objetos do nível `Infrequent Access` ou `Archive Instant` de volta para o nível `Frequent Access`:

- Baixar ou copiar um objeto por meio do console do Amazon S3.
- Invocar [CopyObject](#), [UploadPartCopy](#) ou replicar objetos com a replicação em lote. Nesses casos, os objetos de origem das operações de cópia ou replicação são hierarquizados.
- Invocar [GetObject](#), [PutObject](#), [RestoreObject](#), [CompleteMultipartUpload](#) ou [ListParts](#).

Outras ações não constituem um acesso que automaticamente mova os objetos do nível `Infrequent Access` ou `Archive Instant Access` de volta para o nível `Frequent Access`. Veja a seguir uma amostra, não uma lista definitiva, destas ações:

- Invocar [HeadObject](#), [GetObjectTagging](#), [PutObjectTagging](#), [ListObjects](#), [ListObjectsV2](#) ou [ListObjectVersions](#)
- Invocar [SelectObjectContent](#) não constitui acesso que hierarquiza objetos até um nível `Frequent Access`. Além disso, ele não impede a hierarquização de objetos do nível `Frequent Access` para o nível `Infrequent Access` e, depois, para o nível `Archive Instant`.

É possível configurar o S3 Intelligent-Tiering como sua classe de armazenamento padrão para dados recém-criados especificando `INTELLIGENT-TIERING` em seu cabeçalho de solicitação [PutBucketIntelligentTieringConfiguration](#). 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

A seção a seguir explica os diferentes níveis de acesso automático e opcional. Ao mover objetos entre os níveis de acesso, a classe de armazenamento permanece a mesma (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 Frequent Access oferece baixa latência e throughput de alto desempenho.

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 Infrequent Access oferece baixa latência e throughput de alto desempenho.

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 de acesso de arquivamento instantâneo oferece baixa latência e throughput de alto desempenho.

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 comuns para esse nível de acesso podem variar de três a cinco horas. Se você iniciar a solicitação de restauração usando o Operações em Lote do S3, a

restauração começará em minutos. Para receber mais informações sobre as opções e prazos de recuperação, consulte [the section called “Restaurar objetos dos níveis Archive Access e Deep Archive Access do S3 Intelligent-Tiering”](#).


 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 Acesso instantâneo ao arquivamento 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. Se você iniciar a solicitação de restauração usando o Operações em Lote do S3, a restauração começará em nove horas. Para receber mais informações sobre as opções e prazos de recuperação, consulte [the section called “Restaurar objetos dos níveis Archive Access e Deep Archive Access do S3 Intelligent-Tiering”](#).

 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 de acesso Archive ou Deep Archive, será necessário primeiro restaurar o objeto usando a operação `RestoreObject`.

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 definir as configurações de ciclo de vida do S3 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 configurações de ciclo de vida do S3 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).

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](#).

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.

Exemplo [put-bucket-intelligent-tiering-configuration](#)

JSON

```
{
  "Id": "string",
  "Filter": {
    "Prefix": "string",
    "Tag": {
      "Key": "string",
      "Value": "string"
    },
    "And": {
      "Prefix": "string",
      "Tags": [
        {
          "Key": "string",
          "Value": "string"
        }
        ...
      ]
    }
  },
  "Status": "Enabled"|"Disabled",
  "Tierings": [
    {
      "Days": integer,
      "AccessTier": "ARCHIVE_ACCESS"|"DEEP_ARCHIVE_ACCESS"
    }
    ...
  ]
}
```

```
]
}
```

XML

```
PUT /?intelligent-tiering&id=Id HTTP/1.1
Host: Bucket.s3.amazonaws.com
<?xml version="1.0" encoding="UTF-8"?>
<IntelligentTieringConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>string</Id>
  <Filter>
    <And>
      <Prefix>string</Prefix>
      <Tag>
        <Key>string</Key>
        <Value>string</Value>
      </Tag>
      ...
    </And>
    <Prefix>string</Prefix>
    <Tag>
      <Key>string</Key>
      <Value>string</Value>
    </Tag>
  </Filter>
  <Status>string</Status>
  <Tiering>
    <AccessTier>string</AccessTier>
    <Days>integer</Days>
  </Tiering>
  ...
</IntelligentTieringConfiguration>
```

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.

Começar a usar o S3 Intelligent-Tiering

Para saber mais sobre como usar o S3 Intelligent-Tiering, consulte [Tutorial: Começar a usar o S3 Intelligent-Tiering](#).

Como gerenciar o S3 Intelligent-Tiering

A classe de armazenamento do 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 proteção de dados de objetos.
- Bloqueio de objeto do S3, para evitar exclusão acidental de dados.
- AWS PrivateLink, para acessar o Amazon S3 por um endpoint privado em uma nuvem privada virtual (VPC).

Identificar quais objetos do nível de acesso do S3 Intelligent-Tiering estão armazenados

Para receber uma lista dos objetos e dos metadados correspondentes, incluindo o nível S3 Intelligent-Tiering Access, você pode usar o [the section called “Gerenciamento de inventário”](#). O Inventário S3 fornece arquivos de saída CSV, ORC ou Parquet que listam seus objetos e os metadados correspondentes. Você pode receber esses relatórios de inventário diariamente ou semanalmente para um bucket do Amazon S3 ou um prefixo compartilhado. (Prefixo compartilhado refere-se a objetos que tenham nomes que começam com uma string em comum.)

Visualizar o status do arquivo de um objeto no S3 Intelligent-Tiering

Para receber uma notificação quando um objeto na classe de armazenamento do S3 Intelligent-Tiering for movido ao nível Archive Access ou ao Deep Archive Access, você pode configurar

notificações de evento do S3. 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 ter mais informações, consulte [Notificações de eventos do Amazon S3](#).

Veja a seguir um exemplo de mensagem que o Amazon S3 envia para publicar um evento `s3: IntelligentTiering`. Para ter mais informações, consulte [the section called “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/
JRWeUWerMUE5JgHvAN0jpD"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",
        "bucket": {
          "name": "mybucket",
          "ownerIdentity": {
            "principalId": "A3NL1K0ZZKExample"
          },
          "arn": "arn:aws:s3:::mybucket"
        },
        "object": {
          "key": "HappyFace.jpg",
```

```

        "size":1024,
        "eTag":"d41d8cd98f00b204e9800998ecf8427e",
    },
    "intelligentTieringEventData":{
        "destinationAccessTier": "ARCHIVE_ACCESS"
    }
}
]
}

```

Você também pode usar uma [solicitação do objeto HEAD](#) para visualizar o status de arquivamento de um objeto. Se um objeto for armazenado usando a classe de armazenamento do S3 Intelligent-Tiering e estiver no momento em um dos níveis de arquivamento, a resposta de objeto HEAD mostrará o nível de arquivamento atual. Para mostrar o nível de arquivamento, a solicitação usa o cabeçalho [x-amz-archive-status](#).

A solicitação de objeto HEAD a seguir retorna os metadados de um objeto (nesse caso, *my-image.jpg*).

Example

```

HEAD /my-image.jpg HTTP/1.1
Host: bucket.s3.region.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:02236Q3V0RonhpaBX5sCYVf1bNRuU=

```

Você também pode usar solicitações de objetos HEAD para monitorar o status de uma solicitação `restore-object`. Se a restauração do arquivo estiver em andamento, a resposta de objeto HEAD incluirá o cabeçalho [x-amz-restore](#).

O exemplo de resposta de objeto HEAD a seguir mostra 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: FSVaTMjrmBp3Izs1NnwBZeu7M19iI8UbxMbi0A8AirHANJBo+hEftBuiESACOMJp
x-amz-request-id: E5CEFCB143EB505A
Date: Fri, 13 Nov 2020 00:28:38 GMT
Last-Modified: Mon, 15 Oct 2012 21:58:07 GMT

```

```
ETag: "1accb31fcf202eba0c0f41fa2f09b4d7"  
x-amz-storage-class: 'INTELLIGENT_TIERING'  
x-amz-archive-status: 'ARCHIVE_ACCESS'  
x-amz-restore: 'ongoing-request="true"'  
x-amz-restore-request-date: 'Fri, 13 Nov 2020 00:20:00 GMT'  
Accept-Ranges: bytes  
Content-Type: binary/octet-stream  
Content-Length: 300  
Server: AmazonS3
```

Restaurar objetos dos níveis Archive Access e Deep Archive Access do S3 Intelligent-Tiering

Para acessar objetos nos níveis S3 Intelligent-Tiering Archive Access e Deep Archive Access, você deve iniciar uma [solicitação de restauração](#) e esperar até que o objeto seja movido para o nível Frequent Access. Para obter mais informações sobre objetos arquivados, consulte [the section called “Trabalhando 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). Depois, após um período mínimo de 90 dias consecutivos sem acesso, o objeto é transferido ao nível Archive Access. Depois de um período mínimo de 180 dias consecutivos sem acesso, o objeto é transferido automaticamente ao nível Deep Archive Access. Para ter mais informações, consulte [the section called “Como o S3 Intelligent-Tiering funciona”](#).

Você pode restaurar um objeto arquivado usando o console do Amazon S3, o Operações em Lote do S3, a API REST do Amazon S3, os SDKs da AWS ou a AWS Command Line Interface (AWS CLI). Para ter mais informações, consulte [the section called “Trabalhando com objetos arquivados”](#).

Gerenciando seu ciclo de vida de armazenamento

Para gerenciar objetos e armazená-los 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 ter mais informações, consulte [Uso de classes de armazenamento do Amazon S3](#).

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 ter mais informações, consulte [Expirando objetos](#).

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 o descontinuar imediatamente, você não receberá cobrança 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](#).

Para obter métricas detalhadas do Ciclo de Vida do S3, você pode usar as métricas da Lente de Armazenamento do Amazon S3. A Lente de Armazenamento do S3 é um recurso de análise de armazenamento em nuvem que você pode usar para obter visibilidade em toda a organização sobre o uso e a atividade do armazenamento de objetos. A Lente de Armazenamento do S3 fornece métricas de contagem de regras e métricas que você pode usar para identificar buckets que têm o Versionamento do S3 ativado ou uma alta porcentagem de bytes de versões desatualizadas. Para obter mais informações, consulte [Usar a Lente de Armazenamento do S3 para otimizar os custos do armazenamento](#).

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 archive 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, archive-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 pode criar uma configuração de ciclo de vida usando o console do Amazon S3, a API REST, os SDKs da AWS e a AWS Command Line Interface (AWS CLI). Para ter mais informações, consulte [Definir uma configuração de ciclo de vida em um bucket](#).

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:

[PutBucketLifecycleConfiguration](#)

[GetBucketLifecycleConfiguration](#)

[DeleteBucketLifecycle](#)

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](#)

- [Expirando objetos](#)
- [Definir uma configuração de ciclo de vida em um bucket](#)
- [Ciclo de vida e outras configurações de bucket](#)
- [Configurar notificações de evento de ciclo de vida](#)
- [Elementos de configuração do ciclo de vida](#)
- [Exemplos de configuração do S3 Lifecycle](#)

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](#). Alguns exemplos de quando você pode usar as configurações do ciclo de vida do S3 dessa maneira incluem:

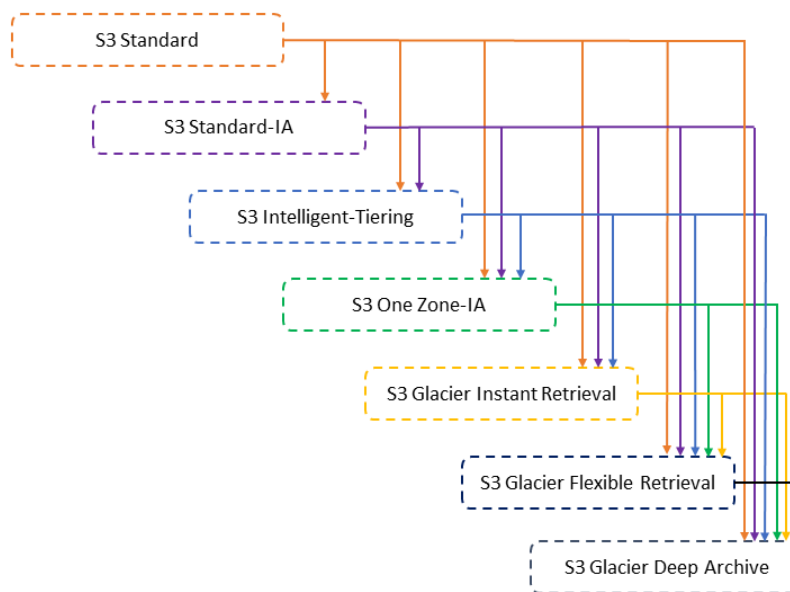
- Quando você sabe que os objetos são acessados com pouca frequência, você pode fazer a transição para a classe de armazenamento S3 Standard – IA.
- Convém arquivar objetos que não precisam de acesso em tempo real à classe de armazenamento S3 Glacier Flexible Retrieval.

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](#).

O Amazon S3 dá suporte a um modelo de cachoeira 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.

Note

Não há nenhuma cobrança de recuperação de dados para transições de ciclo de vida. No entanto, há cobranças de ingestão por solicitação ao usar regras de PUT, COPY ou ciclo de vida para transferir dados a qualquer classe de armazenamento do S3. Considere o custo de ingestão ou transição antes de transferir objetos a qualquer classe de armazenamento. Para obter mais informações sobre considerações de custo, consulte [Definição de preço do Amazon S3](#).

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 KiB: para as seguintes transições, o Amazon S3 não faz transição de objetos menores que 128 KiB:
 - 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.

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](#).

Dias mínimos para transição a S3 Standard-IA ou S3 One Zone-IA

Antes de fazer a transição de objetos para S3 Standard-IA ou S3 One Zone-IA, é necessário armazená-los pelo menos 30 dias no Amazon S3. 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 essa transição nos primeiros 30 dias porque os objetos mais recentes geralmente são acessados com mais frequência ou excluídos mais cedo do que o adequado para o 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. Para obter uma lista da duração mínima de armazenamento para todas as classes de armazenamento, consulte [Comparar as classes de armazenamento do Amazon S3](#).

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 evita os custos 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 archive 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 ter mais informações, consulte [Restaurar um objeto arquivado](#).

- 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.

Note

A aplicação da operação Copy a objetos restaurados não é permitida no console do Amazon S3 em objetos nas classes de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive. Para esse tipo de operação de cópia, use a AWS Command Line Interface(AWS CLI), AWS SDKs ou a API REST.

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 adicionais 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 agregar 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 arquivamento. 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 uma mistura de objetos que inclui objetos pequenos, especialmente objetos com menos de 128 KB, recomendamos usar o filtro de tamanho do objeto do ciclo de vida para remover objetos pequenos da transição e reduzir os custos da solicitação. As classes S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive não bloqueiam automaticamente a transição de objetos com menos de 128 KB.
- 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](#).

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 arquivamento, você paga pelo arquivamento (taxa de S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive) e por uma cópia restaurada temporariamente (taxa de armazenamento de S3 Standard). 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](#).

Expirando objetos

Quando um objeto atinge o fim do ciclo de vida, com base na respectiva configuração de ciclo de vida, o Amazon S3 realiza uma ação de acordo com o estado em que o bucket está.

- Bucket sem versionamento: o Amazon S3 coloca o objeto em uma fila para remoção e o remove de maneira assíncrona, eliminando-o permanentemente.
- Bucket com versionamento habilitado: 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: o Amazon S3 cria 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.

Para um bucket com versão (ou seja, versionamento habilitado ou suspenso), há várias considerações que orientam como o Amazon S3 trata a ação de expiração. Para buckets com versionamento habilitado ou suspenso, o seguinte se aplica:

- A expiração do objeto se aplica somente à versão atual de um objeto (não afeta versões desatualizadas de objetos).
- 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. Por exemplo, consulte [Exemplo 7: Remover marcadores de exclusão de objetos expirados](#).

- Você pode usar o elemento de ação `NoncurrentVersionExpiration` para instruir o Amazon S3 a excluir permanentemente versões de objetos desatualizadas. Esses objetos excluídos não podem ser recuperados. Esse prazo de validade pode se basear em determinado número de dias desde o momento em que os objetos se tornam desatualizados. Além do número de dias, você também pode fornecer um número máximo de versões desatualizadas que devem ser retidas (até cem). Esse valor especifica quantas versões desatualizadas mais recentes devem existir antes que o Amazon S3 possa executar a ação associada em uma determinada versão. Para especificar o número máximo de versões não atuais, também é necessário fornecer um elemento `Filter`. Se não especificar um elemento `Filter`, o Amazon S3 gerará um erro `InvalidRequest` quando você fornecer um número máximo de versões não atuais. Para obter mais informações sobre como usar o elemento de ação `NoncurrentVersionExpiration`, consulte [the section called “Elementos para descrever ações de ciclo de vida”](#).

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

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](#).

Como descobrir quando os objetos expiram

Para descobrir quando um objeto está programado para expirar, use as operações de API [HeadObject](#) ou [GetObject](#). Essas operações de API retornam os cabeçalhos de resposta que fornecem a data e a hora em que o objeto não pode mais ser armazenado em cache.

Note

- 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.
- Antes de atualizar, desabilitar ou excluir regras de ciclo de vida, use as operações de API LIST (como [ListObjectsV2](#), [ListObjectVersions](#) e [ListMultipartUploads](#)) ou [Inventário do Amazon S3](#) para verificar se o Amazon S3 fez a transição e descontinuou os objetos elegíveis com base em seus casos de uso.

Cobrança da duração mínima de armazenamento

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 [Preço do Amazon S3](#).

Definir uma configuração de ciclo de vida em um bucket

Esta seção explica como você pode definir uma configuração do Amazon S3 Lifecycle em um bucket usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI), os SDKs da AWS ou a API REST do Amazon S3. Para obter informações sobre a configuração do S3 Lifecycle, consulte [Gerenciando seu ciclo de vida de armazenamento](#).

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.

Note

Antes de atualizar, desabilitar ou excluir regras de ciclo de vida, use as operações de API LIST (como [ListObjectsV2](#), [ListObjectVersions](#) e [ListMultipartUploads](#)) ou [Inventário do Amazon S3](#) para verificar se o Amazon S3 fez a transição e descontinuou os objetos elegíveis com base em seus casos de uso. Se você estiver enfrentando problemas com a atualização, a desabilitação ou a exclusão de regras da funcionalidade Ciclo de Vida, consulte [Solução de problemas do ciclo de vida do Amazon S3](#).

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 regras de ciclo de vida para todos os objetos ou para um subconjunto de objetos em um bucket usando um prefixo compartilhado (nomes de objetos que começam com uma string comum) ou uma tag. Na regra de ciclo de vida, é possível definir ações específicas para versões atuais e não atuais do objeto. Para mais informações, consulte:

- [Gerenciando seu ciclo de vida de armazenamento](#)
- [Usando o versionamento em buckets do S3](#)

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 prefixos de nome de objeto, consulte [Criar nomes de chave de objeto](#). Para obter mais informações sobre tags de objeto, consulte [Categorizando seu armazenamento usando tags](#).


- 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 de objeto ou tamanho máximo de objeto, o valor deverá ser maior que 0 byte e no máximo 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:
- S3 Standard – IA
 - S3 Intelligent-Tiering
 - S3 One Zone-IA

- S3 Glacier Flexible Retrieval
 - S3 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](#). É possível definir transições para a versão atual ou as versões anteriores do objeto, ou para ambas. O versionamento permite que você mantenha várias versões de um objeto em um bucket. Para obter mais informações sobre versionamento, consulte [Uso do console do S3](#).

 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 ter mais informações, consulte [Transição de objetos usando o Amazon S3 Lifecycle](#).

9. Para fazer a transição de versões não atuais de objetos entre classes de armazenamento, em 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:
- S3 Standard – IA
 - S3 Intelligent-Tiering
 - S3 One Zone-IA
 - S3 Glacier Flexible Retrieval
 - S3 Glacier Deep Archive
- b. Em Dias após o objeto ficar desatualizado, insira o número de dias após a criação para fazer a transição do objeto.
10. Para expirar as versões atuais dos objetos, em Expire current versions of objects (Expirar versões atuais 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](#).

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](#).

13. Selecione Criar regra.

Se a regra não contiver erros, o Amazon S3 a 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 [Trabalhar com modelos do AWS CloudFormation](#) e [AWS::S3::Bucket](#) no Guia do usuário do AWS CloudFormation.

Usando 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](#).

A configuração do ciclo de vida do Amazon S3 é um arquivo XML. Porém, ao usar a AWS CLI, você não pode especificar o formato XML. Em vez disso, você deve especificar o formato JSON. Veja a seguir exemplos de configurações de ciclo de vida em XML e as configurações JSON equivalentes que podem ser especificadas em um comando da AWS CLI.

Considere o seguinte exemplo de configuração do S3 Lifecycle.

Example Exemplo 1

Example

XML

```
<LifecycleConfiguration>
  <Rule>
    <ID>ExampleRule</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>3650</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

JSON

```
{
  "Rules": [
    {
      "Filter": {
        "Prefix": "documents/"
      },
      "Status": "Enabled",
```

```
    "Transitions": [  
      {  
        "Days": 365,  
        "StorageClass": "GLACIER"  
      }  
    ],  
    "Expiration": {  
      "Days": 3650  
    },  
    "ID": "ExampleRule"  
  }  
]
```

Example Exemplo 2

Example

XML

```
<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">  
  <Rule>  
    <ID>id-1</ID>  
    <Expiration>  
      <Days>1</Days>  
    </Expiration>  
    <Filter>  
      <And>  
        <Prefix>myprefix</Prefix>  
        <Tag>  
          <Key>mytagkey1</Key>  
          <Value>mytagvalue1</Value>  
        </Tag>  
        <Tag>  
          <Key>mytagkey2</Key>  
          <Value>mytagvalue2</Value>  
        </Tag>  
      </And>  
    </Filter>  
    <Status>Enabled</Status>  
  </Rule>  
</LifecycleConfiguration>
```

JSON

```
{
  "Rules": [
    {
      "ID": "id-1",
      "Filter": {
        "And": {
          "Prefix": "myprefix",
          "Tags": [
            {
              "Value": "mytagvalue1",
              "Key": "mytagkey1"
            },
            {
              "Value": "mytagvalue2",
              "Key": "mytagkey2"
            }
          ]
        }
      },
      "Status": "Enabled",
      "Expiration": {
        "Days": 1
      }
    }
  ]
}
```

Teste o comando `put-bucket-lifecycle-configuration` da seguinte forma.

Como testar a configuração

1. Salve a configuração de ciclo de vida JSON em um arquivo (por exemplo, *lifecycle.json*).
2. Execute o comando da AWS CLI a seguir para definir a configuração do ciclo de vida no bucket. Substitua os *user input placeholders* por suas próprias informações.

```
$ aws s3api put-bucket-lifecycle-configuration \
--bucket DOC-EXAMPLE-BUCKET \
--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 DOC-EXAMPLE-BUCKET
```

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 DOC-EXAMPLE-BUCKET
```

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](#).

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 imprimindo o 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](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketLifecycleConfiguration;
import com.amazonaws.services.s3.model.BucketLifecycleConfiguration.Transition;
import com.amazonaws.services.s3.model.StorageClass;
import com.amazonaws.services.s3.model.Tag;
import com.amazonaws.services.s3.model.lifecycle.LifecycleAndOperator;
import com.amazonaws.services.s3.model.lifecycle.LifecycleFilter;
import com.amazonaws.services.s3.model.lifecycle.LifecyclePrefixPredicate;
import com.amazonaws.services.s3.model.lifecycle.LifecycleTagPredicate;

import java.io.IOException;
import java.util.Arrays;

public class LifecycleConfiguration {

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

        // Create a rule to archive objects with the "glacierobjects/"
prefix to Glacier
        // immediately.
        BucketLifecycleConfiguration.Rule rule1 = new
BucketLifecycleConfiguration.Rule()
            .withId("Archive immediately rule")
            .withFilter(new LifecycleFilter(new
LifecyclePrefixPredicate("glacierobjects/")))
            .addTransition(new
Transition().withDays(0).withStorageClass(StorageClass.Glacier))
            .withStatus(BucketLifecycleConfiguration.ENABLED);
```

```
        // Create a rule to transition objects to the Standard-Infrequent
Access storage
        // class
        // after 30 days, then to Glacier after 365 days. Amazon S3 will
delete the
        // objects after 3650 days.
        // The rule applies to all objects with the tag "archive" set to
"true".
        BucketLifecycleConfiguration.Rule rule2 = new
BucketLifecycleConfiguration.Rule()
            .withId("Archive and then delete rule")
            .withFilter(new LifecycleFilter(new
LifecycleTagPredicate(new Tag("archive", "true"))))
            .addTransition(new Transition().withDays(30)

.withStorageClass(StorageClass.StandardInfrequentAccess))
            .addTransition(new
Transition().withDays(365).withStorageClass(StorageClass.Glacier))
            .withExpirationInDays(3650)
            .withStatus(BucketLifecycleConfiguration.ENABLED);

        // Add the rules to a new BucketLifecycleConfiguration.
        BucketLifecycleConfiguration configuration = new
BucketLifecycleConfiguration()
            .withRules(Arrays.asList(rule1, rule2));

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

            // Save the configuration.
            s3Client.setBucketLifecycleConfiguration(bucketName,
configuration);

            // Retrieve the configuration.
            configuration =
s3Client.getBucketLifecycleConfiguration(bucketName);

            // Add a new rule with both a prefix predicate and a tag
predicate.
```

```
configuration.getRules().add(new
BucketLifecycleConfiguration.Rule().withId("NewRule")
                                .withFilter(new LifecycleFilter(new
LifecycleAndOperator(
                                Arrays.asList(new
LifecyclePrefixPredicate("YearlyDocuments/"),
                                new
LifecycleTagPredicate(new Tag(
                                "expire_after",
                                "ten_years"))))))))
                                .withExpirationInDays(3650)
.withStatus(BucketLifecycleConfiguration.ENABLED));

// Save the configuration.
s3Client.setBucketLifecycleConfiguration(bucketName,
configuration);

// Retrieve the configuration.
configuration =
s3Client.getBucketLifecycleConfiguration(bucketName);

// Verify that the configuration now has three rules.
configuration =
s3Client.getBucketLifecycleConfiguration(bucketName);
System.out.println("Expected # of rules = 3; found: " +
configuration.getRules().size());

// Delete the configuration.
s3Client.deleteBucketLifecycleConfiguration(bucketName);

// Verify that the configuration has been deleted by
attempting to retrieve it.
configuration =
s3Client.getBucketLifecycleConfiguration(bucketName);
String s = (configuration == null) ? "No configuration
found." : "Configuration found.";
System.out.println(s);
} catch (AmazonServiceException e) {
// The call was transmitted successfully, but Amazon S3
couldn't process
// it, so it returned an error response.
```



```
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the
client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

.NET

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](#).

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](#).

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

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

        private static async Task AddUpdateDeleteLifecycleConfigAsync()
        {
            try
            {
                var lifeCycleConfiguration = new LifecycleConfiguration()
                {
                    Rules = new List<LifecycleRule>
                    {
                        new LifecycleRule
                        {
                            Id = "Archive immediately rule",
                            Filter = new LifecycleFilter()
                            {
                                LifecycleFilterPredicate = new
LifecyclePrefixPredicate()
                                {
                                    Prefix = "glacierobjects/"
                                }
                            },
                            Status = LifecycleRuleStatus.Enabled,
                            Transitions = new List<LifecycleTransition>
```

```
        {
            new LifecycleTransition
            {
                Days = 0,
                StorageClass = S3StorageClass.Glacier
            }
        },
    },
    new LifecycleRule
    {
        Id = "Archive and then delete rule",
        Filter = new LifecycleFilter()
        {
            LifecycleFilterPredicate = new
LifecyclePrefixPredicate()
            {
                Prefix = "projectdocs/"
            }
        },
        Status = LifecycleRuleStatus.Enabled,
        Transitions = new List<LifecycleTransition>
        {
            new LifecycleTransition
            {
                Days = 30,
                StorageClass =
S3StorageClass.StandardInfrequentAccess
            },
            new LifecycleTransition
            {
                Days = 365,
                StorageClass = S3StorageClass.Glacier
            }
        },
        Expiration = new LifecycleRuleExpiration()
        {
            Days = 3650
        }
    }
};

// Add the configuration to the bucket.
```

```
        await AddExampleLifecycleConfigAsync(client,
        lifecycleConfiguration);

        // Retrieve an existing configuration.
        lifecycleConfiguration = await RetrieveLifecycleConfigAsync(client);

        // Add a new rule.
        lifecycleConfiguration.Rules.Add(new LifecycleRule
        {
            Id = "NewRule",
            Filter = new LifecycleFilter()
            {
                LifecycleFilterPredicate = new LifecyclePrefixPredicate()
                {
                    Prefix = "YearlyDocuments/"
                }
            },
            Expiration = new LifecycleRuleExpiration()
            {
                Days = 3650
            }
        });

        // Add the configuration to the bucket.
        await AddExampleLifecycleConfigAsync(client,
        lifecycleConfiguration);

        // Verify that there are now three rules.
        lifecycleConfiguration = await RetrieveLifecycleConfigAsync(client);
        Console.WriteLine("Expected # of rulest=3; found:{0}",
        lifecycleConfiguration.Rules.Count);

        // Delete the configuration.
        await RemoveLifecycleConfigAsync(client);

        // Retrieve a nonexistent configuration.
        lifecycleConfiguration = await RetrieveLifecycleConfigAsync(client);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered ***. Message:'{0}' when writing
        an object", e.Message);
    }
}
```

```
        catch (Exception e)
        {
            Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
        }
    }

    static async Task AddExampleLifecycleConfigAsync(IAmazonS3 client,
LifecycleConfiguration configuration)
    {
        PutLifecycleConfigurationRequest request = new
PutLifecycleConfigurationRequest
        {
            BucketName = bucketName,
            Configuration = configuration
        };
        var response = await client.PutLifecycleConfigurationAsync(request);
    }

    static async Task<LifecycleConfiguration>
RetrieveLifecycleConfigAsync(IAmazonS3 client)
    {
        GetLifecycleConfigurationRequest request = new
GetLifecycleConfigurationRequest
        {
            BucketName = bucketName
        };
        var response = await client.GetLifecycleConfigurationAsync(request);
        var configuration = response.Configuration;
        return configuration;
    }

    static async Task RemoveLifecycleConfigAsync(IAmazonS3 client)
    {
        DeleteLifecycleConfigurationRequest request = new
DeleteLifecycleConfigurationRequest
        {
            BucketName = bucketName
        };
        await client.DeleteLifecycleConfigurationAsync(request);
    }
}
```

```
}
```

Ruby

Você pode usar o AWS SDK for Ruby para gerenciar a configuração do S3 Lifecycle em um bucket usando 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](#). Para obter mais informações sobre o gerenciamento da configuração do ciclo de vida, consulte [Gerenciando seu ciclo de vida de armazenamento](#).

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.

- [PutBucketLifecycleConfiguration](#)
- [GetBucketLifecycleConfiguration](#)
- [DeleteBucketLifecycle](#)

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](#).

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 ter mais informações, consulte [Elementos de configuração do ciclo de vida](#).

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](#).

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”](#).

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 habilitado 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.
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 interrompe o carregamento fracionado 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.

Solução de problemas de ciclo de vida do S3

Para receber mais informações sobre como solucionar problemas comuns com o ciclo de vida do S3, consulte [Solução de problemas do ciclo de vida do Amazon S3](#).

Mais informações

- [Elementos de configuração do ciclo de vida](#)
- [Transição para as classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive \(arquivamento de objetos\)](#)
- [Definir uma configuração de ciclo de vida em um bucket](#)

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 no momento em que 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 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 ter mais informações, consulte [Notificações de eventos do Amazon S3](#).

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",
      "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/
JRWeUWerMUE5JgHvAN0jpD"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",
        "bucket": {
          "name": "DOC-EXAMPLE-BUCKET",
          "ownerIdentity": {
            "principalId": "A3NL1K0ZZKExample"
          },
          "arn": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
        },
        "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](#)
- [Elemento Status](#)
- [Elemento Filter](#)
- [Elementos para descrever ações de ciclo de vida](#)

Você especifica uma configuração do Amazon S3 Lifecycle como XML, que consiste em uma ou mais regras de ciclo de vida.

```
<LifecycleConfiguration>
  <Rule>
    ...
  </Rule>
  <Rule>
    ...
  </Rule>
</LifecycleConfiguration>
```

Cada regra consiste no seguinte:

- Metadados de regra que incluem um ID de regra e um status que indica se a regra está habilitada ou desabilitada. Se uma regra estiver desativada, o Amazon S3 não executará as ações especificadas nela.
- Um filtro que identifica os objetos aos quais a regra se aplica. Você pode especificar um filtro usando o tamanho do objeto, o prefixo de chave do objeto, uma ou mais tags 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](#).

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 do elemento <Status> pode ser `Enabled` ou `Disabled`. 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 chave: este exemplo mostra uma regra do S3 Lifecycle 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 chave `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](#).

- Especificação de 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 (*key*) e valor (*value*). 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 encapsular as tags no elemento `<And>`, como 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 a ambos: à chave e ao valor. Se você especificar somente um elemento `<Key>` e nenhum elemento `<Value>`, a regra será aplicada apenas aos objetos que corresponderem à chave da tag e que não tenham um valor especificado.
- 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 incluir 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 chave especificado e as tags especificadas. 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 5 TB. Algumas classes de armazenamento têm limitações mínimas de tamanho. Para ter mais informações, consulte [Comparar as classes de armazenamento do Amazon S3](#).

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
      <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
    </Filter>
    <Status>Enabled</Status>
    transition/expiration actions
  </Rule>
</LifecycleConfiguration>
```

Se você estiver especificando um intervalo de tamanho de objeto, o número inteiro `ObjectSizeGreaterThan` deve ser menor que o valor de `ObjectSizeLessThan`. 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 64.000 bytes.

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
      <And>
        <Prefix>key-prefix</Prefix>
        <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
        <ObjectSizeLessThan>64000</ObjectSizeLessThan>
      </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 de ação **Transition**: 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](#). 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 **Expiration**: a ação `Expiration` descontinua 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](#). O Amazon S3 deixa todos os objetos expirados indisponíveis. A remoção permanente dos objetos depende do estado de versionamento do bucket.
 - Bucket sem versionamento: 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 buckets com versionamento habilitado ou suspenso, o seguinte se aplica:
 - 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](#).

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

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 `null` como ID de versão. Esse marcador de exclusão substitui qualquer versão de objeto por um ID de versão nulo na hierarquia de versões, que exclui o objeto.

Além disso, o Amazon S3 fornece as seguintes ações que você pode usar para gerenciar versões de objeto não atuais em um bucket com versão (isto é, buckets com versionamento ativado e suspenso).

- Elemento de ação **NoncurrentVersionTransition**: use essa ação para especificar quando o Amazon S3 deve realizar a transição de objetos para a classe de armazenamento especificada. Esse prazo de validade pode se basear em determinado número de dias desde o momento em que os objetos se tornam desatualizados. Além do número de dias, você também pode fornecer um número máximo de versões desatualizadas que devem ser retidas (até cem). Esse valor determina quantas versões desatualizadas mais recentes devem existir antes que o Amazon S3 possa executar a ação associada em determinada versão. O Amazon S3 fará a transição de qualquer versão adicional não atual que ultrapasse o número especificado para reter.

Para especificar o número máximo de versões não atuais, também é necessário fornecer um elemento `Filter`. Se não especificar um elemento `Filter`, o 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](#). 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](#).

- Elemento de ação **`NoncurrentVersionExpiration`**: use essa ação para especificar que o Amazon S3 exclua permanentemente versões de objetos não atuais. Esses objetos excluídos não podem ser recuperados. Esse prazo de validade pode se basear em determinado número de dias desde o momento em que os objetos se tornam desatualizados. Além do número de dias, você também pode fornecer um número máximo de versões desatualizadas que devem ser retidas (até cem). Esse valor especifica quantas versões desatualizadas mais recentes devem existir antes que o Amazon S3 possa executar a ação associada em uma determinada versão. O Amazon S3 excluirá permanentemente qualquer versão adicional não atual que ultrapasse o número especificado para reter.

Para especificar o número máximo de versões não atuais, também é necessário fornecer um elemento `Filter`. Se não especificar um elemento `Filter`, o 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.º/1/2014, às 10h30 UTC, você crie um objeto denominado `photo.gif` (ID de versão 111111). Em 2/1/2014, às 11h30 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 8/1/2014, à 0h UTC, a regra de ciclo de vida para expiração executa e exclui permanentemente `photo.gif` (ID de versão 111111), cinco dias depois de se tornar 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](#).

Note

As configurações de ciclo de vida de expiração do objeto não removem carregamentos fracionados incompletos. Para remover carregamentos fracionados 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 as ações de configuração de ciclo de vida a seguir para instruir o Amazon S3 a interromper carregamentos fracionados incompletos.

- Elemento de ação **AbortIncompleteMultipartUpload**: use esse elemento para definir o tempo máximo (em dias) em que você deseja permitir que os carregamentos fracionados continuem em andamento. Se os carregamentos fracionados aplicáveis (determinados pelo prefix de nome de chave especificado na regra de ciclo de vida) não forem concluídos no período predefinido, o Amazon S3 interromperá os carregamentos fracionados incompletos. Para ter mais informações, consulte [Abortar um multipart upload](#).

 Note

Não é possível especificar essa ação de ciclo de vida em uma regra que tenha um filtro que use tags de objeto.

- Elemento de ação **ExpiredObjectDeleteMarker**: em um bucket com versionamento habilitado, 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 Amazon S3 a remover marcadores de exclusão de objeto expirado. Para ver um exemplo, consulte [Exemplo 7: Remover marcadores de exclusão de objetos expirados](#).

 Note

Não é possível especificar essa ação de ciclo de vida em uma regra que tenha um filtro que use 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 [Restaurar versões anteriores](#), você pode usar qualquer um dos dois métodos seguintes para recuperar versões anteriores de um objeto:

- Método 1: copiar 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.
- Método 2: excluir 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 S3 Lifecycle com buckets habilitados para versionamento, a prática recomendada é usar o método 1.

O S3 Lifecycle opera sob um modelo de consistência final. Existe a possibilidade de uma versão atual excluída permanentemente não desaparecer enquanto as alterações não forem propagadas para todos os sistemas do Amazon S3. (Por isso, temporariamente, o Amazon S3 pode não perceber essa 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. Portanto, copiar a versão antiga, como recomendado no método 1, 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) do objeto, no qual o Amazon S3 pode realizar a ação especificada.

Quando você especificar o número de dias nas ações *Transition* e *Expiration* em uma configuração de ciclo de vida do S3, observe o seguinte:

- O valor que você especifica é o número de dias que devem transcorrer desde a criação do objeto para que a ação ocorra.
- 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 15/1/2014, às 10h30 UTC, e você especificar três dias em uma regra de transição, a data de transição do objeto será calculada como 19/1/2014, 0h 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 Última modificação no painel Propriedades do objeto. Quando você cria inicialmente um objeto, essa data reflete a data em que o objeto é criado. Se você substituir o objeto, a data será alterada conforme necessário. Portanto, a data de criação é idêntica à data da Última modificação.


Ao especificar o número de dias nas ações `NoncurrentVersionTransition` e `NoncurrentVersionExpiration` em uma configuração de ciclo de vida, observe o seguinte:

- O valor que você especifica é o número de dias que devem transcorrer desde o momento em que a versão do objeto se torna desatualizada (ou seja, quando o objeto é sobrescrito ou excluído) para que o Amazon S3 execute a ação em um ou mais 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 1º/1/2014, às 10h30 UTC. Se a nova versão do objeto que substitui a atual tiver sido criada em 15/1/2014, às 10h30 UTC, e você especificar uma regra de transição de três dias, a data de transição do objeto será calculada como 19/1/2014, 0h 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 do S3 Lifecycle 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 `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 Amazon S3 também continuará invalidando todos os objetos que você criar no bucket. Para interromper a ação de ciclo de vida, você deve remover a ação da regra 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

Não é possível criar regras de ciclo de vida com base em data usando o console do Amazon S3, mas você pode visualizar, desabilitar 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](#)
- [Exemplo 2: desabilitar uma regra do Lifecycle](#)
- [Exemplo 3: Rebaixar uma classe de armazenamento pela duração do ciclo de vida de um objeto](#)
- [Exemplo 4: Especificar várias regras](#)
- [Exemplo 5: sobreposição de filtros, ações de ciclo de vida conflitantes e o que o Amazon S3 faz com buckets sem versionamento](#)
- [Exemplo 6: Especificar uma regra de ciclo de vida para um bucket com versionamento habilitado](#)
- [Exemplo 7: Remover marcadores de exclusão de objetos expirados](#)
- [Exemplo 8: Configuração de ciclo de vida para anular multipart uploads](#)
- [Exemplo 9: configuração do ciclo de vida usando regras baseadas em tamanho](#)

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>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>3650</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

Em vez de especificar a idade do objeto em termos de dias após a criação, você pode especificar uma data para cada ação. No entanto, você não pode usar Date e Days na mesma regra.

- Se você quiser que a regra 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 são elegíveis

para arquivamento na classe S3 Glacier Flexible Retrieval à meia-noite (UTC) após a criação. Para obter mais informações sobre restrições de ciclo de vida, consulte [Restrições](#).

```
<LifecycleConfiguration>
  <Rule>
    <ID>Archive all object same-day upon creation</ID>
    <Filter>
      <Prefix></Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

- Você pode especificar zero ou um prefixo de nome de chave ou zero ou mais tags de objeto em um filtro. O exemplo de código a seguir aplica a regra 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 é elegível para transição para as classes S3 Glacier Flexible Retrieval e S3 Standard-IA (ou S3 One Zone-IA), o Amazon S3 escolhe a transição para a classe 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](#).

Exemplo 2: desabilitar uma regra do Lifecycle

Você pode desabilitar temporariamente uma regra da funcionalidade Ciclo de Vida do S3. 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 configuração, 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>GLACIER</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>Rule2</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Disabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

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.

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 é elegível para transição para as classes S3 Glacier Flexible Retrieval e S3 Standard-IA (ou S3 One Zone-IA), o Amazon S3 escolhe a transição para a classe 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](#).

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 excluí-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>
```

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 é elegível para transição para as classes S3 Glacier Flexible Retrieval e S3 Standard-IA (ou S3 One Zone-IA), o Amazon S3 escolhe a transição para a classe 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](#).

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 sobrepuerem, 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 sobrepuerem, 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.

Exemplo 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.

Exemplo 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 configuração 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.

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 é elegível para transição para as classes S3 Glacier Flexible Retrieval e S3 Standard-IA (ou S3 One Zone-IA), o Amazon S3 escolhe a transição para a classe 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](#).

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](#).) 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, diante da expectativa de que a frequência de acesso das versões atuais diminua 90 dias após a criação, você pode 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>GLACIER</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 configuração da funcionalidade Ciclo de Vida do S3 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.

Nesse caso, você pode usar a configuração de ciclo de vida do S3 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 ter mais informações, consulte [Usando o versionamento em buckets do S3](#).

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](#).

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 (`ObjectSizeGreaterThan`), um tamanho máximo

(`ObjectSizeLessThan`) 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>
        <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
      </And>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Se você especificar um intervalo usando os elementos `ObjectSizeGreaterThan` e `ObjectSizeLessThan`, 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 64.000 bytes.

```
<LifecycleConfiguration>
  <Rule>
    ...
    <And>
      <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
      <ObjectSizeLessThan>64000</ObjectSizeLessThan>
    </And>
  </Rule>
</LifecycleConfiguration>
```

Você também pode criar regras para expirar especificamente objetos desatualizados que não têm nenhum dado, incluindo objetos de marcador de exclusão desatualizados criados em um bucket com versionamento habilitado. O exemplo a seguir usa a ação `NoncurrentVersionExpiration` para remover as versões desatualizadas 30 dias após se tornarem desatualizadas e reter até dez versões desatualizadas dos objetos. Ele também usa o elemento `ObjectSizeLessThan` para filtrar somente objetos sem nenhum dado.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Expire noncurrent with size less than 1 byte</ID>
    <Filter>
      <ObjectSizeLessThan>1</ObjectSizeLessThan>
    </Filter>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
      <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
      <NoncurrentDays>30</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

Inventário do Amazon S3

Important

O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).

É possível usar o Inventário Amazon S3 para ajudar a gerenciar o armazenamento. Por exemplo, 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 Inventário Amazon S3, que fornece uma alternativa programada para as operações síncronas da API List do Amazon S3. O Inventário Amazon S3 não usa as operações de API List para auditar os objetos e não afeta a taxa de solicitação do bucket.

O Inventário Amazon S3 fornece arquivos de saída nos formatos CSV (valores separados por vírgulas), [Apache ORC \(colunar de linhas otimizado do Apache\)](#) ou [Apache Parquet](#) que listam os objetos e os metadados correspondentes, diária ou semanalmente, para um bucket do S3 ou objetos com prefixo compartilhado (ou seja, objetos que tenham nomes que comecem com uma string comum). Se você configurar um inventário semana, um relatório será gerado todo domingo (no fuso horário UTC) depois do 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. Ao configurar uma lista de inventário, é possível especificar o seguinte:

- Quais metadados de objeto incluir no inventário
- Se deve listar todas as versões do objeto ou somente as versões atuais
- Onde armazenar a saída do arquivo da lista de inventário
- Se o inventário deve ser gerado diariamente ou semanalmente
- Se o arquivo da lista de inventário deve ser criptografado

É possível consultar o Inventário Amazon S3 com consultas SQL padrão usando o [Amazon Athena](#), o [Amazon Redshift Spectrum](#) e outras ferramentas, como [Presto](#), [Apache Hive](#) e [Apache Spark](#). Para obter mais informações sobre como usar o Athena para consultar os arquivos de inventário, consulte [the section called “Consultar o inventário com o Athena”](#).

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. É possível obter uma lista de inventários de um bucket inteiro ou filtrar a lista pelo prefixo do nome da chave do 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 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 todos os arquivos de listas de inventários armazenados no bucket de destino. Para ter mais informações, consulte [Manifesto de inventário](#).
- 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. Um arquivo da lista de inventários é armazenado no bucket de destino com um dos seguintes formatos:

- Um arquivo CSV compactado com GZIP
- Como um arquivo colunar de linha otimizado (ORC) do Apache compactado com ZLIB
- Como um arquivo Apache Parquet compactado com Snappy

Note

Não há garantia de classificação em nenhuma ordem dos objetos nos relatórios 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 listado:

- 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 ter mais informações, consulte [Usando o versionamento em buckets do S3](#). (O campo não é incluído caso a lista seja configurada somente para a versão atual dos objetos.)
- **IsLatest:** definido como `True` se o objeto for a versão atual do objeto. (O campo não é incluído caso a lista seja configurada somente para a versão atual dos objetos.)
- **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](#). (Este campo é adicionado automaticamente ao seu relatório se você configurou o relatório para incluir todas as versões dos seus objetos).
- **Tamanho:** o tamanho do objeto em bytes, sem incluir o tamanho de uploads incompletos de várias partes, metadados de objetos e marcadores de exclusão.
- **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 (ETag) 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.
- **Classe de armazenamento:** a classe de armazenamento usada para armazenar o objeto. Defina como `STANDARD`, `REDUCED_REDUNDANCY`, `STANDARD_IA`, `ONEZONE_IA`, `INTELLIGENT_TIERING`, `GLACIER`, `DEEP_ARCHIVE`, `OUTPOSTS`, `GLACIER_IR` ou `SNOW`. Para ter mais informações, consulte [Uso de classes de armazenamento do Amazon S3](#).
- **Sinalizador de multipart upload:** definido como `True` se o objeto foi carregado como um multipart upload. Para ter mais informações, consulte [Carregar e copiar objetos usando multipart upload](#).
- **Replication status (Status da replicação):** defina como `PENDING`, `COMPLETED`, `FAILED`, ou `REPLICA`. Para ter mais informações, consulte [Obtenção de informações sobre o status da replicação](#).
- **Status da criptografia:** o status da criptografia do lado do servidor, dependendo do tipo de chave de criptografia usada: uma chave gerenciada pelo Amazon S3 (SSE-S3), uma chave do AWS Key Management Service (AWS KMS) (SSE-KMS) ou uma chave fornecida pelo cliente (SSE-C). Defina como `SSE-S3`, `SSE-C`, `SSE-KMS` ou `NOT-SSE`. Um status `NOT-SSE` significa que o objeto

não foi criptografado com a criptografia no lado do servidor. Para obter mais informações, consulte [Proteger dados com criptografia](#).

- Prazo de retenção do bloqueio de objetos do S3: a data até a qual o objeto bloqueado não pode ser excluído. Para ter mais informações, consulte [Usar o bloqueio de objetos do S3](#).
- Modo de retenção do bloqueio de objetos do S3: defina como Governance ou Compliance para objetos bloqueados. Para ter mais informações, consulte [Usar o bloqueio de objetos do S3](#).
- Status de retenção legal do bloqueio de objetos do S3: definido como On caso uma retenção legal tenha sido aplicada a um objeto. Caso contrário, ele será definido como Off. Para ter mais informações, consulte [Usar o bloqueio de objetos do S3](#).
- Nível de acesso S3 Intelligent-Tiering: nível de acesso (frequente ou infrequente) do objeto se armazenado na classe de armazenamento S3 Intelligent-Tiering. Defina como FREQUENT, INFREQUENT, ARCHIVE_INSTANT_ACCESS, ARCHIVE ou DEEP_ARCHIVE. Para ter mais informações, consulte [Classe de armazenamento para otimizar automaticamente dados com padrões de acesso alterados ou desconhecidos](#).
- S3 Bucket Key Status (Status da chave do bucket do S3): defina como ENABLED ou DISABLED. Indica se o objeto usa a uma chave de bucket do S3 para SSE-KMS. Para ter mais informações, consulte [Uso de chaves de bucket do Amazon S3](#).
- Algoritmo de soma de verificação: indica o algoritmo usado para criar a soma de verificação do objeto.
- Lista de controle de acesso de objetos: uma lista de controle de acesso (ACL) para cada objeto que define quais Contas da AWS ou grupos recebem acesso a esse objeto e o tipo de acesso concedido. O campo ACL de objetos é definido no formato JSON. Um relatório da funcionalidade Inventário S3 inclui ACLs associadas a objetos no bucket de origem, mesmo quando as ACLs estão desabilitadas para o bucket. Para obter mais informações, consulte [Trabalhar com o campo ACL de objetos](#) e [Visão geral da lista de controle de acesso \(ACL\)](#).

Note

O campo ACL de objetos é definido no formato JSON. Um relatório de inventário exibe o valor do campo ACL de objetos como uma string codificada em base64.

Por exemplo, suponha que você tenha o seguinte campo ACL de objetos no formato JSON:

```
{
  "version": "2022-11-10",
  "status": "AVAILABLE",
```

```
"grants": [{
  "canonicalId": "example-canonical-user-ID",
  "type": "CanonicalUser",
  "permission": "READ"
}]
}
```

O campo ACL de objetos é codificado e mostrado como a seguinte string codificada em base64:

```
eyJ2ZXJzaW9uIjoimjAyMi0xMS0xMCIzInN0YXR1cyI6IkFWQUlMQUMRSIsImdyYW50cyI6W3siY2Fub25pY2Fs
```

Para obter o valor decodificado em JSON para o campo ACL de objetos, você pode consultar esse campo no Amazon Athena. Para obter exemplos de consultas, consulte [Consulta do Amazon S3 Inventory com o Amazon Athena](#).

- Proprietário do objeto: o dono do objeto.

Note

Quando um objeto atinge o fim de seu ciclo de vida, com base em sua configuração de ciclo de vida, o Amazon S3 o coloca em uma fila para remoção e o remove assincronamente. Assim, pode haver um atraso entre a data de expiração e a data em que o Amazon S3 remove um objeto. O relatório de inventário inclui os objetos que expiraram, mas ainda não foram removidos. Para obter mais informações sobre as ações de expiração no ciclo de vida do S3, consulte [Expirando objetos](#).

Recomendamos que você crie uma política de ciclo de vida que exclua listas de inventário antigas. Para ter mais informações, consulte [Gerenciando seu ciclo de vida de armazenamento](#).

Com `s3:PutInventoryConfiguration`, o usuário tem permissão para selecionar todos os campos de metadados listados anteriormente para cada objeto ao configurar uma lista de inventário e especificar o bucket de destino para armazenar o inventário. Um usuário com acesso de leitura aos objetos no bucket de destino pode acessar todos os campos de metadados do objeto que estão disponíveis na lista de inventário. Para restringir o acesso a um relatório de inventário, consulte [Conceder permissões para o Inventário do S3 e análises do S3](#).

Consistência de inventário

Todos os seus objetos podem não aparecer em cada lista de inventários. A lista de inventários fornece uma consistência eventual para solicitações PUT (de objetos novos e substituídos) e para solicitações DELETE. Cada lista de inventário de um bucket é um snapshot dos itens do bucket. Essas listas acabam sendo consistentes (ou seja, uma lista pode não incluir objetos adicionados ou excluídos recentemente).

Para validar o estado de um objeto antes de você realizar uma ação no objeto, recomendamos que faça uma solicitação `HeadObject` 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 [HeadObject](#) na Referência da API 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](#)
- [Configuração de notificações de eventos do Amazon S3 para conclusão de inventário](#)
- [Localização de lista de inventário](#)
- [Consulta do Amazon S3 Inventory com o Amazon Athena](#)
- [Converter strings de ID de versão vazias nos relatórios do Amazon S3 Inventory em cadeias de caracteres null](#)
- [Trabalhar com o campo ACL de objetos](#)

Configurar o inventário do Amazon S3

O Amazon S3 Inventory oferece uma lista de arquivos sem formatação de seus objetos e metadados, de acordo com um agendamento que você define. Você pode usar o Amazon S3 Inventory como uma alternativa agendada para a operação `List` da API síncrona. O S3 Inventory fornece arquivos de saída no formato de valores separados por vírgulas (CSV), [em colunas de linhas otimizados para Apache \(ORC\)](#) ou [Apache Parquet \(Parquet\)](#), que listam seus objetos e os metadados correspondentes.

É possível configurar o S3 Inventory para criar listas de inventário, diária ou semanalmente, para um bucket do S3 ou para os objetos que compartilham um prefixo (objetos com nomes que começam com a mesma string). Para ter mais informações, consulte [Inventário do Amazon S3](#).

Esta seção descreve como configurar um inventário, incluindo detalhes sobre seus buckets de origem e destino do inventário.

Tópicos

- [Visão geral](#)
- [Criação de uma política de bucket de destino](#)
- [Conceder permissão para que o Amazon S3 use a sua chave gerenciada pelo cliente para criptografia](#)
- [Configurar um inventário usando o console do S3](#)
- [Usar a API REST para trabalhar com o Inventário S3](#)

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 console do Amazon S3, mas você também pode usar a API REST do Amazon S3, a AWS Command Line Interface (AWS CLI) ou SDKs da AWS. 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 que conceda permissões ao Amazon S3 para gravar objetos no bucket no local definido. Para ver um exemplo de política, consulte [Conceder permissões para o Inventário do S3 e análises do S3](#).


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 se devem ser listadas todas as versões de objetos ou somente as versões atuais e quais metadados de objetos devem ser incluídos.

Alguns campos de metadados de objetos nas configurações de relatório do Inventário S3 são opcionais, o que significa que estão disponíveis por padrão, mas podem ser restritos quando você concede a permissão `s3:PutInventoryConfiguration` a um usuário. É possível controlar se os usuários podem incluir esses campos de metadados opcionais em seus relatórios usando a chave de condição `s3:InventoryAccessibleOptionalFields`.

Consulte mais informações sobre os campos de metadados opcionais disponíveis no Inventário S3 em [OptionalFields](#) na Referência de API do Amazon Simple Storage Service. Consulte mais informações sobre como restringir o acesso a certos campos de metadados opcionais em uma configuração de inventário em [Controlar a criação da configuração de relatórios do Inventário S3](#).

É possível especificar que o arquivo da lista de inventário seja criptografado usando a criptografia do lado do servidor com uma chave gerenciada do Amazon S3 (SSE-S3) ou uma chave gerenciada pelo cliente do AWS Key Management Service (AWS KMS) (SSE-KMS).

 Note

A Chave gerenciada pela AWS (`aws/s3`) não é compatível com a criptografia SSE-KMS com o S3 Inventory.

Para obter mais informações sobre SSE-S3 e SSE-KMS, consulte [Proteger os dados usando criptografia do lado do servidor](#). 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 um inventário usando o console do S3](#).
 - Para usar a API do Amazon S3 para configurar uma lista de inventários, use a operação da API REST [PutBucketInventoryConfiguration](#) ou o equivalente pela AWS CLI ou por 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 console do Amazon S3, a API REST do Amazon S3 a AWS CLI ou SDKs da AWS. 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 ter mais informações, consulte [Conceder permissão para que o Amazon S3 use a sua chave gerenciada pelo cliente para criptografia](#).

O bucket de destino que armazena o arquivo da lista de inventário pode ser de propriedade de uma Conta da AWS diferente da conta que possui o bucket de origem. Se você usa a criptografia SSE-KMS para as operações entre contas do Inventário Amazon S3, recomendamos usar um ARN de chave do KMS totalmente qualificado ao configurar o Inventário S3.

Para obter mais informações, consulte [Usar criptografia SSE-KMS para operações entre contas](#) e [ServerSideEncryptionByDefault](#) na Referência da API do Amazon Simple Storage Service.

Criação de uma política de bucket de destino

Se você criar a configuração de inventário pelo console do Amazon S3, o Amazon S3 criará automaticamente uma política de bucket no bucket de destino que concede ao Amazon S3 permissão de gravação para o bucket. No entanto, se você criar a configuração de inventário pela AWS CLI, por SDKs da AWS ou pela API REST do Amazon S3, será necessário adicionar manualmente uma política de bucket no bucket de destino. Para ter mais informações, consulte [Conceder permissões para o Inventário do S3 e análises do S3](#). A política do bucket de destino do Inventário S3 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 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 outra conta que não seja a 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 sua chave gerenciada pelo cliente para criptografia

Para conceder permissão para que o Amazon S3 use sua chave gerenciada pelo cliente do AWS Key Management Service (AWS KMS) para criptografia no lado do servidor, é necessário usar uma política de chave. Para atualizar sua política de chaves para que você possa usar uma chave gerenciada pelo cliente, siga este procedimento.

Como conceder permissões ao Amazon S3 para criptografia usando a chave gerenciada pelo cliente

1. Usando a Conta da AWS 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 Chaves gerenciadas pelo cliente, selecione a chave gerenciada pelo cliente que você deseja usar para criptografar os arquivos de inventário.
6. Na seção Key Policy (Política de chaves), selecione Switch to policy view (Alternar para a visualização da política).
7. Para atualizar a política de chave, escolha Edit (Editar).
8. Na página Editar a política de chaves, adicione as seguintes linhas à política de chave existente. Para *source-account-id* e *DOC-EXAMPLE-SOURCE-BUCKET*, forneça os valores apropriados para seu caso de uso.

```
{
  "Sid": "Allow Amazon S3 use of the customer managed key",
  "Effect": "Allow",
  "Principal": {
    "Service": "s3.amazonaws.com"
  },
  "Action": [
    "kms:GenerateDataKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:SourceAccount": "source-account-id"
    }
  }
}
```

```
    },  
    "ArnLike":{  
      "aws:SourceARN": "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET"  
    }  
  }  
}
```

9. Escolha Salvar alterações.

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:

- [Managing keys](#)
- [Key policies in AWS KMS](#)

Configurar um inventário usando o console do S3

Use estas instruções para configurar o inventário usando o console do S3.

Note

Pode levar até 48 horas para o Amazon S3 entregar o primeiro relatório de inventário.


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. Na lista Buckets, escolha o nome do bucket para o qual deseja configurar o Inventário Amazon S3.
3. Escolha a guia Management.
4. Em Inventory configurations (Configurações de inventário), escolha Create inventory configuration (Criar configuração de inventário).
5. Em Nome da configuração do inventário, insira um nome.
6. Em Escopo de inventário, faça o seguinte:
 - Insira um prefixo opcional.
 - Escolha quais versões de objeto incluir, Somente versões atuais ou 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 Destino, selecione o bucket de destino no qual deseja salvar os relatórios de inventário.

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. Ao especificar o bucket de destino, você também pode incluir um prefixo opcional para agrupar os relatórios de inventário.

No campo de bucket Destino, você verá a declaração 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 ter mais informações, consulte [Criação de uma política de bucket de destino](#).

9. Em Frequência, escolha a frequência com que o relatório será gerado, Diário ou Semanal.
10. Em Formato da saída, escolha um dos seguintes formatos para o relatório:
 - CSV: se você planeja usar esse relatório de inventário com as Operações em Lote do S3 ou se quiser analisar esse relatório em outra ferramenta, como o Microsoft Excel, escolha CSV.
 - Apache ORC
 - Apache Parquet
11. Em Status, escolha Enable (Ativar) ou Disable (Desabilitar).
12. Para configurar a criptografia no lado do servidor, em Criptografia de relatório do Inventory, siga estas etapas:
 - a. Em Criptografia do lado do servidor, selecione Não especificar uma chave de criptografia ou Especificar uma chave de criptografia para criptografar dados.
 - Para manter as configurações de bucket para a criptografia padrão de objetos no lado do servidor ao armazená-los no Amazon S3, selecione Não especificar uma chave de criptografia. Desde que o destino do bucket tenha o recurso de chaves de bucket do S3 ativado, a operação de cópia aplicará uma chave de bucket do S3 ao bucket de destino.


 Note

Se a política de bucket para o destino especificado exigir que os objetos sejam criptografados antes de armazená-los no Amazon S3, você deverá escolher

Especificar uma chave de criptografia. Caso contrário, ocorrerá uma falha na cópia de objetos no destino.


- Para criptografar objetos antes de armazená-los no Amazon S3, selecione Especificar uma chave de criptografia.
- b. Se você escolheu Especificar uma chave de criptografia, em Tipo de criptografia, é necessário selecionar Chave gerenciada pelo Amazon S3 (SSE-S3) ou Chave do AWS Key Management Service (SSE-KMS).

A SSE-S3 usa uma das criptografias de bloco mais fortes: o padrão de criptografia avançada de 256 bits (AES-256) para criptografar cada objeto. A SSE-KMS oferece mais controle sobre sua chave. Para ter mais informações sobre SSE-S3, consulte [Usar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#). Para obter mais informações sobre SSE-KMS, consulte [Usar criptografia do lado do servidor com o AWS KMS \(SSE-KMS\)](#).

 Note


Para criptografar o arquivo da lista de inventário com a SSE-KMS, conceda permissão para que o Amazon S3 use a chave gerenciada pelo cliente. Para obter instruções, consulte [Conceder permissão para o Amazon S3 criptografar usando chaves do KMS](#).

- c. Se você escolheu chave do AWS Key Management Service (SSE-KMS), em AWS KMS key, poderá especificar a chave do AWS KMS por meio de uma das opções a seguir.

 Note

Se o bucket de destino que armazena o arquivo da lista de inventários pertencer a outra Conta da AWS, use um ARN de chave do KMS totalmente qualificado para especificar a chave do KMS.

- Para fazer sua escolha em uma lista de chaves do KMS disponíveis, selecione Escolha entre suas chaves do AWS KMS e escolha uma chave do KMS de criptografia simétrica na lista de chaves disponíveis. Verifique se a chave do KMS está na mesma região do bucket.

 Note

A Chave gerenciada pela AWS (aws/s3) e as chaves gerenciadas pelo cliente são exibidas na lista. No entanto, a Chave gerenciada pela AWS (aws/s3) não é compatível com a criptografia SSE-KMS com o S3 Inventory.

- Para inserir o ARN da chave do KMS, selecione Inserir ARN da chave do AWS KMS e insira o ARN da chave do KMS no campo exibido.
- Para criar uma chave gerenciada pelo cliente no console do AWS KMS, selecione Criar uma chave do KMS.

13. Em Campos adicionais de metadados, selecione um ou mais dos seguintes campos para adicionar ao relatório de inventário:

- Tamanho: o tamanho do objeto em bytes, sem incluir o tamanho de uploads incompletos de várias partes, metadados de objetos e marcadores de exclusão.
- Last modified date (Data da última modificação) – a data de criação do objeto ou data da última modificação, a mais atual entre as duas.
- Multipart upload – especifica se o objeto foi carregado como um multipart upload. Para obter mais informações, consulte [Carregar e copiar objetos usando multipart upload](#).
- Replication status (Status da replicação) – o status de replicação do objeto. Para ter mais informações, consulte [Obtenção de informações sobre o status da replicação](#).
- Status da criptografia: o tipo da criptografia do lado do servidor usada para criptografar o objeto. Para ter mais informações, consulte [Proteger os dados usando criptografia do lado do servidor](#).
- Status da chave de bucket: indica se uma chave no nível do bucket gerada pelo AWS KMS se aplica ao objeto. Para ter mais informações, consulte [Redução do custo do SSE-KMS com chaves de bucket do Amazon S3](#).
- Lista de controle de acesso de objetos: uma lista de controle de acesso (ACL) para cada objeto que define quais Contas da AWS ou grupos recebem acesso a esse objeto e o tipo de acesso concedido. Para obter mais informações sobre esse campo, consulte [Trabalhar com o campo ACL de objetos](#). Para obter mais informações sobre ACLs, consulte [Visão geral da lista de controle de acesso \(ACL\)](#).
- Proprietário do objeto: o dono do objeto.
- Classe de armazenamento: a classe de armazenamento usada para armazenar o objeto.

- **Intelligent Tiering: nível de acesso:** indica o nível de acesso (frequente ou infrequente) do objeto se armazenado na classe de armazenamento S3 Intelligent-Tiering. Para ter mais informações, consulte [Classe de armazenamento para otimizar automaticamente dados com padrões de acesso alterados ou desconhecidos](#).
- **ETag:** a tag de entidade (ETag) é um hash do objeto. O ETag reflete as alterações apenas no conteúdo de um objeto, não em seus metadados. O ETag pode ou não ser um digest MD5 do objeto de dados. Tudo depende de como o objeto foi criado e de como está criptografado. Para obter mais informações, consulte [Object](#) na Referência da API do Amazon Simple Storage Service.
- **Algoritmo de soma de verificação:** indica o algoritmo usado para criar a soma de verificação do objeto.
- **Todas as configurações de bloqueio de objetos:** o status do bloqueio de objetos, incluindo as seguintes configurações:
 - **Bloqueio de objetos: modo de retenção:** o nível de proteção aplicado ao objeto, Governança ou Conformidade.
 - **Bloqueio de objetos: prazo de retenção:** a data até a qual o objeto bloqueado não pode ser excluído.
 - **Bloqueio de objetos: status da retenção jurídica:** o status de retenção jurídica do objeto bloqueado.

Para obter mais informações sobre bloqueio de objetos do S3, consulte [Como o bloqueio de objetos do S3 funciona](#).

Para obter mais informações sobre o conteúdo de um relatório de inventário, consulte [Lista do Amazon S3 Inventory](#).

Consulte mais informações sobre como restringir o acesso a certos campos de metadados opcionais em uma configuração de inventário em [Controlar a criação da configuração de relatórios do Inventário S3](#).

14. Escolha Criar.

Quando uma lista de inventários é publicada, é possível consultar o arquivo de lista de inventários no Amazon S3 Select. Para obter mais informações sobre como localizar sua lista de inventários e consultar o arquivo da lista de inventários com o Amazon S3 Select, consulte [Localização de lista de inventário](#).

Usar a API REST para trabalhar com o Inventário S3

Veja a seguir as operações REST que você pode usar para trabalhar com o Inventário Amazon S3.

- [DeleteBucketInventoryConfiguration](#)
- [GetBucketInventoryConfiguration](#)
- [ListBucketInventoryConfigurations](#)
- [PutBucketInventoryConfiguration](#)

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 ter mais informações, consulte [Notificações de eventos do Amazon S3](#).

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>
  </QueueConfiguration>
</NotificationConfiguration>
```

```
</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
```

- *destination-prefix* é o prefixo do nome da chave do objeto que é especificado opcionalmente na configuração do inventário. É possível usar esse prefixo para agrupar todos os arquivos da lista de inventários em um local comum no bucket de destino.
- *source-bucket* é o bucket de origem ao qual a lista de inventário se refere. O nome do bucket de origem é adicionado para evitar colisões quando vários relatórios do inventário de diferentes buckets de origem são enviados ao mesmo bucket de destino.
- O *config-ID* 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 durante a configuração.
- *YYYY-MM-DDTHH-MMZ* é o carimbo de data/hora que consiste na data e hora de início em que o processo de geração do relatório de inventário começa a verificar o bucket. Por exemplo, 2016-11-06T21-32Z.
- *manifest.json* é o arquivo manifesto.
- *manifest.checksum* é o hash MD5 do conteúdo do arquivo *manifest.json*.
- *symlink.txt* é o arquivo de 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 do nome da chave do objeto que é especificado opcionalmente na configuração do inventário. É possível usar esse prefixo para agrupar todos os arquivos da lista de inventários em um local comum no bucket de destino.
- *source-bucket* é o bucket de origem ao qual a lista de inventário se refere. O nome do bucket de origem 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`.

É possível consultar um arquivo de lista de inventário com o Amazon S3 Select. No console do Amazon S3, escolha o nome da lista de inventário (por exemplo, *prefixo-de-destino/bucket-de-origem/ID-de-config/dados/nome-do-arquivo-de-exemplo.csv.gz*). Depois, escolha Ações de objeto e Consultar com o S3 Select. Para ver um exemplo de como usar uma função agregada do S3 Select para consultar um arquivo de lista de inventário, consulte [Exemplo de SUM](#)

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:

- O nome do bucket de origem
- O nome do bucket de destino
- A versão do inventário

- O carimbo de data/hora de criação no formato de data de referência (epoch) que consiste na hora de início e na data em que o processo de geração do relatório de inventário começa a verificar o bucket
- O formato e o esquema de arquivos de inventário
- Uma 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 hash 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 nos formatos 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, ObjectAccessControlList, ObjectOwner",
  "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>"
  "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); optional
binary object_access_control_list (UTF8); optional binary object_owner (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 permitindo que o Hive detecte automaticamente arquivos de inventário e os arquivos de dados associados. O manifesto compatível com o Hive funciona com os serviços compatíveis com o Hive, o Athena e o Amazon Redshift Spectrum. Ele também funciona com aplicações compatíveis com o Hive, incluindo [Presto](#), [Apache Hive](#), [Apache Spark](#) e muitos outros.

Important

O arquivo manifesto `symlink.txt` compatível com Apache Hive atualmente não funciona com o AWS Glue.

Ler o arquivo `symlink.txt` com [Apache Hive](#) e [Apache Spark](#) não é compatível com arquivos de inventário no formato Parquet e ORC.

Consulta do Amazon S3 Inventory com o Amazon Athena

É possível consultar arquivos do Inventário Amazon S3 com consultas SQL padrão usando 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 Inventário Amazon S3 no formato [colunar de linhas otimizado \(ORC\) do Apache](#), [Apache Parquet](#) ou valores separados por vírgula (CSV). Quando você usar o Athena para consultar arquivos de inventário, 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 Inventário Amazon S3 estão disponíveis em todas as Regiões da AWS.

Para usar o Athena para consultar arquivos do Inventário Amazon S3

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.
2. Crie sua consulta usando um dos exemplos de modelos de consulta a seguir, dependendo se você está consultando um relatório de inventário no formato ORC, Parquet ou CSV.
 - Ao usar o Athena para consultar um relatório de inventário no formato ORC, use o exemplo de consulta a seguir como modelo.

O exemplo de consulta a seguir inclui todos os campos opcionais no relatório de inventário no formato ORC.

Para usar esse exemplo de consulta, faça o seguinte:

- Substitua *your_table_name* pelo nome da tabela do Athena que você criou.
- Remova todos os campos opcionais que você não selecionou no inventário para que a consulta corresponda aos campos escolhidos.
- Substitua o nome do bucket a seguir e o local do inventário (o ID de configuração) conforme apropriado para sua configuração.

```
s3://DOC-EXAMPLE-BUCKET/config-ID/hive/
```

- Substitua a data *2022-01-01-00-00* abaixo de `projection.dt.range` com o primeiro dia do intervalo de tempo em que você particiona os dados no Athena. Para obter mais informações, consulte [Particionamento de dados no Athena](#).

```
CREATE EXTERNAL TABLE your_table_name(  
    bucket string,  
    key string,  
    version_id string,  
    is_latest boolean,  
    is_delete_marker boolean,  
    size bigint,  
    last_modified_date timestamp,  
    e_tag string,  
    storage_class string,  
    is_multipart_uploaded boolean,
```



```

        replication_status string,
        encryption_status string,
        object_lock_retain_until_date bigint,
        object_lock_mode string,
        object_lock_legal_hold_status string,
        intelligent_tiering_access_tier string,
        bucket_key_status string,
        checksum_algorithm string,
        object_access_control_list string,
        object_owner string
    ) PARTITIONED BY (
        dt string
    )
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
  STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
  OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat'
  LOCATION 's3://source-bucket/config-ID/hive/'
  TBLPROPERTIES (
    "projection.enabled" = "true",
    "projection.dt.type" = "date",
    "projection.dt.format" = "yyyy-MM-dd-HH-mm",
    "projection.dt.range" = "2022-01-01-00-00,NOW",
    "projection.dt.interval" = "1",
    "projection.dt.interval.unit" = "HOURS"
  );

```

- Ao usar o Athena para consultar um relatório de inventário no formato Parquet, use a consulta de amostra para um relatório no formato ORC. No entanto, use o seguinte Parquet SerDe no lugar do ORC SerDe na declaração 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 exemplo de consulta a seguir como modelo.

O exemplo de consulta a seguir inclui todos os campos opcionais no relatório de inventário no formato CSV.

Para usar esse exemplo de consulta, faça o seguinte:

- Substitua *your_table_name* pelo nome da tabela do Athena que você criou.
- Remova todos os campos opcionais que você não selecionou no inventário para que a consulta corresponda aos campos escolhidos.

- Substitua o nome do bucket a seguir e o local do inventário (o ID de configuração) conforme apropriado para sua configuração.

`s3://DOC-EXAMPLE-BUCKET/config-ID/hive/`

- Substitua a data `2022-01-01-00-00` abaixo de `projection.dt.range` com o primeiro dia do intervalo de tempo em que você particiona os dados no Athena. Para obter mais informações, consulte [Particionamento de dados no Athena](#).

```
CREATE EXTERNAL TABLE your_table_name(
    bucket string,
    key string,
    version_id string,
    is_latest boolean,
    is_delete_marker boolean,
    size string,
    last_modified_date string,
    e_tag string,
    storage_class string,
    is_multipart_uploaded boolean,
    replication_status string,
    encryption_status string,
    object_lock_retain_until_date string,
    object_lock_mode string,
    object_lock_legal_hold_status string,
    intelligent_tiering_access_tier string,
    bucket_key_status string,
    checksum_algorithm string,
    object_access_control_list string,
    object_owner string
) PARTITIONED BY (
    dt string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
STORED AS INPUTFORMAT 'org.apache.hadoop.hive.q1.io.SymlinkTextInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.q1.io.IgnoreKeyTextOutputFormat'
LOCATION 's3://source-bucket/config-ID/hive/'
TBLPROPERTIES (
    "projection.enabled" = "true",
    "projection.dt.type" = "date",
    "projection.dt.format" = "yyyy-MM-dd-HH-mm",
    "projection.dt.range" = "2022-01-01-00-00,NOW",
    "projection.dt.interval" = "1",
```

```
"projection.dt.interval.unit" = "HOURS"
);
```

3. Agora você pode executar várias consultas no inventário, conforme mostrado nos exemplos a seguir. Substitua cada *user input placeholder* por suas próprias informações.

```
# Get a list of the latest inventory report dates available.
SELECT DISTINCT dt FROM your_table_name ORDER BY 1 DESC limit 10;

# Get the encryption status for a provided report date.
SELECT encryption_status, count(*) FROM your_table_name WHERE dt = 'YYYY-MM-DD-HH-MM' GROUP BY encryption_status;

# Get the encryption status for inventory report dates in the provided range.
SELECT dt, encryption_status, count(*) FROM your_table_name
WHERE dt > 'YYYY-MM-DD-HH-MM' AND dt < 'YYYY-MM-DD-HH-MM' GROUP BY dt,
encryption_status;
```

Quando você configura o Inventário S3 para adicionar o campo Lista de controle de acesso de objetos (ACL de objetos) a um relatório de inventário, o relatório exibe o valor do campo ACL de objetos como uma string codificada em base64. Para obter o valor decodificado em JSON para o campo ACL de objetos, você pode consultar esse campo usando o Athena. Veja os exemplos de consulta a seguir. Para obter mais informações sobre o campo ACL do objeto, consulte [Trabalhar com o campo ACL de objetos](#).

```
# Get the S3 keys that have Object ACL grants with public access.
WITH grants AS (
  SELECT key,
    CAST(
      json_extract(from_utf8(from_base64(object_access_control_list)),
        '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))
    ) AS grants_array
  FROM your_table_name
)
SELECT key,
  grants_array,
  grant
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE element_at(grant, 'uri') = 'http://acs.amazonaws.com/groups/global/AllUsers'
```

```
# Get the S3 keys that have Object ACL grantees in addition to the object owner.
WITH grants AS
  (SELECT key,
    from_utf8(from_base64(object_access_control_list)) AS
  object_access_control_list,
    object_owner,
    CAST(json_extract(from_utf8(from_base64(object_access_control_list)),
    '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))) AS grants_array
  FROM your_table_name)
SELECT key,
  grant,
  objectowner
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE cardinality(grants_array) > 1 AND element_at(grant, 'canonicalId') !=
  object_owner;
```

```
# Get the S3 keys with READ permission that is granted in the Object ACL.
WITH grants AS (
  SELECT key,
    CAST(
      json_extract(from_utf8(from_base64(object_access_control_list)),
    '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))
    ) AS grants_array
  FROM your_table_name
)
SELECT key,
  grants_array,
  grant
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE element_at(grant, 'permission') = 'READ';
```

```
# Get the S3 keys that have Object ACL grants to a specific canonical user ID.
WITH grants AS (
  SELECT key,
    CAST(
      json_extract(from_utf8(from_base64(object_access_control_list)),
    '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))
    ) AS grants_array
  FROM your_table_name
```

```
)  
SELECT key,  
       grants_array,  
       grant  
FROM grants, UNNEST(grants_array) AS t(grant)  
WHERE element_at(grant, 'canonicalId') = 'user-canonical-id';
```

```
# Get the number of grantees on the Object ACL.  
SELECT key,  
       object_access_control_list,  
       json_array_length(json_extract(object_access_control_list, '$.grants')) AS  
       grants_count  
FROM your_table_name;
```

Para obter mais informações sobre como usar o Athena, consulte o [Guia do usuário do Amazon Athena](#).

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](#).
 - 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.q1.io.SymlinkTextInputFormat' OUTPUTFORMAT
'org.apache.hadoop.hive.q1.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  
$outs = New-Object System.IO.StreamWriter File-location\temp.csv
```

```
try {
    $skip = 0
    while ( !$ins.EndOfStream ) {
        $line = $ins.ReadLine();
        if ( $skip -ne 0 ) {
            $outs.WriteLine($line);
        } else {
            $skip = 1
        }
    }
} finally {
    $outs.Close();
    $ins.Close();
}
Move-Item File-location\temp.csv File-location\file.csv -Force
```

10. 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](#).

Trabalhar com o campo ACL de objetos

Um relatório do Inventário Amazon S3 contém uma lista dos objetos no bucket de origem do S3 e os metadados de cada objeto. O campo lista de controle de acesso (ACL) de objetos é um campo de metadados que está disponível no Inventário Amazon S3. Especificamente, o campo ACL de objetos contém a lista de controle de acesso (ACL) para cada objeto. A ACL de um objeto define quais Contas da AWS ou grupos recebem acesso a esse objeto e ao tipo de acesso concedido. Para obter mais informações, consulte [Visão geral da lista de controle de acesso \(ACL\)](#) e [Lista do Amazon S3 Inventory](#).

O campo ACL de objetos nos relatórios do Inventário Amazon S3 é definido no formato JSON. Os dados JSON incluem os seguintes campos:

- **version**: a versão do formato do campo ACL de objetos nos relatórios de inventário. Está no formato de data yyyy-mm-dd.


- **status**: os valores possíveis são `AVAILABLE` ou `UNAVAILABLE` para indicar se uma ACL de objetos está disponível para um objeto. Quando o status da ACL de objetos é `UNAVAILABLE`, o valor do campo `Proprietário` do objeto no relatório de inventário também é `UNAVAILABLE`.
- **grants**: pares de concessão e permissão que listam o status da permissão de cada beneficiário da ACL de objetos. Os valores disponíveis para um beneficiário são `CanonicalUser` e `Group`. Para obter mais informações sobre os beneficiários, consulte [Beneficiários em listas de controle de acesso](#).

Para um beneficiário com o tipo `Group`, um par de concessão e permissão inclui os seguintes atributos:

- **uri**: um grupo predefinido do Amazon S3.
- **permission**: as permissões de ACL concedidas ao objeto. Para obter mais informações, consulte [Permissões de ACL em um objeto](#).
- **type**: o tipo `Group`, o que indica que o beneficiário é um grupo.

Para um beneficiário com o tipo `CanonicalUser`, um par de concessão e permissão inclui os seguintes atributos:

- **canonicalId**: uma forma ofuscada do ID da Conta da AWS. O ID de usuário canônico de uma Conta da AWS é específico dessa conta. Você pode recuperar o ID de usuário canônico. Para obter mais informações, consulte [Find the canonical user ID for your Conta da AWS](#) no Guia de referência de gerenciamento de contas da AWS.

 Note

Se um beneficiário em uma ACL for o endereço de e-mail de uma Conta da AWS, o Inventário S3 usa o `canonicalId` dessa Conta da AWS e o tipo `CanonicalUser` para especificar esse beneficiário. Para obter mais informações, consulte [Beneficiários em listas de controle de acesso](#).

- **permission**: as permissões de ACL concedidas ao objeto. Para obter mais informações, consulte [Permissões de ACL em um objeto](#).
- **type**: o tipo `CanonicalUser`, o que indica que o beneficiário é uma Conta da AWS.

O exemplo a seguir mostra os valores possíveis para o campo ACL de objetos no formato JSON:

```
{
```

```
"version": "2022-11-10",
"status": "AVAILABLE",
"grants": [{
  "uri": "http://acs.amazonaws.com/groups/global/AllUsers",
  "permission": "READ",
  "type": "Group"
}, {
  "canonicalId": "example-canonical-id",
  "permission": "FULL_CONTROL",
  "type": "CanonicalUser"
}]
}
```

Note

O campo ACL de objetos é definido no formato JSON. Um relatório de inventário exibe o valor do campo ACL de objetos como uma string codificada em base64.

Por exemplo, suponha que você tenha o seguinte campo ACL de objetos no formato JSON:

```
{
  "version": "2022-11-10",
  "status": "AVAILABLE",
  "grants": [{
    "canonicalId": "example-canonical-user-ID",
    "type": "CanonicalUser",
    "permission": "READ"
  }]
}
```

O campo ACL de objetos é codificado e mostrado como a seguinte string codificada em base64:

```
eyJ2ZXJzaW9uIjoiaWoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IktFWQU1MQUMRSIsImdyYW50cyI6I3siY2Fub25pY2FsSw
```

Para obter o valor decodificado em JSON para o campo ACL de objetos, você pode consultar esse campo no Amazon Athena. Para obter exemplos de consultas, consulte [Consulta do Amazon S3 Inventory com o Amazon Athena](#).

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 Replicação Entre Regiões Diferentes (CRR). 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](#).

Para habilitar 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 ter mais informações, consulte [Configurações de replicação adicionais](#).

Para obter métricas detalhadas para a Replicação do S3, incluindo métricas de contagem de regras de replicação, você pode usar a Lente de Armazenamento do Amazon S3. A Lente de Armazenamento do S3 é um recurso de análise de armazenamento em nuvem que você pode usar para obter visibilidade em toda a organização sobre o uso e a atividade do armazenamento de objetos. Para obter mais informações, consulte [Como usar a Lente de Armazenamento do S3 para proteger seus dados](#). Para obter uma lista completa de métricas, consulte o [Glossário de métricas da Lente de Armazenamento do S3](#).

Tópicos

- [Por que usar a replicação](#)
- [Quando usar a replicação entre regiões](#)
- [Quando usar Replicação na mesma região](#)
- [Quando usar a replicação bidirecional](#)

- [Quando usar S3 Batch Replication](#)
- [Requisitos para replicação](#)
- [O que o Amazon S3 replica?](#)
- [Configuração da replicação](#)
- [Replicar objetos existentes com o S3 Batch Replication](#)
- [Configurações de replicação adicionais](#)
- [Obtenção de informações sobre o status da replicação](#)
- [Considerações adicionais](#)

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 configurações de ciclo de vida nos buckets 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”](#).

- 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](#).
- Replicar objetos e fazer failover para um bucket em outra Região da AWS: para manter todos os metadados e objetos sincronizados entre os buckets durante a replicação de dados, use regras de replicação bidirecional antes de configurar os controles de failover para pontos de acesso multirregionais do Amazon S3. As regras de replicação bidirecional ajudam a garantir que, quando os dados são gravados no bucket do S3 para o qual o tráfego faz failover, esses dados sejam replicados de volta para o bucket de origem.

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](#).

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 a replicação bidirecional

- Criar conjuntos de dados compartilhados entre várias Regiões da AWS: com a sincronização de modificações da réplica, você pode replicar facilmente alterações de metadados, como listas de controle de acesso (ACLs) de objetos, etiquetas de objetos ou bloqueios de objetos, em objetos de replicação. Essa replicação bidirecional é importante se você quiser manter todos os objetos e alterações nos metadados dos objetos em sincronia. Você pode [ativar a sincronização de modificações da réplica](#) em uma regra de replicação nova ou existente ao realizar a replicação bidirecional entre dois ou mais buckets na mesma região ou em Regiões da AWS diferentes.
- Manter os dados sincronizados entre regiões durante o failover: você pode sincronizar dados em buckets entre Regiões da AWS configurando regras de replicação bidirecional com a Replicação do S3 Entre Regiões Diferentes (CRR) diretamente de um ponto de acesso multirregional. Para tomar uma decisão informada sobre quando iniciar um failover, você também pode ativar as métricas de replicação do S3 para monitorar a replicação no Amazon CloudWatch, no Controle de Tempo de Replicação do S3 (S3 RTC) ou no ponto de acesso multirregional.
- Tornar a aplicação altamente disponível: mesmo no caso de uma interrupção no tráfego regional, você pode usar regras de replicação bidirecional para manter todos os metadados e objetos sincronizados entre os buckets durante a replicação de dados.

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 habilitar ou desabilitar uma Região da AWS, consulte [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](#).
- 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](#).
- 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 \(ACL\)](#).

- 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 habilitar a replicação em um bucket que tenha a funcionalidade Bloqueio de Objetos habilitada, você deve usar a AWS Command Line Interface, a API REST ou os AWS SDKs. Para obter mais informações gerais, consulte [Usar o bloqueio de objetos do S3](#).

Note

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 ter mais informações, consulte [Configuração de permissões](#).

Para ter mais informações, consulte [Configuração da replicação](#).

Se você estiver definindo a configuração de replicação em um cenário entre contas, no qual 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](#).
- 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](#).

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?](#)
- [O que não é replicado com as configurações de replicação?](#)
- [Criptografia e replicação de bucket padrão](#)

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 usando chaves fornecidas pelo cliente (SSE-C), objetos criptografados em repouso por uma chave gerenciada pelo Amazon S3 (SSE-S3) ou uma chave do KMS armazenada no AWS Key Management Service (SSE-KMS). Para ter mais informações, consulte [the section called “Replicar objetos criptografados \(SSE-C, SSE-S3, SSE-KMS, DSSE-KMS\)”](#).
- 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](#).
- 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](#).

- 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](#).

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 informações de retenção aplicadas, ele aplica esse mesmo controles de retenção à 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](#).

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](#).
 - 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 de versão do objeto para excluir 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](#).

- 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](#).

- 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, copie os objetos de origem no local com um trabalho de cópia em lote. Copiar esses objetos no local criará versões dos objetos 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](#).
- 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](#).

- Objetos armazenados nas classes ou nos níveis de armazenamento S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, S3 Intelligent-Tiering Archive Access ou S3 Intelligent-Tiering Deep Archive Access. Não é possível replicar esses objetos enquanto eles não forem restaurados e copiados para uma classe de armazenamento diferente.

Para saber mais sobre as classes S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive, consulte [Classes de armazenamento para arquivamento de objetos](#).

Para saber mais sobre a classe S3 Intelligent-Tiering, consulte [Amazon S3 Intelligent-Tiering](#).

- 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](#).

- 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 nos buckets de origem e de destino.

- Ações realizadas pela configuração do ciclo de vida.

Por exemplo, se a configuração de ciclo de vida estiver habilitada apenas no seu bucket de origem, o Amazon S3 criará marcadores de exclusão para objetos expirados, mas não replicará esses marcadores. 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](#).

Criptografia e replicação de bucket padrão

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. Como resultado, as tags de entidade (ETags) dos objetos de origem diferem dos ETags dos objetos de réplica. Se você tiver aplicações que usam ETags, será necessário atualizar essas aplicações para considerar essa diferença.
- Se os objetos no bucket de origem forem criptografados usando a criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3), a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS) ou a criptografia de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS), os objetos de réplica no bucket de destino usarão o mesmo tipo de criptografia que os objetos de origem. As configurações de criptografia padrão do bucket de destino não são usadas.

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](#).

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 Legal/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](#)
- [Configuração de permissões](#)
- [Demonstrações: exemplos para configurar a replicação](#)

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](#).

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. Os filtros têm como alvo objetos que correspondem aos valores exatos que você especifica.

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/`. Para ter mais informações sobre prefixos de chave de objeto, consulte [Organizar objetos usando prefixos](#).

As seções a seguir fornecem informações adicionais.

Tópicos

- [Configuração da regra básica](#)
- [Opcional: Especificação de um filtro](#)
- [Configurações adicionais de destino](#)
- [Exemplo de configurações de replicação](#)
- [Compatibilidade retroativa](#)

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 ter mais informações, consulte [Compatibilidade retroativa](#).

```
...
  <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>
    ...
  </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](#)
- [Adicionar vários buckets de destino](#)

- [Especificar parâmetros diferentes para cada regra de replicação com vários buckets de destino](#)
- [Alterar a propriedade da réplica](#)
- [Ativar o controle do tempo de replicação do S3](#)
- [Replicar objetos criados com criptografia do lado do servidor com o uso do AWS KMS.](#)

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>
</Rule>
```

```

    <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 ter mais informações, consulte [Alterar o proprietário da réplica](#).

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 \(S3 RTC\)](#). 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 [Replicar objetos criados com criptografia do lado do servidor \(SSE-C, SSE-S3, SSE-KMS, DSSE-KMS\)](#).

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>

```

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>
```

Exemplo 4: Demonstrações de exemplo

Para obter exemplos de demonstrações, consulte [Demonstrações: exemplos para configurar a replicação](#).

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 adiciona 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 não baseadas em etiquetas. Para ter mais informações, consulte [Replicação de marcadores de exclusão entre intervalos](#).

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](#)
- [Conceder permissões quando os buckets de origem e de destino pertencerem a Contas da AWS diferentes](#)

- [Conceder permissões para operações em lote do S3](#)
- [Alterar a propriedade da réplica](#)
- [Habilitar o recebimento de objetos replicados de um bucket de origem](#)

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](#).

- 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"
    }
  ]
}
```

- O exemplo a seguir mostra uma política de confiança na qual você identifica o Amazon S3 e o S3 Batch Operations como entidades principais de serviço. Isso é útil se você estiver criando um trabalho de replicação em lote. Para ter mais informações, consulte [Criar um trabalho de replicação em lote para uma primeira regra de replicação ou um novo destino](#).

```
{
```

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

Para obter mais informações sobre funções do IAM, consulte [Perfis 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.)
- `s3:GetObjectVersionForReplication` e `s3:GetObjectVersionAcl`: 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](#).

Note

As permissões para a ação `s3:ReplicateObject` no bucket `DOC-EXAMPLE-BUCKET2` (o bucket de destino) também permitem a replicação de metadados, como etiquetas de objeto e ACLS. Portanto, você não precisa conceder permissão explícita para a ação `s3:ReplicateTags`.

- `s3:GetObjectVersionTagging`: as permissões para essa ação em objetos no bucket *DOC-EXAMPLE-BUCKET1* permitem que o Amazon S3 leia etiquetas de objeto para replicação. Para ter mais informações, consulte [Categorizando seu armazenamento usando tags](#). Se o Amazon S3 não tiver essas permissões, ele replicará os objetos, mas não as tags do objeto.

Consulte uma lista de ações do Amazon S3 em [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço.

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 \(ACL\)](#).

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](#).

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.

Note

O formato do ARN do perfil pode parecer diferente. Se o perfil tiver sido criado usando o console, o formato do ARN será `arn:aws:iam::account-ID:role/service-role/role-name`. Se o perfil tiver sido criado usando a AWS CLI, o formato do ARN será `arn:aws:iam::account-`

ID:role/role-name. Para obter mais informações, consulte [Perfis do IAM](#) no Guia do usuário do IAM.

```
{
  "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](#).

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](#).
- 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/*"
    }
  ]
  ...
```

Conceder permissões para operações em lote do S3

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. Você pode criar um trabalho único de replicação em lote ao criar a primeira regra em uma configuração de replicação nova ou ao adicionar um novo destino a uma configuração existente por meio do AWS Management Console. Você também pode iniciar a replicação em lote para uma configuração de replicação existente criando uma tarefa de operações em lote.

Para obter exemplos de políticas e perfis do IAM de replicação em lote, consulte [Configurar políticas do IAM para replicação em lote](#).

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](#).

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](#).

- [Instalar a 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](#).

Para saber mais sobre como usar a Replicação do S3 para replicar dados, consulte [Tutorial: Replicar dados em uma região ou entre Regiões da AWS usando a Replicação do S3](#).

Tópicos

- [Configuração da replicação para buckets de origem e de destino pertencentes à mesma conta](#)
- [Configurar a replicação quando os buckets de origem e destino pertencerem a contas diferentes](#)
- [Alterar o proprietário da réplica quando os buckets de origem e de destino forem de propriedade de contas diferentes](#)
- [Replicar objetos criptografados](#)
- [Replicação de objetos com o Controle de Tempo de Replicação do S3 \(S3 RTC\)](#)
- [Gerenciamento de regras de replicação usando o console do Amazon S3](#)

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](#).

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

Para configurar uma regra de replicação quando o bucket de destino estiver na mesma Conta da AWS que o bucket de origem, siga estas etapas.


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](#).

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 desejado.
4. Selecione a guia Gerenciamento, role para baixo até Regras de replicação e selecione Criar regra de replicação.
5. Na seção Configuração da regra de replicação, em Nome da regra de replicação, insira um nome para sua regra para ajudar a identificá-la posteriormente. O nome é obrigatório e precisa ser exclusivo dentro do bucket.
6. Em Status, Enabled (Habilitado) é selecionado por padrão. Uma regra ativada começa a funcionar assim que você a salva. Se você quiser ativar a regra posteriormente, selecione 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](#).

8. Em Bucket de origem, você tem as seguintes opções para definir a origem da replicação:

- Para replicar todo o bucket, escolha Apply to all objects in the bucket (Aplicar 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 o prefixo especificado (por exemplo, pictures). Insira um prefixo na caixa Prefixo.


 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 Adicionar tag e insira o par de chave-valor 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](#).

O novo esquema XML de configuração de replicação 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](#). 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](#). O novo esquema é descrito como configuração de replicação XML V2.

9. Em 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. Atualmente, a AWS tem três partições: aws (regiões padrão), aws-cn (regiões da China) e aws-us-gov

(regiões AWS GovCloud (US)). Para solicitar um aumento da cota do bucket de destino, você pode usar [cotas de serviço](#).

- Para replicar para um bucket ou buckets em sua conta, selecione Escolher um bucket nesta conta e digite ou procure nomes de buckets de destino.
- Para replicar em um ou mais buckets em uma Conta da AWS diferente, selecione Selecionar 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 ter mais informações, consulte [Conceder permissões quando os buckets de origem e de destino pertencerem a Contas da AWS diferentes](#).

Opcionalmente, se você quiser padronizar a propriedade de novos objetos no bucket de destino, escolha 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](#).

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. Configure uma função do AWS Identity and Access Management (IAM) que o Amazon S3 possa assumir para replicar objetos em seu nome.

Para configurar um perfil do IAM, na seção Perfil do IAM, selecione uma das seguintes opções na lista suspensa de perfis do IAM:

- 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 ter mais informações, consulte [Conceder permissões ao usuário para transmitir um perfil a um AWS service \(Serviço da AWS\)](#) no Guia do usuário do IAM.

11. Para replicar objetos no bucket de origem que estão criptografados com a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS), em Criptografia, selecione Replicar objetos criptografados com o AWS KMS. Em chaves do AWS KMS 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. Para limitar a seleção de chaves do KMS, você pode selecionar um alias ou um ID de chave.

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 [Replicar objetos criados com criptografia do lado do servidor \(SSE-C, SSE-S3, SSE-KMS, DSSE-KMS\)](#).

 Important

Ao replicar objetos que estão criptografados com AWS KMS, a taxa de solicitações do AWS KMS dobra 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 uma cota de taxas de solicitação por conta de chamada por região. Para ter informações sobre os padrões de limite, consulte [Cotas 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 do que a metade do limite da taxa padrão do AWS KMS para sua conta, recomendamos que você solicite um aumento da cota 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. Consulte informações sobre como visualizar métricas do CloudWatch em [Uso do console do S3](#).

Se você escolheu replicar objetos criptografados com o AWS KMS, faça o seguinte:

- Em AWS KMS key para criptografar objetos de destino, especifique sua chave do KMS de uma das seguintes formas:
 - Para escolher em uma lista de chaves do KMS disponíveis, selecione Escolher entre suas AWS KMS keys e escolha sua chave do KMS na lista de chaves disponíveis.

As chaves Chave gerenciada pela AWS (aws/s3) e as chaves gerenciadas pelo cliente são exibidas nessa lista. Para ter mais informações sobre chaves gerenciadas pelo cliente, consulte [Chaves de clientes e chaves da AWS](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para inserir o nome do recurso da Amazon (ARN) da chave do KMS, selecione Inserir ARN da AWS KMS key e insira o ARN da chave do KMS no campo exibido. Isso criptografa as réplicas no bucket de destino. Você pode encontrar o ARN da chave do KMS no [console do IAM](#) em Chaves de criptografia.
- Para criar uma chave gerenciada pelo cliente no console do AWS KMS, selecione Criar uma chave do KMS.

Para ter mais informações sobre como criar uma AWS KMS key, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.


⚠ Important

Você só pode usar chaves do KMS habilitadas na mesma Região da AWS que o bucket. Quando você seleciona 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 Inserir o ARN da AWS KMS key e insira o ARN de sua 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 ter mais informações, consulte [Identificar chaves do KMS 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 [Usar criptografia do lado do servidor com o AWS KMS \(SSE-KMS\)](#).

12. Em Classe de armazenamento de destino, se você quiser replicar seus dados em uma classe de armazenamento específica no destino, escolha 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.
13. Você tem as seguintes opções adicionais ao definir as Opções de replicação adicionais:
 - Se você quiser ativar o Controle do Tempo de Replicação do S3 (S3 RTC) na configuração de replicação, selecione Controle do Tempo de Replicação (RTC). Para obter mais informações sobre essa opção, consulte [Atendimento aos requisitos de conformidade usando o Controle do tempo de replicação do S3 \(S3 RTC\)](#).
 - Se você quiser habilitar métricas de replicação do S3 na configuração de replicação, selecione Replication metrics and events (Métricas e eventos de replicação). Consulte mais informações em [Monitorar o progresso com métricas de replicação e Notificações de eventos do S3](#).

- Se quiser habilitar a replicação de marcadores de exclusão na configuração de replicação, selecione Delete marker replication (Excluir replicação de marcador). Consulte mais informações em [Replicação de marcadores de exclusão entre intervalos](#).
- 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). Consulte mais informações em [Replicação de alterações de metadados com sincronização de modificação de réplica do Amazon S3](#).

 Note

Quando você usa métricas de replicação do S3 RTC ou S3, aplicam-se taxas adicionais.

14. Para terminar, escolha Save (Salvar).
15. Depois de salvar sua regra, você pode editar, ativar, desativar ou excluir sua regra selecionando sua regra e escolhendo Edit rule (Editar regra).

Usando 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 fornecer 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
```

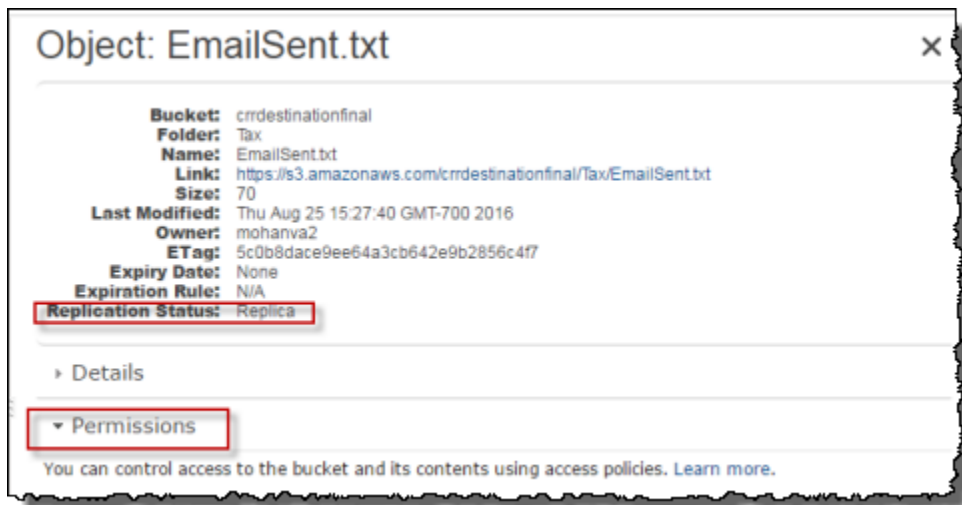
6. Teste a configuração no console do Amazon S3:
 - a. Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
 - b. No bucket de *source*, crie uma pasta denominada Tax.
 - c. Adicione exemplos de objetos à pasta Tax no bucket de *source*.

 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](#).

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](#).



Uso dos AWS SDKs

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](#).

```
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(
            "\\\"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
\\n      \\\"Resource\\\\\":[\\r\\n      ");
        permissionPolicy.append("\\\\\"arn:aws:s3:::\\");
        permissionPolicy.append(sourceBucket);
        permissionPolicy.append("/.*\\\\\\\"\\r\\n      ]\\r\\n      },\\r\\n
      {\\r\\n      ");
        permissionPolicy.append(
            "\\\"Effect\\\\\":\\\\\\\"Allow\\\\\\\",\\r\\n      \\
\\\"Action\\\\\":[\\r\\n      ");
        permissionPolicy.append(
            "\\\"s3:ListBucket\\\\\\\",\\r\\n      \\
\\\"s3:GetReplicationConfiguration\\\\\\\"\\r\\n      ");
        permissionPolicy.append("],\\r\\n      \\\"Resource\\\\\":[\\r\\n
        \\\"arn:aws:s3:::\\");
        permissionPolicy.append(sourceBucket);
        permissionPolicy.append("\\r\\n      ");
        permissionPolicy

```

```

        .append("]\\r\\n      },\\r\\n      {\\r\\n
\\\\"Effect\\":\\"Allow\\",\\r\\n
        permissionPolicy.append(
            "\\\"Action\\":[\\r\\n
\\s3:ReplicateObject\\",\\r\\n
        permissionPolicy
            .append("\\\"s3:ReplicateDelete\\",\\r\\n
\\\"s3:ReplicateTags\\",\\r\\n
        permissionPolicy.append("\\\"s3:GetObjectVersionTagging\\\"\\r\\n\\r
\\n      ],\\r\\n
        permissionPolicy.append("\\\"Resource\\":\\"arn:aws:s3::");
        permissionPolicy.append(destinationBucket);
        permissionPolicy.append("/.*\\\"\\r\\n      }\\r\\n      ]\\r\\n}");

        PutRolePolicyRequest putRolePolicyRequest = new
PutRolePolicyRequest()
            .withRoleName(roleName)
            .withPolicyDocument(permissionPolicy.toString())
            .withPolicyName("crrRolePolicy");

        iamClient.putRolePolicy(putRolePolicyRequest);
    }
}

```

C#

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](#).

```

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](#).

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](#) 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*.

Note

Para usar o exemplo a seguir, substitua os *user input placeholders* por suas próprias informações. Substitua *DOC-EXAMPLE-BUCKET* pelo nome do bucket de destino. Substitua *source-bucket-acct-ID:role/service-role/source-acct-IAM-role* pelo perfil que você está usando para essa configuração de replicação. Se você criou o perfil de serviço do IAM manualmente, defina o caminho do perfil como *role/service-role/*, conforme mostrado no exemplo de política a seguir. Para obter mais informações, consulte [ARNs do IAM](#) no Guia do usuário do IAM.

```
{
  "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](#).

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](#).

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](#).

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](#).

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](#).

Uso do console do S3

Para obter instruções detalhadas, consulte [Configuração da replicação para buckets de origem e de destino pertencentes à mesma conta](#). Este tópico traz instruções para definir a configuração da replicação quando os buckets pertencerem às mesmas Contas da AWS ou a contas diferentes.

Usando 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)

1. 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.

2. 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
```

3. 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 dos AWS SDKs

Para obter um exemplo de código para adicionar a configuração da replicação, consulte [Uso dos AWS SDKs](#). É necessário modificar a configuração de replicação de acordo. Para obter informações conceituais, consulte [Alterar o proprietário da réplica](#).

Replicar objetos criptografados

Por padrão, o Amazon S3 não replica objetos criptografados por meio da criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS) ou da criptografia de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS). Para replicar objetos criptografados com SSE-KMS ou DSSE-KMS, você deve modificar 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 ter mais informações, consulte [Replicar objetos criados com criptografia do lado do servidor \(SSE-C, SSE-S3, SSE-KMS, DSSE-KMS\)](#).

Note

Quando uma chave de bucket do S3 estiver habilitada para o bucket de origem ou de destino, o contexto de criptografia será o nome do recurso da Amazon (ARN) do bucket e

não o ARN do objeto. Você precisa atualizar suas políticas do IAM a fim de usar o ARN do bucket para o contexto de criptografia. Para ter mais informações, consulte [Chaves de bucket do S3 e replicação](#).

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](#). Este tópico traz instruções para definir a configuração da replicação quando os buckets pertencerem às mesmas Contas da AWS ou a contas diferentes.

Usando a AWS CLI

Para replicar objetos criptografados com a AWS CLI, faça o seguinte:

- Crie os buckets de origem e destino e habilite o versionamento neles.
- Crie um perfil de serviço do AWS Identity and Access Management (IAM) que dê ao Amazon S3 permissão para replicar objetos. As permissões do perfil do IAM incluem aquelas necessárias para replicar objetos criptografados.
- Adicione uma configuração de replicação ao bucket de origem. A configuração de replicação fornece informações relacionadas à replicação de objetos que são criptografados por meio das chaves do KMS.
- Adicione objetos criptografados ao bucket de origem.
- Teste a configuração para confirmar se os objetos criptografados estão sendo replicados no bucket de destino.

Os procedimentos a seguir demonstram esse processo.

Para replicar objetos criptografados no lado do servidor (AWS CLI)

1. Neste exemplo, criamos os buckets *DOC-EXAMPLE-SOURCE-BUCKET* e *DOC-EXAMPLE-DESTINATION-BUCKET* na mesma Conta da AWS. Defina também um perfil de credenciais para a AWS CLI. Este exemplo usa 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. Para usar os comandos neste exemplo, substitua *user input placeholders* pelas suas informações.

2. Use os comandos a seguir para criar o bucket *DOC-EXAMPLE-SOURCE-BUCKET* e habilitar o versionamento nele. O comando *DOC-EXAMPLE-SOURCE-BUCKET* a seguir cria um bucket na região Leste dos EUA (Norte da Virgínia) (us-east-1).

```
aws s3api create-bucket \  
--bucket DOC-EXAMPLE-SOURCE-BUCKET \  
--region us-east-1 \  
--profile acctA
```

```
aws s3api put-bucket-versioning \  
--bucket DOC-EXAMPLE-SOURCE-BUCKET \  
--versioning-configuration Status=Enabled \  
--profile acctA
```

3. Use os comandos a seguir para criar o bucket *DOC-EXAMPLE-DESTINATION-BUCKET* e habilitar o versionamento nele. Os comandos de exemplo a seguir criam o bucket *DOC-EXAMPLE-DESTINATION-BUCKET* na região Oeste dos EUA (Oregon) (us-west-2).

Note

Para configurar a replicação quando os buckets *DOC-EXAMPLE-SOURCE-BUCKET* e *DOC-EXAMPLE-DESTINATION-BUCKET* estiverem na mesma Conta da AWS, use o mesmo perfil. Neste exemplo, usamos *acctA*. Para configurar a replicação quando os buckets pertencerem a Contas da AWS distintas, especifique diferentes perfis para cada um.

```
aws s3api create-bucket \  
--bucket DOC-EXAMPLE-DESTINATION-BUCKET \  

```

```
--region us-west-2 \  
--create-bucket-configuration LocationConstraint=us-west-2 \  
--profile acctA
```

```
aws s3api put-bucket-versioning \  
--bucket DOC-EXAMPLE-DESTINATION-BUCKET \  
--versioning-configuration Status=Enabled \  
--profile acctA
```

4. Em seguida, crie um perfil de serviço do IAM. Esse perfil é especificado na configuração da replicação que posteriormente você adiciona ao bucket *DOC-EXAMPLE-SOURCE-BUCKET*. O Amazon S3 assume essa função para replicar objetos em seu nome. A função do IAM é criada em duas etapas:

- Crie uma função de serviço.
- Anexar uma política de permissões à função.

a. Para criar um perfil de serviço do IAM, faça o seguinte:

- 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 serviço Amazon S3 as principais permissões para assumir o perfil a fim de 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. Use o comando a seguir para criar o perfil.

```
$ aws iam create-role \  

```

```
--role-name replicationRolekmsobj \  
--assume-role-policy-document file:///s3-role-trust-policy-kmsobj.json \  
--profile acctA
```

- b. Em seguida, anexe uma política de permissões ao perfil. 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 um perfil do IAM e, posteriormente, anexe a política a ele.

⚠ Important

Na política de permissões, especifique os IDs de chave do AWS KMS que serão usados para criptografar os buckets *DOC-EXAMPLE-SOURCE-BUCKET* e *DOC-EXAMPLE-DESTINATION-BUCKET*. Você deve criar duas chaves separadas do KMS para os buckets *DOC-EXAMPLE-SOURCE-BUCKET* e *DOC-EXAMPLE-DESTINATION-BUCKET*. As AWS KMS keys não são compartilhadas fora da Região da AWS na qual elas foram criadas.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Action": [  
        "s3:ListBucket",  
        "s3:GetReplicationConfiguration",  
        "s3:GetObjectVersionForReplication",  
        "s3:GetObjectVersionAcl",  
        "s3:GetObjectVersionTagging"  
      ],  
      "Effect": "Allow",  
      "Resource": [  
        "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET",  
        "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET/*"  
      ]  
    },  
    {  
      "Action": [  
        "s3:ReplicateObject",
```

```

        "s3:ReplicateDelete",
        "s3:ReplicateTags"
    ],
    "Effect": "Allow",
    "Condition": {
        "StringLikeIfExists": {
            "s3:x-amz-server-side-encryption": [
                "aws:kms",
                "AES256",
                "aws:kms:dsse"
            ],
            "s3:x-amz-server-side-encryption-aws-kms-key-id": [
                "AWS KMS key IDs(in ARN format) to use for encrypting  
object replicas"
            ]
        }
    },
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET/*"
},
{
    "Action": [
        "kms:Decrypt"
    ],
    "Effect": "Allow",
    "Condition": {
        "StringLike": {
            "kms:ViaService": "s3.us-east-1.amazonaws.com",
            "kms:EncryptionContext:aws:s3:arn": [
                "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET/*"
            ]
        }
    },
    "Resource": [
        "AWS KMS key IDs(in ARN format) used to encrypt source  
objects."
    ]
},
{
    "Action": [
        "kms:Encrypt"
    ],
    "Effect": "Allow",
    "Condition": {
        "StringLike": {

```

```

        "kms:ViaService":"s3.us-west-2.amazonaws.com",
        "kms:EncryptionContext:aws:s3:arn":[
            "arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET/*"
        ]
    },
    "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. Depois, adicione a configuração de replicação a seguir ao bucket *DOC-EXAMPLE-SOURCE-BUCKET*. Isso instrui o Amazon S3 a replicar objetos com o prefixo *Tax/* no bucket *DOC-EXAMPLE-DESTINATION-BUCKET*.

Important

Na configuração da replicação, especifique o perfil 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 essa permissão. Para ter mais informações, consulte [Conceder permissões ao usuário para transmitir um perfil a um AWS service \(Serviço da AWS\)](#) no Guia do usuário do IAM.

```

<ReplicationConfiguration>
  <Role>IAM-Role-ARN</Role>
  <Rule>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
      <Status>Disabled</Status>
    </DeleteMarkerReplication>
  </Rule>
</ReplicationConfiguration>

```



```

</DeleteMarkerReplication>
<Filter>
  <Prefix>Tax</Prefix>
</Filter>
<Status>Enabled</Status>
<SourceSelectionCriteria>
  <SseKmsEncryptedObjects>
    <Status>Enabled</Status>
  </SseKmsEncryptedObjects>
</SourceSelectionCriteria>
<Destination>
  <Bucket>arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET</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 *DOC-EXAMPLE-SOURCE-BUCKET*, faça o seguinte:

- a. 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:::DOC-EXAMPLE-DESTINATION-BUCKET",
        "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 *DOC-EXAMPLE-DESTINATION-BUCKET*, os *AWS KMS key IDs (in ARN format)* e o *IAM-role-ARN*. Salve as alterações.
- c. Use o comando a seguir para adicionar a configuração de replicação ao bucket *DOC-EXAMPLE-SOURCE-BUCKET*. Não deixe fornecer o nome do bucket de *DOC-EXAMPLE-SOURCE-BUCKET*.

```

$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket DOC-EXAMPLE-SOURCE-BUCKET \
--profile acctA

```

6. Teste a configuração para verificar se os objetos criptografados estão replicados. No console do Amazon S3, faça o seguinte:
 - 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 *DOC-EXAMPLE-SOURCE-BUCKET*, crie uma pasta denominada 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 *DOC-EXAMPLE-DESTINATION-BUCKET* contém as réplicas dos objetos e se elas são criptografadas usando a chave do KMS especificada na configuração. Para ter mais informações, consulte [the section called "Obtenção de status da replicação"](#).

Uso da SDKs AWS

Para obter um exemplo de código que mostra como adicionar uma configuração da replicação, consulte [Uso dos AWS SDKs](#). Você precisa modificar a configuração de replicação adequadamente.

Para obter informações conceituais, consulte [Replicar objetos criados com criptografia do lado do servidor \(SSE-C, SSE-S3, SSE-KMS, DSSE-KMS\)](#).

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 S3 no CloudWatch”](#).

Uso do console do S3

Para obter instruções detalhadas, consulte [Configuração da replicação para buckets de origem e de destino pertencentes à mesma conta](#). 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.

Usando 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 \(S3 RTC\)](#).

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](#).

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](#).

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 de 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](#).

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](#).

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](#).

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](#) e [Demonstrações: exemplos para configurar a replicação](#).
- 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 configuração 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](#).
- 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. A exclusão e a recriação do bucket de destino não iniciarão a replicação.

Para obter mais informações sobre cópia em lotes, consulte [Exemplos que usam o Batch Operations para copiar objetos](#).

- Se você estiver usando uma regra de replicação no bucket do S3, [atualize sua configuração de replicação](#), concedendo ao perfil do IAM anexado à regra de replicação as permissões adequadas para replicar objetos. O perfil do IAM deve ter permissões para executar a ação do S3 no bucket de origem e de destino.
- Se você enviar vários trabalhos de replicação em lote para o mesmo bucket em um curto espaço de tempo, o S3 executará esses trabalhos simultaneamente.
- Se você enviar vários trabalhos de replicação em lote para dois buckets diferentes, esteja ciente de que o S3 pode não executar todos os trabalhos simultaneamente. Se você exceder o número de trabalhos de replicação em lote que podem ser executados ao mesmo tempo em sua conta, o S3 pausará os trabalhos de menor prioridade para trabalhar nos de maior prioridade. Depois que os itens de maior prioridade forem concluídos, todos os trabalhos pausados se tornarão ativos novamente.
- A replicação em lote não é compatível para objetos armazenados nas classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive.
- Para replicar em lote objetos do S3 Intelligent-Tiering armazenados no nível de armazenamento Archive Access ou Deep Archive Access, é necessário primeiro iniciar uma solicitação de [restauração](#) e aguardar até que os objetos sejam movidos para o nível Frequent Access.

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](#).

Se você preferir que o Amazon S3 gere um arquivo de manifesto em seu nome, os objetos listados usarão o mesmo bucket de origem, prefixo e tags de todas as configurações do bucket de origem. 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](#).

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".

Relatório de conclusão da replicação em lote

Ao criar um trabalho de replicação em lote, solicite um relatório de conclusão CSV. Este relatório mostra objetos, códigos de sucesso ou falha na replicação, saídas e descrições. Para obter mais informações sobre monitoramento de trabalhos e relatórios de conclusão, consulte [Relatórios de conclusão](#).

Para obter uma lista de códigos e descrições de falhas de replicação, consulte [Motivos de falha da replicação do Amazon S3](#).

Introdução à replicação em lote

Para saber mais sobre como usar a replicação em lote, consulte o [Tutorial: replicar objetos existentes nos buckets do Amazon S3 com o S3 Batch Replication](#).

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 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. Consulte mais informações em [Especificando um manifesto para um trabalho de replicação em lote](#).

Política se usar e armazenar 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 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. Selecione 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. Selecione 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 (Perfis) e selecione o perfil 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](#).

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](#).

- 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. Se você quiser executar esse trabalho agora, escolha Sim, replicar objetos existentes.

Se quiser executar esse trabalho posteriormente, selecione Não replicar 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 [Especificar um manifesto](#).

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](#).

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](#).

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](#).


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](#).

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](#).

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 [Especificar um manifesto](#). 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 [Especificar um manifesto](#).

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](#).
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 nesta conta ou 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 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](#).

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](#).

13. (Opcional) Adicione etiquetas de trabalho à tarefa de replicação em lote.
14. Escolha 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 de Replicação em Lote do S3 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](#).

```
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](#).

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 de versão especificado no manifesto será replicado. Para obter mais informações sobre como criar um manifesto, consulte [Especificar um manifesto](#).

```
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_CSV_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](#).

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](#).

Tópicos

- [Monitorar o progresso com métricas de replicação e Notificações de eventos do S3](#)
- [Atendimento aos requisitos de conformidade usando o Controle do tempo de replicação do S3 \(S3 RTC\)](#)
- [Replicação de marcadores de exclusão entre intervalos](#)
- [Replicação de alterações de metadados com sincronização de modificação de réplica do Amazon S3](#)
- [Alterar o proprietário da réplica](#)
- [Replicar objetos criados com criptografia do lado do servidor \(SSE-C, SSE-S3, SSE-KMS, DSSE-KMS\)](#)

Monitorar o progresso com métricas de replicação e Notificações de eventos do S3

As métricas da funcionalidade Replicação do S3 são métricas detalhadas para as regras de replicação na configuração de replicação. Com as métricas de replicação, você pode monitorar o andamento minuto a minuto rastreando bytes pendentes, operações pendentes, operações com falha na replicação e latência de replicação.

As métricas de replicação do S3 são ativadas automaticamente quando você habilita o Controle de Tempo de Replicação do S3 (S3 RTC). Você também pode ativar as métricas de replicação do S3 independentemente do S3 RTC ao criar ou editar uma regra. O S3 RTC inclui outros recursos, como um Acordo de Serviço (SLA) e notificações para limites perdidos. Para ter mais informações, consulte [Atendimento aos requisitos de conformidade usando o Controle do tempo de replicação do S3 \(S3 RTC\)](#).

As métricas de bytes pendentes, operações pendentes e latência de replicação se aplicam somente a novos objetos que são replicados com Replicação do S3 Entre Regiões Diferentes (S3 CRR) ou Replicação do S3 na Mesma Região (S3 SRR). A métrica de falha na replicação das operações rastreia tanto os novos objetos que são replicados com a S3 CRR ou a S3 SRR quanto os objetos existentes que são replicados com a replicação em lote do S3. Para auxiliar na solução de problemas de configuração, você também pode configurar Notificações de eventos do Amazon S3 para receber eventos de falha de replicação.

Quando habilitadas, 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 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 determinada regra de replicação.
- Operações pendentes de replicação: o número de operações pendentes de replicação para determinada regra de replicação. Essa métrica rastreia operações relacionadas a objetos, marcadores de exclusão, tags, listas de controle de acesso (ACLs) e bloqueio de objetos do S3.
- Operações pendentes de replicação: o número de operações pendentes de replicação para determinada regra de replicação. Essa métrica rastreia operações relacionadas a objetos, marcadores de exclusão, tags, ACLs e bloqueio de objetos. Ao contrário de outras métricas de replicação, essa métrica se aplica a novos objetos que são replicados com a S3 CRR ou a S3 SRR quanto os objetos existentes que são replicados com a replicação em lote do S3.

Note

Operações de replicação com falha rastreia as falhas de replicação do S3 agregadas em um intervalo por minuto. Para identificar os objetos específicos que falharam na replicação e seus motivos, assine o evento `OperationFailedReplication` nas Notificações de eventos do Amazon S3. Para ter mais informações, consulte [Receber eventos de falha de replicação com Notificações de Eventos do Amazon S3](#).

Se um trabalho não for executado, as métricas não serão enviadas ao Amazon CloudWatch. Por exemplo, seu trabalho não será executado se você não tiver as permissões necessárias para

executar um trabalho de replicação em lote do S3 ou se as tags ou o prefixo em sua configuração de replicação não coincidirem.

Tópicos

- [Habilitar métricas de replicação do S3](#)
- [Receber eventos de falha de replicação com Notificações de Eventos do Amazon S3](#)
- [Visualizar métricas de replicação usando o console do Amazon S3](#)
- [Motivos de falha da replicação do Amazon S3](#)

Habilitar 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 ativar as métricas de replicação do S3 em sua configuração da replicação quando os buckets de origem e destino pertencerem às mesmas Contas da AWS ou a outras.

Para ativar métricas de replicação usando a AWS Command Line Interface (AWS CLI), você deve adicionar uma configuração de replicação ao bucket de origem com `Metrics` ativado. Neste exemplo de configuração, os objetos sob o prefixo `Tax` são replicados para o bucket de destino `DOC-EXAMPLE-BUCKET`, e são geradas métricas 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](#).

Para obter mais informações sobre como exibir métricas de replicação no console do S3, consulte [Visualizar métricas de replicação usando o console do Amazon S3](#).

Note

As métricas de replicação de S3 são cobradas usando a mesma taxa das métricas personalizadas do Amazon CloudWatch. Para obter mais informações, consulte [Preço do Amazon CloudWatch](#).

Receber eventos de falha de replicação com Notificações de Eventos do Amazon S3

As Notificações de eventos do S3 podem notificar você em instâncias quando os objetos não são replicados para a Região da AWS 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 ter mais informações, consulte [the section called “Notificações de eventos do Amazon S3”](#).

Para obter uma lista dos códigos de falha capturados pelas Notificações de eventos do S3, consulte [Motivos de falha da replicação do Amazon S3](#).

Visualizar 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 do S3 são ativadas automaticamente quando você habilita a replicação com o Controle de Tempo de Replicação do S3 (S3 RTC) usando o AWS Management Console ou a API do Amazon S3. Você também pode ativar as métricas de replicação do S3 independentemente do S3 RTC ao criar ou editar uma regra.

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 métricas do CloudWatch para o Amazon S3, consulte [Monitoramento de métricas com o Amazon CloudWatch](#).

Pré-requisitos

Ative uma regra de replicação que tenha métricas de replicação do S3.

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. No painel de navegação à esquerda, escolha Buckets. Na lista Buckets, selecione 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 grafos de latência de replicação (em segundos), bytes pendentes de replicação, operações pendentes de replicação e operações de replicação com falha.

Depois, você pode visualizar as métricas de replicação Latência de replicação (em segundos), Operações pendentes de replicação e Replicação de bytes pendentes para as regras selecionadas. Se você estiver usando o controle de tempo de replicação do S3, o Amazon CloudWatch começará a relatar métricas de replicação 15 minutos após você ativar o S3 RTC na respectiva regra de replicação. Você pode visualizar métricas de replicação no console do Amazon S3 ou do CloudWatch. Para ter mais informações, consulte [Métricas de replicação com o S3 RTC](#).

Motivos de falha da replicação do Amazon S3

A tabela a seguir relaciona os motivos de falha da replicação do Amazon S3. Você pode visualizar esses motivos ao receber o evento failureReason com as Notificações de eventos do Amazon S3. Você pode receber Notificações de eventos do S3 por meio do Amazon Simple Queue Service (Amazon SQS), do Amazon Simple Notification Service (Amazon SNS) ou do AWS Lambda. Para ter mais informações, consulte [Notificações de eventos do Amazon S3](#).

Você também pode visualizar esses motivos de falha em um relatório de conclusão da replicação em lote do S3. Para ter mais informações, consulte [Relatório de conclusão da replicação em lote](#).

Motivos de falha da replicação	Descrição
AssumeRoleNotPermitted	O Amazon S3 não pode assumir o perfil do AWS Identity and Access Management (IAM) especificado na configuração de replicação ou no trabalho de operações em lote.
DstBucketInvalidRegion	O bucket de destino não está na mesma Região da AWS especificada pelo trabalho de operações em lote. Esse erro é específico da replicação em lote.
DstBucketNotFound	O Amazon S3 não consegue encontrar o bucket de destino especificado na configuração de replicação.
DstBucketObjectLockConfigMissing	Para replicar objetos de um bucket de origem com o bloqueio de objetos habilitado, o destino também deve ter o bloqueio de objetos habilitado. O erro indica que o bloqueio de objetos pode não estar ativado no bucket de destino. Para ter mais informações, consulte Considerações sobre a funcionalidade Bloqueio de Objetos .
DstBucketUnversioned	O versionamento não está ativado no bucket de destino do S3. Para ativar objetos com replicação do S3, ative o versionamento no bucket de destino.
DstDeleteNotPermitted	O Amazon S3 não consegue replicar marcadores de exclusão no bucket de destino. A permissão <code>s3:ReplicateDelete</code> pode estar faltando para o bucket de destino.
DstKmsKeyInvalidState	A chave do AWS Key Management Service (AWS KMS) para o bucket de destino está

Motivos de falha da replicação	Descrição
	em estado inválido. Revise e ative a chave do AWS KMS necessária. Para ter mais informações sobre o gerenciamento de chaves do AWS KMS, consulte Estados das chaves do AWS KMS no Guia do desenvolvedor do AWS Key Management Service.
DstKmsKeyNotFound	A chave AWS KMS configurada para o bucket de destino na configuração de replicação não existe.
DstMultipartCompleteNotPermitted	O Amazon S3 não consegue concluir o multipart uploads de objetos para o bucket de destino. A permissão <code>s3:ReplicateObject</code> pode estar faltando para o bucket de destino.
DstMultipartInitNotPermitted	O Amazon S3 não consegue iniciar multipart uploads de objetos para o bucket de destino. A permissão <code>s3:ReplicateObject</code> pode estar faltando para o bucket de destino.
DstMultipartPartUploadNotPermitted	O Amazon S3 não consegue fazer upload de objetos fracionados no bucket de destino. A permissão <code>s3:ReplicateObject</code> pode estar faltando para o bucket de destino.
DstObjectHardDeleted	A replicação em lote não é compatível com a nova replicação de objetos excluídos com o ID de versão do objeto do bucket de destino. Esse erro é específico da replicação em lote.

Motivos de falha da replicação	Descrição
<code>DstPutAclNotPermitted</code>	O Amazon S3 não consegue replicar listas de controle de acesso (ACLs) de objetos no bucket de destino. A permissão <code>s3:ReplicateObject</code> pode estar faltando para o bucket de destino.
<code>DstPutLegalHoldNotPermitted</code>	O Amazon S3 não consegue colocar uma retenção legal de bloqueio de objetos nos objetos de destino ao replicar objetos imutáveis. A permissão <code>s3:PutObjectLegalHold</code> pode estar faltando para o bucket de destino. Para ter mais informações, consulte Retenções legais .
<code>DstPutObjectNotPermitted</code>	O Amazon S3 não consegue replicar objetos no bucket de destino. As permissões <code>s3:ReplicateObject</code> ou <code>s3:ObjectOwnerOverrideToBucketOwner</code> podem estar faltando para o bucket de destino.
<code>DstPutTaggingNotPermitted</code>	O Amazon S3 não consegue replicar etiquetas de objetos no bucket de destino. A permissão <code>s3:ReplicateObject</code> pode estar faltando para o bucket de destino.
<code>DstVersionNotFound</code>	O Amazon S3 não consegue encontrar a versão necessária do objeto no bucket de destino para o qual os metadados precisam ser replicados.

Motivos de falha da replicação	Descrição
<code>InitiateReplicationNotPermitted</code>	O Amazon S3 não consegue iniciar a replicação em objetos. A permissão <code>s3:InitiateReplication</code> pode estar ausente para o trabalho de operações em lote. Esse erro é específico da replicação em lote.
<code>SrcBucketInvalidRegion</code>	O bucket de origem não está na mesma Região da AWS especificada pelo trabalho de operações em lote. Esse erro é específico da replicação em lote.
<code>SrcBucketNotFound</code>	O Amazon S3 não consegue encontrar o bucket de origem.
<code>SrcBucketReplicationConfigMissing</code>	O Amazon S3 não conseguiu encontrar uma configuração de replicação para o bucket de origem.

Motivos de falha da replicação	Descrição
<code>SrcGetAclNotPermitted</code>	<p>O Amazon S3 não consegue acessar o objeto no bucket de origem para replicação. A permissão <code>s3:GetObjectVersionAcl</code> pode estar faltando para o objeto do bucket de origem.</p> <p>Os objetos no bucket de origem devem ser de do proprietário do bucket. Se as ACLs estiverem habilitadas, verifique se a “Propriedade do objeto” está definida como “Proprietário do bucket preferido” ou “Autor do objeto”. Se a “Propriedade do objeto” estiver definida como “Proprietário do bucket preferido”, os objetos do bucket de origem deverão ter a ACL <code>bucket-owner-full-control</code> para que o proprietário do bucket se torne o proprietário do objeto. A conta de origem pode se tornar proprietária de todos os objetos no bucket definindo a “Propriedade do objeto” como “Imposto pelo proprietário do bucket” e desabilitando as ACLs.</p>
<code>SrcGetLegalHoldNotPermitted</code>	<p>O Amazon S3 não consegue acessar as informações de retenção legal do Bloqueio de objetos do S3.</p>
<code>SrcGetObjectNotPermitted</code>	<p>O Amazon S3 não consegue acessar o objeto no bucket de origem para replicação. A permissão <code>s3:GetObjectVersionForReplication</code> pode estar faltando para o bucket de origem.</p>
<code>SrcGetRetentionNotPermitted</code>	<p>O Amazon S3 não consegue acessar as informações do período de retenção do Bloqueio de objetos do S3.</p>

Motivos de falha da replicação	Descrição
<code>SrcGetTaggingNotPermitted</code>	O Amazon S3 não consegue acessar as informações da tag do objeto pelo bucket de origem. A permissão <code>s3:GetObjectVersionTagging</code> pode estar faltando para o bucket de origem.
<code>SrcHeadObjectNotPermitted</code>	O Amazon S3 não consegue recuperar os metadados do objeto pelo bucket de origem. A permissão <code>s3:GetObjectVersionForReplication</code> pode estar faltando para o bucket de origem.
<code>SrcKeyNotFound</code>	O Amazon S3 não consegue encontrar a chave do objeto de origem para replicar. O objeto de origem pode ter sido excluído antes da conclusão da replicação.
<code>SrcKmsKeyInvalidState</code>	A chave do AWS KMS para o bucket de origem não está em estado válido. Revise e ative a chave do AWS KMS necessária. Para ter mais informações sobre o gerenciamento de chaves do AWS KMS, consulte Estados das chaves do AWS KMS no Guia do desenvolvedor do AWS Key Management Service.
<code>SrcObjectNotEligible</code>	Alguns objetos não são elegíveis para replicação. O motivo disso pode ser a falta de correspondência entre a classe de armazenamento do objeto ou as tags do objeto com a configuração da replicação.
<code>SrcObjectNotFound</code>	O objeto de origem não existe.
<code>SrcReplicationNotPending</code>	O Amazon S3 já replicou esse objeto. Esse objeto não está mais pendente de replicação.

Motivos de falha da replicação	Descrição
SrcVersionNotFound	O Amazon S3 não consegue encontrar a versão do objeto de origem para replicar. A versão do objeto de origem pode ter sido excluída antes da conclusão da replicação.

Tópicos relacionados da

[Configuração de permissões](#)

[Solução de problemas de replicação](#)

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 são replicados em até 15 minutos e quando esses objetos são replicados depois do limite de 15 minutos. Os eventos do Amazon S3 estão disponíveis por meio do Amazon SQS, Amazon SNS ou AWS Lambda. Para ter mais informações, consulte [the section called “Notificações de eventos do Amazon S3”](#).

Tópicos

- [Habilitar o Controle do tempo de replicação do S3](#)
- [Métricas de replicação com o S3 RTC](#)

- [Usar notificações de eventos do Amazon S3 para rastrear objetos de replicação](#)
- [Melhores práticas e diretrizes do S3 RTC](#)

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 \(S3 RTC\)](#).

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 as métricas de replicação, é possível monitorar o número total de operações de API do S3 com replicação pendente, o tamanho total dos objetos com replicação pendente, o tempo máximo de replicação para a região de destino e o número total de operações cuja replicação apresentou falha. 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](#).

Para obter mais informações sobre como encontrar métricas de replicação por meio do console do Amazon S3, consulte [Visualizar métricas de replicação usando o console do Amazon S3](#).

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 elegível para replicação usando o S3 RTC não foi replicado em 15 minutos e quando esse objeto é replicado depois do limite de 15 minutos.

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 ter mais informações, consulte [Notificações de eventos do Amazon S3](#).

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](#)
- [Estimar as taxas de solicitação de replicação](#)
- [Exceder os limites da taxa de transferência de dados do S3 RTC](#)
- [Taxas de solicitação de replicação de objetos criptografados do AWS KMS](#)

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 prefixos em um bucket. Você pode aumentar sua performance de leitura ou gravação paralelizando as leituras. Por exemplo, se você criar 10 prefixos 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 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](#).

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](#).

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](#). 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](#) 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](#). 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](#).

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](#). Para obter mais informações sobre como definir a configuração de replicação, consulte [Replicação de objetos](#).

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](#). 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](#).

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](#).

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>
```

```

    </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.

Replicar objetos criados com criptografia do lado do servidor (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS)

Important

O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).


Há algumas considerações especiais quando você está replicando objetos que foram criptografados usando criptografia do lado do servidor. O Amazon S3 agora é compatível com seguintes tipos de criptografia do lado do servidor:

- Criptografia do 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 de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS)
- Criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C)

Para obter mais informações sobre criptografia no lado do servidor, consulte [the section called “Criptografia do lado do servidor”](#).

Este tópico explica as permissões necessárias para direcionar o Amazon S3 a replicar objetos que foram criptografados usando a criptografia do lado do servidor. Este tópico também fornece elementos de configuração adicionais aos quais você pode adicionar e dar exemplos de políticas do AWS Identity and Access Management (IAM) que concedem as permissões necessárias para replicar objetos criptografados.

Para obter um exemplo com instruções passo a passo, consulte [Replicar objetos criptografados](#). Para obter informações sobre como criar uma configuração da replicação, consulte [Replicação de 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.

Tópicos

- [Como a criptografia de bucket padrão afeta a replicação](#)
- [Replicar objetos criptografados com SSE-C](#)
- [Replicar objetos criptografados com SSE-S3, SSE-KMS ou DSSE-KMS](#)

Como a criptografia de bucket padrão afeta 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. Como resultado, as tags de entidade (ETags) dos objetos de origem diferem dos ETags dos objetos de réplica. Se você tiver aplicações que usam ETags, será necessário atualizar essas aplicações para considerar essa diferença.
- Se os objetos no bucket de origem forem criptografados usando a criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3), a criptografia do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS) ou a criptografia de camada dupla do lado do servidor com chaves do AWS KMS (DSSE-KMS), os objetos de réplica no bucket de destino usarão o mesmo tipo de criptografia que os objetos de origem. As configurações de criptografia padrão do bucket de destino não são usadas.

Replicar objetos criptografados com SSE-C

Ao usar a criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C), é possível gerenciar chaves de criptografia proprietárias. Com o SSE-C, você gerencia as chaves enquanto o Amazon S3 gerencia o processo de criptografia e descriptografia. Você deve fornecer uma chave de criptografia como parte de sua solicitação, mas não precisa escrever nenhum código para realizar a criptografia ou descriptografia de objetos. Quando você faz upload de um objeto, o Amazon S3 criptografa o objeto usando a chave que você forneceu. Depois, o Amazon S3 remove essa chave da memória. Quando você recupera um objeto, deve fornecer a mesma chave de criptografia como parte de sua solicitação. Para ter mais informações, consulte [the section called “Chaves de criptografia fornecidas pelo cliente \(SSE-C\)”](#).

A replicação do S3 oferece suporte a objetos criptografados com SSE-C. Você pode configurar a replicação de objetos com SSE-C no console do Amazon S3 ou com os AWS SDKs, da mesma forma que configura a replicação para objetos não criptografados. Não há permissões de SSE-C adicionais além das atualmente necessárias para replicação.

A Replicação do S3 replica automaticamente objetos criptografados com SSE-C recém-carregados, se eles forem elegíveis, de acordo com sua configuração de Replicação do S3. Para replicar objetos existentes nos buckets, use a replicação em lote do S3. Para obter mais informações sobre replicação de objetos, consulte [the section called “Configuração da replicação”](#) e [the section called “Replicar objetos existentes”](#).

Não há cobranças adicionais pela replicação de objetos SSE-C. Para obter detalhes sobre os preços de replicação, consulte a [página de preços do Amazon S3](#).

Replicar objetos criptografados com SSE-S3, SSE-KMS ou DSSE-KMS

Por padrão, o Amazon S3 não replica objetos criptografados com SSE-KMS ou DSSE-KMS. Esta seção explica outros elementos de configuração que você pode adicionar para orientar o Amazon S3 a replicar esses objetos.

Para obter um exemplo com instruções passo a passo, consulte [Replicar objetos criptografados](#). Para obter informações sobre como criar uma configuração da replicação, consulte [Replicação de objetos](#).

Especificar informações adicionais na configuração de replicação

Na configuração de replicação, você faz o seguinte:

- No elemento `Destination` na configuração de replicação, adicione o ID da chave do AWS KMS simétrica gerenciada pelo cliente que você deseja que o Amazon S3 use para criptografar réplicas de objetos, conforme mostrado no exemplo de configuração de replicação a seguir.
- Aceite explicitamente ao habilitar a replicação de objetos criptografados por meio das chaves do KMS (SSE-KMS ou DSSE-KMS). Para ativar, adicione o elemento `SourceSelectionCriteria`, conforme mostrado no exemplo de configuração de replicação a seguir.

```
<ReplicationConfiguration>
  <Rule>
    ...
    <SourceSelectionCriteria>
      <SseKmsEncryptedObjects>
        <Status>Enabled</Status>
      </SseKmsEncryptedObjects>
    </SourceSelectionCriteria>

    <Destination>
      ...
      <EncryptionConfiguration>
        <ReplicaKmsKeyID>AWS KMS key ARN or Key Alias ARN that's in the same
        Região da AWS as 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 operação de API `PutBucketReplication` não verifica a validade de chaves do KMS. Se você usar uma chave do KMS inválida, receberá o código de status HTTP 200 OK como resposta, mas a replicação falhará.

O exemplo a seguir mostra uma configuração de replicação, que inclui elementos de configuração opcionais. 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 AWS KMS key especificado para criptografar essas réplicas de objeto.

```
<?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::DOC-EXAMPLE-DESTINATION-BUCKET</Bucket>
      <EncryptionConfiguration>
        <ReplicaKmsKeyID>AWS KMS key ARN or Key Alias ARN that's in the same Região da AWS as the destination bucket. (S3 uses this key to encrypt object replicas.)</
ReplicaKmsKeyID>
      </EncryptionConfiguration>
    </Destination>
    <SourceSelectionCriteria>
      <SseKmsEncryptedObjects>
        <Status>Enabled</Status>
      </SseKmsEncryptedObjects>
    </SourceSelectionCriteria>
  </Rule>
</ReplicationConfiguration>
```

Conceder permissões adicionais para a função do IAM

Para replicar objetos criptografados em repouso usando SSE-S3, SSE-KMS ou DSSE-KMS, conceda as permissões adicionais a seguir ao perfil do AWS Identity and Access Management (IAM) especificado na configuração da replicação. Você concede essas permissões ao atualizar a política de permissões associada ao perfil do IAM.

- Ação **s3:GetObjectVersionForReplication** para objetos de origem: essa ação permite que o Amazon S3 replique objetos não criptografados e objetos criados com criptografia do lado do servidor usando SSE-S3, SSE-KMS ou DSSE-KMS.

 Note

Recomendamos que você use a ação `s3:GetObjectVersionForReplication` em vez da ação `s3:GetObjectVersion`, porque `s3:GetObjectVersionForReplication` fornece ao Amazon S3 somente as permissões mínimas necessárias para a replicação. Além disso, a ação `s3:GetObjectVersion` permite a replicação de objetos não criptografados e objetos criptografados com SSE-S3, mas não objetos criptografados usando chaves do KMS (SSE-KMS ou DSSE-KMS).

- Ações **kms:Decrypt** e **kms:Encrypt** AWS KMS para as chaves do KMS
 - Você deve conceder permissões `kms:Decrypt` para a AWS KMS key que é usada para descriptografar o objeto de origem.
 - Você deve conceder permissões `kms:Encrypt` para a AWS KMS key que é usada para criptografar a réplica do objeto.
- A ação **kms:GenerateDataKey** para replicar objetos de texto sem formatação: se você estiver replicando objetos de texto sem formatação em um bucket com a criptografia SSE-KMS ou DSSE-KMS habilitada por padrão, é necessário incluir a permissão `kms:GenerateDataKey` para o contexto de criptografia de destino e a chave do KMS na política do IAM.

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 do perfil do IAM precisa ter permissões para as ações `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 proprietária do perfil 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 S3 e replicação

Para usar a replicação com uma chave de bucket do S3, a política AWS KMS key para a chave do KMS usada para criptografar a réplica do objeto deve incluir a permissão `kms:Decrypt` para a entidade principal da chamada. A chamada para `kms:Decrypt` verifica a integridade da chave do

bucket do S3 antes de usá-la. Para ter mais informações, consulte [Uso de uma chave de bucket do S3 com replicação](#).

Quando uma chave de bucket do S3 estiver habilitada para o bucket de origem ou de destino, o contexto de criptografia será o nome do recurso da Amazon (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 do bucket para o contexto de criptografia:

```
"kms:EncryptionContext:aws:s3:arn": [
  "arn:aws:s3:::bucket_ARN"
]
```

Para obter mais informações, consulte [Contexto de criptografia \(x-amz-server-side-encryption-context\)](#) (na seção “Usar a API REST”) e [Alterações na observação antes de habilitar uma chave de bucket do S3](#).

Exemplo de políticas: Usar SSE-3 e SSE-KMS com replicação

Os exemplos de políticas do IAM a seguir mostram instruções para o uso de SSE-S3 e SSE-KMS com replicação.

Example – Usar SSE-KMS com buckets de destino separados

A política de exemplo a seguir mostra instruções para usar SSE-KMS com buckets de destino separados.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": ["kms:Decrypt"],
      "Effect": "Allow",
      "Condition": {
        "StringLike": {
          "kms:ViaService": "s3.source-bucket-region.amazonaws.com",
          "kms:EncryptionContext:aws:s3:arn": [
            "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET/key-prefix1*"
          ]
        }
      }
    },
    {
      "Resource": [
        "List of AWS KMS key ARNs that are used to encrypt source objects."
      ]
    }
  ]
}
```

```

    ]
  },
  {
    "Action": ["kms:Encrypt"],
    "Effect": "Allow",
    "Condition": {
      "StringLike": {
        "kms:ViaService": "s3.destination-bucket-1-region.amazonaws.com",
        "kms:EncryptionContext:aws:s3:arn": [
          "arn:aws:s3::DOC-EXAMPLE-DESTINATION-BUCKET1/key-prefix1*"
        ]
      }
    },
    "Resource": [
      "AWS KMS key ARNs (in the same Região da AWS as destination bucket 1). Used to encrypt object replicas created in destination bucket 1."
    ]
  },
  {
    "Action": ["kms:Encrypt"],
    "Effect": "Allow",
    "Condition": {
      "StringLike": {
        "kms:ViaService": "s3.destination-bucket-2-region.amazonaws.com",
        "kms:EncryptionContext:aws:s3:arn": [
          "arn:aws:s3::DOC-EXAMPLE-DESTINATION-BUCKET2/key-prefix1*"
        ]
      }
    },
    "Resource": [
      "AWS KMS key ARNs (in the same Região da AWS as destination bucket 2). Used to encrypt object replicas created in destination bucket 2."
    ]
  }
]
}

```

Example – Replicar objetos criados com SSE-S3 e SSE-KMS

Veja a seguir uma política completa do IAM que concede as permissões necessárias para replicar objetos não criptografados, objetos criados com SSE-3 e objetos criados com SSE-KMS.

```
{
```

```
"Version":"2012-10-17",
"Statement":[
  {
    "Effect":"Allow",
    "Action":[
      "s3:GetReplicationConfiguration",
      "s3:ListBucket"
    ],
    "Resource":[
      "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET"
    ]
  },
  {
    "Effect":"Allow",
    "Action":[
      "s3:GetObjectVersionForReplication",
      "s3:GetObjectVersionAcl"
    ],
    "Resource":[
      "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET/key-prefix1*"
    ]
  },
  {
    "Effect":"Allow",
    "Action":[
      "s3:ReplicateObject",
      "s3:ReplicateDelete"
    ],
    "Resource":"arn:aws:s3:::DOC-EXAMPLE-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:::DOC-EXAMPLE-SOURCE-BUCKET/key-prefix1*"
        ]
      }
    }
  },
  "Resource":[
```

```

    "List of the AWS KMS key ARNs that are used to encrypt source objects."
  ],
},
{
  "Action":[
    "kms:Encrypt"
  ],
  "Effect":"Allow",
  "Condition":{"
    "StringLike":{"
      "kms:ViaService":"s3.destination-bucket-region.amazonaws.com",
      "kms:EncryptionContext:aws:s3:arn":["
        "arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET/prefix1*"
      ]
    }
  },
  "Resource":["
    "AWS KMS key ARNs (in the same Região da AWS as the destination bucket) to use for encrypting object replicas"
  ]
}
]
}

```

Example – Replicar objetos com chaves de bucket do S3

Veja a seguir uma política completa do IAM que concede as permissões necessárias para replicar objetos com chaves de bucket do S3.

```

{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:GetReplicationConfiguration",
        "s3:ListBucket"
      ],
      "Resource":["
        "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET"
      ]
    },
    {

```

```

    "Effect": "Allow",
    "Action": [
      "s3:GetObjectVersionForReplication",
      "s3:GetObjectVersionAcl"
    ],
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET/key-prefix1*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:ReplicateObject",
      "s3:ReplicateDelete"
    ],
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-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:::DOC-EXAMPLE-SOURCE-BUCKET"
        ]
      }
    },
    "Resource": [
      "List of the AWS KMS key ARNs that are used to encrypt source objects."
    ]
  },
  {
    "Action": [
      "kms:Encrypt"
    ],
    "Effect": "Allow",
    "Condition": {
      "StringLike": {
        "kms:ViaService": "s3.destination-bucket-region.amazonaws.com",
        "kms:EncryptionContext:aws:s3:arn": [
          "arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET"
        ]
      }
    }
  }
}

```

```
    ]
  },
  "Resource": [
    "AWS KMS key ARNs (in the same Região da AWS as the destination bucket) 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 do KMS 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.

Note

Objetos criptografados por meio de [Chaves gerenciadas pela AWS](#) não podem ser compartilhados entre contas porque você não pode modificar as políticas de chaves. Se você precisar replicar dados de SSE-KMS entre contas, é necessário use uma [chave gerenciada pelo cliente](#) do AWS KMS.

Como conceder ao proprietário do bucket de origem permissão para usar a chave do KMS (console do AWS KMS)

1. Faça login no AWS Management Console e abra o console do 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. Na seção Configuração geral, selecione a guia Política de chaves.
6. Role para baixo até Outras Contas da AWS.

7. Selecione Adicionar outras Contas da AWS.

A caixa de diálogo Outras Contas da AWS é exibida.

8. Na caixa de diálogo, escolha Adicionar outra Conta da AWS. Para `arn:aws:iam::`, insira o ID da conta do bucket de origem.
9. Escolha Salvar alterações.

Para conceder ao proprietário do bucket de origem permissão para usar a chave do KMS (AWS CLI).

- Para obter informações sobre o comando `put-key-policy` da AWS Command Line Interface (AWS CLI), consulte [put-key-policy](#) na Referência de comandos da AWS CLI. Para obter informações sobre a operação de API subjacente `PutKeyPolicy`, consulte [PutKeyPolicy](#) na [Referência de API do AWS Key Management Service](#).

Considerações sobre cotas de transação do AWS KMS

Ao adicionar muitos novos objetos com a criptografia do AWS KMS depois de habilitar replicação entre regiões (CRR), pode ocorrer controle de utilização (erros de respostas HTTP 503 `Service Unavailable`). O controle de utilização acontece quando o número de transações do AWS KMS por segundo excede a cota atual. Para obter mais informações, consulte [Cotas do](#) no Guia do desenvolvedor do AWS Key Management Service.

Para solicitar um aumento de cota, use Service Quotas. Consulte [Requesting a quota increase](#) (Como solicitar um aumento de cota) para obter mais informações. 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 `REPLICA`.

Tópicos

- [Visão geral do status da replicação](#)
- [Status da replicação, se replicar para vários intervalos de destino](#)
- [Status da replicação se a sincronização de modificação de réplica do Amazon S3 estiver ativada](#)
- [Localização do status de replicação](#)

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 `REPLICA`.

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](#).

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.
4. Na guia Properties (Propriedades), encontre Object management overview (Visão geral do gerenciamento de objetos) para conferir o Replication status (Status de replicação).

Usando 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.HIM0fYiD_9rGbSkmroXsFj3fqZ.",
  "ETag": "\"6805f2cfc46c0f04559748bb039d69ae\"",
  "Metadata": {}
}
```

Uso dos AWS SDKs

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);
```

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](#).
- Hospedagem de sites: para obter mais informações, consulte [Hospedagem de um site estático usando o Amazon S3](#).
- Acesso ao bucket por meio de uma política ou lista de controle de acesso (ACL) — Para obter mais informações, consulte [Políticas de bucket para o Amazon S3](#) e [Visão geral da lista de controle de acesso \(ACL\)](#).
- Armazenamento de logs: para obter mais informações, consulte [Registrar em log as solicitações com registro em log de acesso ao servidor](#).
- Gerenciamento do ciclo de vida dos objetos de um bucket: para obter mais informações, consulte [Gerenciando seu ciclo de vida de armazenamento](#).

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](#)
- [Configuração do versionamento e configuração de replicação](#)
- [Usar a replicação do S3 com o S3 Intelligent-Tiering](#)
- [Configuração de log e de replicação](#)
- [CRR e a região de destino](#)
- [Pausar a replicação](#)

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 configuração de ciclo de vida estiver ativada em um bucket de destino, as regras de ciclo de vida honrarão 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 configuração de ciclo de vida de expiração de um objeto, depois de ativar o versionamento, adicione uma política `NonCurrentVersionExpiration` para manter o mesmo comportamento de exclusão permanente de antes de ativar o versionamento.
- Se você tiver uma configuração de ciclo de vida de transição, depois de ativar o versionamento, considere adicionar uma 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`.

Usar a replicação do S3 com o S3 Intelligent-Tiering

A classe de armazenamento do S3 Intelligent-Tiering foi projetada para otimizar os custos de armazenamento, movendo automaticamente os dados para o nível de acesso mais econômico. 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.

Replicação de objetos armazenados no S3 Intelligent-Tiering com replicação em lote do S3 ou invocação de [CopyObject](#) ou [UploadPartCopy](#) constitui acesso. Nesses casos, os objetos de origem das operações de cópia ou replicação são hierarquizados.

Para obter mais informações sobre o S3 Intelligent-Tiering, consulte [Amazon S3 Intelligent-Tiering](#).

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](#)) ou os logs do AWS CloudTrail ([Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail](#)) estiverem habilitados no bucket de origem ou de destino, o 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 tags de objeto do Amazon S3 são representadas internamente em UTF-16. Observe que, em UTF-16, os caracteres utilizam posições de um ou de dois caracteres.
- 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](#).

Exemplos

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 permissões a objetos somente leitura com etiquetas 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 operação de 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. É possível usar as operações em lote do S3 por meio do console do Amazon S3, da AWS CLI, de AWS SDKs ou da API REST. Para ter mais informações, consulte [the section called “Conceitos básicos do Batch Operations”](#).


Para obter mais informações sobre tags de objeto, consulte [Gerenciar tags de objeto](#).

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 de solicitação. Há dois cenários distintos de gerenciamento de tags de objeto usando essa API.
 - O objeto não tem tags – Usando essa API, você pode adicionar um conjunto de tags a um objeto (o objeto não tem nenhuma tag anterior).
 - O objeto tem um conjunto de tags existentes – Para modificar o conjunto de tags existente, você deve primeiro recuperar o conjunto de tags existente, modificá-lo no lado do cliente e usar essa API para substituir o conjunto de tags.

 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 ultrapasarem 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](#).

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](#).

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 ter mais informações, consulte [Configuração da replicação](#).

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](#)
- [Gerenciar tags de objeto](#)

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:

- [Operações com objetos](#)
- [Operações de buckets](#)

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 políticas de bucket usando chaves de condição](#). 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 a chave e o valor de uma etiqueta específica

A política de permissões a seguir limita a capacidade de leitura dos usuários para que leiam somente objetos que tenham a chave e o valor da tag `environment: production`. Essa política usa a chave de condição `s3:ExistingObjectTag` para especificar a chave e o valor da etiqueta.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:role/JohnDoe"
        ]
      },
      "Effect": "Allow",
      "Action": ["s3:GetObject", "s3:GetObjectVersion"],
      "Resource": "arn:aws:s3::DOC-EXAMPLE-BUCKET/*",
      "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/environment": "production"
        }
      }
    }
  ]
}
```

Example 2: Restringir quais chaves de etiqueta de objeto os usuários podem adicionar

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 usa a chave de condição `s3:RequestObjectTagKeys` para especificar as chaves de etiqueta permitidas, como `Owner` ou `CreationDate`. Para obter mais informações, consulte [Criar uma condição que testa vários valores de chave](#) no Guia do usuário do IAM.

A política garante que cada chave de etiqueta especificada na solicitação seja uma chave de etiqueta autorizada. O qualificador `ForAnyValue` na condição garante que pelo menos uma das chaves especificadas esteja presente na solicitação.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:role/JohnDoe"
        ]
      },
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ],
      "Condition": {
        "ForAnyValue:StringEquals": {
          "s3:RequestObjectTagKeys": [
            "Owner",
            "CreationDate"
          ]
        }
      }
    }
  ]
}
```

Example 3: Exigir a chave e o valor de uma etiqueta específica ao permitir que os usuários adicionem etiquetas de objeto

O exemplo de política a seguir concede ao usuário permissão para executar a ação `s3:PutObjectTagging`, que permite que o usuário adicione etiquetas a um objeto existente. A

condição requer que o usuário inclua uma chave de etiqueta específica (como *Project*) com o valor definido como *X*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:user/JohnDoe"
        ]
      },
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ],
      "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. As tags de objeto do Amazon S3 são representadas internamente em UTF-16. Observe que, em UTF-16, os caracteres utilizam posições de um ou de dois caracteres.
- As chaves e os valores diferenciam letras maiúsculas de minúsculas.

Para obter mais informações sobre tags de objeto, consulte [Categorizando seu armazenamento usando tags](#). 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](#) e [Fazer upload de objetos](#) 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](#). Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3](#).


```
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](#).

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

```
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](#).

O relatório de alocação de custos indica o uso da sua conta na AWS por categoria de produto e usuário de conta vinculado. 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](#)) 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](#).
- 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](#)
- [Relatórios do AWS Billing para o Amazon S3](#)

Relatórios de uso e faturamento do Amazon S3

Important

Em 13 de maio de 2024, começamos a implantar uma alteração para eliminar cobranças por solicitações não autorizadas que não foram iniciadas pelo proprietário do bucket. Depois que a implantação dessa alteração for concluída, os proprietários do bucket nunca incorrerão em cobranças de solicitação ou largura de banda para solicitações que retornem erros `AccessDenied` (HTTP 403 `Forbidden`) quando forem iniciadas de fora de sua conta da AWS individual ou de organização da AWS. Consulte uma lista completa de códigos de status HTTP 3XX e 4XX que não serão cobrados em [Cobrança pelas respostas de erro do Amazon S3](#). Essa alteração de cobrança não exige atualizações nas aplicações e é destinada a todos os buckets do S3. Quando a implantação dessa alteração for concluída em todas as Regiões da AWS, atualizaremos nossa documentação.

Ao usar o Amazon S3, você não tem que pagar nenhuma taxa inicial nem assumir nenhum compromisso em relação à quantidade de conteúdo que armazenará. Assim como outros Serviços da AWS, você paga somente pelo que usar.

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](#). Para obter mais informações sobre relatórios de faturamento, consulte [Relatórios do AWS Billing para o Amazon S3](#).
- 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](#).

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

Tópicos

- [Relatórios do AWS Billing para o Amazon S3](#)
- [AWSRelatório de uso da para o Amazon S3](#)
- [Como entender os relatórios de uso e faturamento da AWS para o Amazon S3](#)
- [Cobrança pelas respostas de erro do Amazon S3](#)

Relatórios do AWS Billing para o Amazon S3

Sua fatura mensal da AWS separa suas informações de uso e o custo por AWS service (Serviço da AWS) e por função. Há vários relatórios do AWS Billing 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.

Para acompanhar o seu uso da AWS e fornecer as cobranças estimadas associadas à sua conta, você pode configurar Cost and Usage Reports da AWS. Para obter mais informações, consulte [O que é são os Cost and Usage Reports da AWS?](#) no Guia de exportações de dados 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](#).

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 cobrada depende do tamanho dos seus objetos, do tempo que você os armazenou durante o mês e das classes de armazenamento. O Amazon S3 oferece as seguintes classes de armazenamento: S3 Standard, S3 Express One Zone, S3 Intelligent-Tiering, S3 Standard-IA (IA para acesso infrequente), S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive ou Reduced Redundancy Storage (RRS). Para obter mais

Cobrança	Comentários
	<p>informações sobre classes de armazenamento, consulte Uso de classes de armazenamento do Amazon S3.</p> <p>Esteja ciente de que, se a funcionalidade Versionamento do S3 estiver habilitada, você será cobrado por cada versão de um objeto retido. Para obter mais informações sobre versionamento, consulte Como funciona o Versionamento do S3.</p>
Monitoramento e automação	<p>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.</p>
Solicitações	<p>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.</p>
Recuperações	<p>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.</p>

Cobrança	Comentários
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 (Inventário Amazon S3, 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 a funcionalidade Aceleração de Transferências do Amazon S3.</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](#).

Mais informações

- [AWSRelatório de uso da para o Amazon S3](#)
- [Usar tags de alocação de custos para buckets do S3](#)
- [Gerenciamento de Faturamento e Custos da AWS Billing](#)
- [Preços do Amazon S3](#)

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 agregados. Para obter mais informações sobre os relatórios de uso, consulte [AWS Usage Report](#) no Guia do usuário de exportações de dados da AWS.

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](#).
- 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](#).

- 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](#).

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](#).

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, selecione seu nome de usuário ou ID da conta, depois escolha Gerenciamento de faturamento e custos.
3. No painel de navegação, escolha Relatórios de custos e uso.
4. Em Relatório de uso da AWS, selecione Criar um relatório de uso.
5. Na página Baixar relatório de uso, escolha as seguintes configurações:
 - Serviços: escolha Amazon Simple Storage Service.
 - 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](#).
 - Operation (Operação): para obter uma explicação detalhada sobre as operações do Amazon S3, consulte [Controle de operações em seus relatórios de uso](#).
 - 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.
6. Escolha Fazer download, selecione o formato de download (Relatório XML ou Relatório CSV) 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](#)
- [Relatórios do AWS Billing para o Amazon S3](#)

Como entender os relatórios de uso e faturamento da AWS para o Amazon S3

Important

Em 13 de maio de 2024, começamos a implantar uma alteração para eliminar cobranças por solicitações não autorizadas que não foram iniciadas pelo proprietário do bucket. Depois que a implantação dessa alteração for concluída, os proprietários do bucket nunca incorrerão em cobranças de solicitação ou largura de banda para solicitações que retornem erros AccessDenied (HTTP 403 Forbidden) quando forem iniciadas de fora de sua conta da AWS individual ou de organização da AWS. Consulte uma lista completa de códigos de status HTTP 3XX e 4XX que não serão cobrados em [Cobrança pelas respostas de erro do Amazon S3](#). Essa alteração de cobrança não exige atualizações nas aplicações e é destinada a todos os buckets do S3. Quando a implantação dessa alteração for concluída em todas as Regiões da AWS, atualizaremos nossa documentação.

Os relatórios de uso e faturamento do Amazon S3 usam códigos e abreviações. Para os tipos de uso na tabela a seguir, substitua *region*, *region1* e *region2* pelas abreviações 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)
- APS5: Ásia-Pacífico (Hyderabad)
- APS6: Ásia-Pacífico (Melbourne)
- CAN1: Canadá (Central)
- CNN1: China (Pequim)
- CNW1: China (Ningxia)
- AFS1: África (Cidade do Cabo)

- EUC2: Europa (Zurique)
- EUN1: UE (Estocolmo)
- EUS2: Europa (Espanha)
- EUC1: UE (Frankfurt)
- EU: UE (Irlanda)
- EUS1: Europa (Milão)
- EUW2: UE (Londres)
- EUW3: UE (Paris)
- ILC1: Israel (Tel Aviv)
- MEC1: Oriente Médio (EAU)
- 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 tipos de uso de pontos de acesso multirregionais do S3 na tabela a seguir, substitua *regiongroup1* e *regiongroup2* por abreviações desta lista:

- AP: Ásia-Pacífico
- AU: Austrália
- EU: Europa
- IN: Índia
- NA: América do Norte
- SA: América do Sul

Os grupos de regiões são agrupamentos geográficos de várias Regiões da AWS. Para ter mais informações, consulte [Regiões e zonas de disponibilidade](#). 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-region2</i> -AWS-In-A Bytes	GB	Por hora	A quantidade de dados acelerados transferidos de <i>region2</i> para <i>region1</i> .
<i>region1-region2</i> -AWS-In-A Bytes-T1	GB	Por hora	A quantidade de dados acelerados T1 transferidos de <i>region2</i> para <i>region1</i> , em que T1 refere-se a solicitações do CloudFront para pontos de presença (POPs) nos Estados Unidos, na Europa e no Japão.
<i>region1-region2</i> -AWS-In-A Bytes-T2	GB	Por hora	A quantidade de dados acelerados T2 transferidos de <i>region2</i> para <i>region1</i> , em que T2 refere-se a solicitações do CloudFront para POPs em todos os outros locais da borda da AWS.
<i>region1-region2</i> -AWS-In-Bytes	GB	Por hora	A quantidade de dados transferidos de <i>region2</i> para <i>region1</i> .
<i>region1-region2</i> -AWS-Out-A Bytes	GB	Por hora	A quantidade de dados acelerados transferidos

Tipo de uso	Unidades	Granularity	Descrição
			dos de <i>region1</i> para <i>region2</i> .
<i>region1-region2</i> -AWS-Out-Bytes-T1	GB	Por hora	A quantidade de dados acelerados T1 transferidos de <i>region1</i> para <i>region2</i> , em que T1 refere-se a solicitações do CloudFront para POPs nos Estados Unidos, na Europa e no Japão.
<i>region1-region2</i> -AWS-Out-Bytes-T2	GB	Por hora	A quantidade de dados acelerados T2 transferidos de <i>region1</i> para <i>region2</i> , em que T2 refere-se a solicitações do CloudFront para POPs em todos os outros locais da borda da AWS.
<i>region1-region2</i> -AWS-Out-Bytes	GB	Por hora	A quantidade de dados transferidos de <i>region1</i> para <i>region2</i> .
<i>region</i> -BatchOperations-Jobs	Contagem	Por hora	O número de trabalhos de Operações em lote do S3 executados
<i>region</i> -BatchOperations-Objects	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 da classe de armazenamento S3 Glacier Deep Archive.
<i>region</i> -BytesDeleted-GIR	GB	Mensal	A quantidade de dados excluídos por uma operação DeleteObject da classe de armazenamento S3 Glacier Instant Retrieval.
<i>region</i> -BytesDeleted-GLACIER	GB	Mensal	A quantidade de dados excluídos por uma operação DeleteObject da classe de armazenamento S3 Glacier Flexible Retrieval.
<i>region</i> -BytesDeleted-INT	GB	Mensal	A quantidade de dados excluídos por uma operação DeleteObject da classe de armazenamento S3 Intelligent-Tiering.

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -BytesDeleted-RRS	GB	Mensal	A quantidade de dados excluídos por uma operação DeleteObject da classe de armazenamento Reduced Redundancy Storage (RRS).
<i>region</i> -BytesDeleted-SIA	GB	Mensal	A quantidade de dados excluídos por uma operação DeleteObject da classe de armazenamento S3 Standard-IA.
<i>region</i> -BytesDeleted-STANDARD	GB	Mensal	A quantidade de dados excluídos por uma operação DeleteObject da classe de armazenamento S3 Standard.
<i>region</i> -BytesDeleted-ZIA	GB	Mensal	A quantidade de dados excluídos por uma operação DeleteObject da classe de armazenamento S3 One Zone-IA.
<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

Tipo de uso	Unidades	Granularity	Descrição
<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

Tipo de uso	Unidades	Granularity	Descrição
<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
<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.

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -EarlyDelete-GIR-S mObjects	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-S mObjects	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

Tipo de uso	Unidades	Granularity	Descrição
<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
<i>region</i> -Inventory-Objects Listed	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> -MRAP-Out-Bytes	GB	Por hora	A quantidade de dados transferidos para fora dos buckets de uma região por meio de um endpoint de pontos de acesso multirregionais do S3 (preços de roteamento de dados MRAP).

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -MRAP-In-Bytes	GB	Por hora	A quantidade de dados transferidos para fora dos buckets de uma região por meio de um endpoint de pontos de acesso multirregionais do S3 (preços de roteamento de dados MRAP).
<i>regiongroup1-regiongroup2</i> -MRAP-Out-Bytes	GB	Por hora	A quantidade de dados transferidos para fora de um bucket em <i>regiongroup1</i> por meio de um endpoint de pontos de acesso de várias regiões do S3 para um cliente em <i>regiongroup2</i> localizado fora da rede da AWS.
<i>regiongroup1-regiongroup2</i> -MRAP-In-Bytes	GB	Por hora	A quantidade de dados transferidos de um cliente em <i>regiongroup2</i> localizado fora da rede da AWS para um bucket em <i>regiongroup1</i> por meio de um endpoint de pontos de acesso de várias regiões do S3.

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -OverwriteBytes-Copy-GDA	GB	Mensal	A quantidade de dados substituídos por uma operação CopyObject da classe de armazenamento S3 Glacier Deep Archive.
<i>region</i> -OverwriteBytes-Copy-GIR	GB	Mensal	A quantidade de dados substituídos por uma operação CopyObject da classe de armazenamento S3 Glacier Instant Retrieval.
<i>region</i> -OverwriteBytes-Copy-GLACIER	GB	Mensal	A quantidade de dados substituídos por uma operação CopyObject da classe de armazenamento S3 Glacier Flexible Retrieval.
<i>region</i> -OverwriteBytes-Copy-INT	GB	Mensal	A quantidade de dados substituídos por uma operação CopyObject da classe de armazenamento S3 Intelligent-Tiering.
<i>region</i> -OverwriteBytes-Copy-RRS	GB	Mensal	A quantidade de dados sobrescritos por uma operação CopyObject da classe de armazenamento Reduced Redundancy Storage (RRS).

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -OverwriteBytes-Copy-SIA	GB	Mensal	A quantidade de dados substituídos por uma operação CopyObject da classe de armazenamento S3 Standard-IA.
<i>region</i> -OverwriteBytes-Copy-STANDARD	GB	Mensal	A quantidade de dados substituídos por uma operação CopyObject da classe de armazenamento S3 Standard.
<i>region</i> -OverwriteBytes-Copy-ZIA	GB	Mensal	A quantidade de dados substituídos por uma operação CopyObject da classe de armazenamento S3 One Zone-IA.
<i>region</i> -OverwriteBytes-Put-GDA	GB	Mensal	A quantidade de dados substituídos por uma operação PutObject da classe de armazenamento S3 Glacier Deep Archive.
<i>region</i> -OverwriteBytes-Put-GIR	GB	Mensal	A quantidade de dados substituídos por uma operação PutObject da classe de armazenamento S3 Glacier Instant Retrieval.

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -OverwriteBytes-Put-GLACIER	GB	Mensal	A quantidade de dados substituídos por uma operação PutObject da classe de armazenamento S3 Glacier Flexible Retrieval.
<i>region</i> -OverwriteBytes-Put-INT	GB	Mensal	A quantidade de dados substituídos por uma operação PutObject da classe de armazenamento S3 Intelligent-Tiering.
<i>region</i> -OverwriteBytes-Put-RRS	GB	Mensal	A quantidade de dados sobrescritos por uma operação PutObject da classe de armazenamento Reduced Redundancy Storage (RRS).
<i>region</i> -OverwriteBytes-Put-SIA	GB	Mensal	A quantidade de dados substituídos por uma operação PutObject da classe de armazenamento S3 Standard-IA.
<i>region</i> -OverwriteBytes-Put-STANDARD	GB	Mensal	A quantidade de dados substituídos por uma operação PutObject da classe de armazenamento S3 Standard.

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -OverwriteBytes-Put-ZIA	GB	Mensal	A quantidade de dados substituídos por uma operação PutObject da classe de armazenamento S3 One Zone-IA.
<i>region1</i> - <i>region2</i> -S3RTC-In-Bytes	GB	Mensal	A quantidade de dados transferidos para a funcionalidade Controle de Tempo de Replicação do S3 (S3 RTC) de <i>region2</i> para <i>region1</i> .
<i>region1</i> - <i>region2</i> -S3RTC-Out-Bytes	GB	Mensal	A quantidade de dados transferidos para a funcionalidade Controle de Tempo de Replicação do S3 (S3 RTC) de <i>region1</i> para <i>region2</i> .
<i>region</i> -Requests-GDA-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY, POST, CreateMultipartUpload, UploadPart ou CompleteMultipartUpload em objetos da classe S3 Glacier Deep Archive. ⁶
<i>region</i> -Requests-GDA-Tier2	Contagem	Por hora	O número de solicitações GET e HEAD em objetos da classe S3 Glacier Deep Archive.

Tipo de uso	Unidades	Granularity	Descrição
<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 da classe 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 da classe S3 Glacier Instant Retrieval nível 1 em objetos da classe S3 Glacier Instant Retrieval.
<i>region</i> -Requests-GLACIER-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY, POST, CreateMultipartUpload, UploadPart ou CompleteMultipartUpload em objetos da classe S3 Glacier Flexible Retrieval. ⁶
<i>region</i> -Requests-GLACIER-Tier2	Contagem	Por hora	O número de solicitações GET e todas as outras não listadas em objetos da classe S3 Glacier Flexible Retrieval.

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -Requests-INT-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY ou POST em objetos da classe S3 Intelligent-Tiering.
<i>region</i> -Requests-INT-Tier2	Contagem	Por hora	O número de solicitações GET e todas as outras solicitações que não são do nível 1 para objetos da classe S3 Intelligent-Tiering.
<i>region</i> -Requests-SIA-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY ou POST em objetos da classe S3 Standard-IA.
<i>region</i> -Requests-SIA-Tier2	Contagem	Por hora	O número de solicitações GET e todas as outras solicitações que não são da classe S3 Glacier Instant Retrieval nível 1 em objetos da classe S3 Standard-IA.
<i>region</i> -Requests-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY ou POST para S3 Standard, RRS e etiquetas, além de solicitações LIST para todos os buckets e objetos.

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -Requests-Tier2	Contagem	Por hora	O número de solicitações GET e todas as outras solicitações que não são do nível 1.
<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-Tier8	Contagem	Por hora	O número de solicitações da funcionalidade Concessões de Acesso do S3.

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -Requests-XZ-Tier1	Contagem	Por hora	O número de solicitações PUT ou COPY em objetos da classe S3 Express One Zone.
<i>region</i> -Requests-XZ-Tier2	Contagem	Por hora	O número de solicitações GET e todas as outras solicitações que não são da classe S3 Express One Zone nível 1 em objetos da classe S3 Express One Zone.
<i>region</i> -Requests-ZIA-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY ou POST em objetos da classe S3 One Zone-IA.
<i>region</i> -Requests-ZIA-Tier2	Contagem	Por hora	O número de solicitações GET e todas as outras solicitações que não são da classe S3 One Zone-IA nível 1 em objetos da classe S3 One Zone-IA.
<i>region</i> -Retrieval-GIR	GB	Por hora	A quantidade de dados recuperados do armazenamento S3 Glacier Instant Retrieval.
<i>region</i> -Retrieval-SIA	GB	Por hora	A quantidade de dados recuperados do armazenamento S3 Standard – IA

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -Retrieval-XZ	GB	Por hora	A parte dos dados que excede 512 KB em determinada solicitação de recuperação (PUT ou COPY) com a classe de armazenamento S3 Express One Zone.
<i>region</i> -Retrieval-ZIA	GB	Por hora	A quantidade de dados recuperados do armazenamento S3 One Zone-IA
<i>region</i> -S3DSSE-In-Bytes	GB	Mensal	A quantidade de dados com criptografia dupla do Amazon S3.
<i>region</i> -S3DSSE-Out-Bytes	GB	Mensal	A quantidade de dados com criptografia dupla descriptografados pelo Amazon S3.
<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

Tipo de uso	Unidades	Granularity	Descrição
<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

Tipo de uso	Unidades	Granularity	Descrição
<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
<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

Tipo de uso	Unidades	Granularity	Descrição
<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	Por dia	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	Por dia	O número de objetos exclusivos em cada painel do S3 Storage Lens que são monitorados pelas métricas de uso do S3 Storage Lens.
StorageObjectCount	Contagem	Diariamente	O número de objetos armazenados em um determinado bucket
<i>region</i> -TagStorage-TagHrs	Tag-Hours	Diariamente	O total de tags em todos os objetos no bucket informados por hora

Tipo de uso	Unidades	Granularity	Descrição
<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	Por dia	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	Por dia	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	Por dia	O número de GB-horas em que os dados ficaram no armazenamento S3 Glacier Flexible Retrieval

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -TimedStorage-GlacierStaging	GB-Horas	Por dia	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 Frequent Access da classe de armazenamento S3 Intelligent-Tiering. ⁵
<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	Por dia	O número de GB-horas em que os dados permaneceram armazenados no nível de acesso de arquivamento do armazenamento S3 Intelligent-Tiering


Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -TimedStorage-INT-AIA-ByteHrs	GB-Horas	Por dia	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	Por dia	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

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -TimedStorage-XZ-ByteHrs	GB-Horas	Por dia	O número de GB-horas em que os dados foram armazenados na classe de armazenamento S3 Express One Zone.
<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
<i>region</i> -TimedStorage-ZIA-SmObjects	GB-Horas	Por dia	O número de GB-horas em que pequenos objetos (menores que 128 KB) ficaram na classe de armazenamento S3 One Zone-IA.
<i>region</i> -Upload-XZ	GB	Por hora	A quantidade de dados que excede 512 KB em determinada solicitação de upload (PUT ou COPY) com a classe S3 Express One Zone.

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 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 as classes S3 Glacier Instant Retrieval e S3 Glacier Flexible Retrieval ou de 180 dias para a classe S3 Glacier Deep Archive, há uma cobrança pro rata por gigabyte para os dias restantes.
3. Para objetos que estão na classe de 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 proporcional por gigabyte pelos dias restantes.
 4. Para pequenos objetos (menores que 128 KB) que estão na classe de 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 proporcional 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.
 6. Quando você inicia uma solicitação `CreateMultipartUpload`, `UploadPart` ou `UploadPartCopy` para as classes de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, as solicitações são cobradas de acordo com as taxas de solicitação da classe de armazenamento S3 Standard até que você conclua o upload multiparte. Quando o upload for concluído, a única solicitação `CompleteMultipartUpload` será cobrada de acordo com a taxa de PUT da classe de armazenamento S3 Glacier de destino. As partes de um upload multiparte em andamento para uma PUT na classe de armazenamento S3 Glacier Flexible Retrieval são cobradas como S3 Glacier Flexible Retrieval Staging Storage de acordo com as taxas da classe de armazenamento S3 Standard até que o upload seja concluído. De maneira semelhante, as partes de um upload multiparte em andamento para uma PUT na classe de armazenamento S3 Glacier Deep Archive são cobradas como S3 Glacier Deep Archive Staging Storage de acordo com as taxas da classe de armazenamento S3 Standard até que o upload seja concluído.
 7. A classe de armazenamento S3 Express One Zone aplica uma taxa fixa por solicitação para solicitações de até 512 KB. Uma taxa adicional por GB é aplicada para a parte que exceder 512 KB em solicitações PUT e GET.
 8. Para obter informações sobre os recursos compatíveis com a classe de armazenamento S3 Express One Zone, consulte [Recursos do Amazon S3 não compatíveis com a classe S3 Express One Zone](#).
 9. Os tipos de uso com unidades cobrados em GB são calculados em bytes nos relatórios de uso.

 Note

Em geral, os proprietários de bucket do S3 são cobrados pelas solicitações com respostas HTTP 200 OK bem-sucedidas e respostas de erro 4XX do cliente HTTP. Os proprietários de bucket não são cobrados pelas respostas de erro 5XX do servidor HTTP, como erros HTTP 503 Slow Down. Consulte mais informações sobre códigos de erro do S3 em HTTP 3XX e códigos de status 4XX que não são cobrados em [Cobrança pelas respostas de erro do Amazon S3](#). Consulte mais informações sobre as cobranças de um bucket de pagamento pelo solicitante em [Como funcionam as cobranças de Pagamento pelo solicitante](#).


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](#)
- [Relatórios do AWS Billing para o Amazon S3](#)
- [Preços do Amazon S3](#)
- [Perguntas frequentes sobre o Amazon S3](#)

Cobrança pelas respostas de erro do Amazon S3

 Important

Em 13 de maio de 2024, começamos a implantar uma alteração para eliminar cobranças por solicitações não autorizadas que não foram iniciadas pelo proprietário do bucket. Depois que a implantação dessa alteração for concluída, os proprietários do bucket nunca incorrerão em cobranças de solicitação ou largura de banda para solicitações que retornem erros AccessDenied (HTTP 403 Forbidden) quando forem iniciadas de fora de sua conta da AWS individual ou de organização da AWS. A página atual mostra uma lista completa de códigos de status HTTP 3XX e de 4XX que não serão cobrados. Essa alteração de cobrança

não exige atualizações nas aplicações e é destinada a todos os buckets do S3. Quando a implantação dessa alteração for concluída em todas as Regiões da AWS, atualizaremos nossa documentação.

Em geral, os proprietários de bucket do S3 são cobrados pelas solicitações com respostas HTTP 200 OK bem-sucedidas e respostas de erro 4XX do cliente HTTP. Os proprietários de bucket não são cobrados pelas respostas de erro 5XX do servidor HTTP, como erros HTTP 503 Slow Down. Consulte mais informações sobre as cobranças de um bucket de pagamento pelo solicitante em [Como funcionam as cobranças de Pagamento pelo solicitante](#).

A tabela a seguir lista códigos de erro específicos em HTTP 3XX e códigos de status 4XX que não são cobrados. Para buckets configurados com hospedagem de sites, a solicitação aplicável e outras cobranças ainda serão aplicadas quando o S3 retornar um [documento de erro personalizado](#) ou para redirecionamentos personalizados.

Note

Para AccessDenied (HTTP 403 Forbidden), o S3 não cobra do proprietário do bucket quando a solicitação é iniciada fora da conta da AWS individual dele ou da organização da AWS à qual ele pertence.

Código de status HTTP	Código de erro	Descrição do código de erro
301 Moved Permanently (301 movido permanentemente)	PermanentRedirect	O bucket que você está tentando acessar deve ser endereçado usando o endpoint especificado. Envie todas as solicitações futuras para esse endpoint.
	PermanentRedirectControlError	A operação de API que você está tentando

Código de status HTTP	Código de erro	Descrição do código de erro	
		acessar deve ser endereçada usando o endpoint especificado. Envie todas as solicitações futuras para esse endpoint.	
307 Redirecionamento temporário	TemporaryRedirect	Será feito um redirecionamento ao bucket enquanto o servidor Sistema de Nomes de Domínio (DNS) está sendo atualizado.	
400 solicitação inválida	AuthorizationHeaderMalformed	O cabeçalho de autorização que você forneceu não é válido.	
	AuthorizationQueryParametersError	Os parâmetros da consulta de autorização que você forneceu não são válidos.	
	ExpiredToken	O token fornecido expirou.	

Código de status HTTP	Código de erro	Descrição do código de erro	
	IllegalLocationConstraintException	Você está tentando acessar um bucket de uma região diferente daquela em que o bucket existe. Para evitar esse erro, use a opção <code>--region</code> . Por exemplo: <pre>aws s3 cp <i>awsexample.txt</i> s3://<i>DOC-EXAMPLE-BUCKET</i> / --region <i>ap-east-1</i> .</pre>	
	InvalidArgument	Esse erro poderá ocorrer pelos seguintes motivos: <ul style="list-style-type: none">• O argumento especificado não é válido.• A solicitação não tinha o cabeçalho exigido.• O argumento especificado estava incompleto ou no formato errado.• O argumento especificado deve ter um comprimento maior ou igual a três.	

Código de status HTTP	Código de erro	Descrição do código de erro	
	InvalidDigest	O valor de Content-MD5 ou da soma de verificação que você especificou não é válido.	
	InvalidEncryptionAlgorithmError	A solicitação de criptografia que você especificou não é válida. O valor válido é AES256.	

Código de status HTTP	Código de erro	Descrição do código de erro	
	InvalidRequest	<p>Esse erro poderá ocorrer pelos seguintes motivos:</p> <ul style="list-style-type: none">• A solicitação está usando a versão incorreta da assinatura a. Use o AWS4-HMAC-SHA256 (Signature versão 4).• Um ponto de acesso só pode ser criado para um bucket existente.• O ponto de acesso não está em um estado em que possa ser excluído.• Um ponto de acesso só pode ser listado para um bucket existente.• O próximo token não é válido.• Pelo menos uma ação deve ser especificada em uma regra de ciclo de vida.• Pelo menos uma regra de ciclo de vida deve ser especificada.	

Código de status HTTP	Código de erro	Descrição do código de erro	
		<ul style="list-style-type: none">• O número de regras de ciclo de vida não deve exceder o limite permitido de 1.000 regras.• O intervalo do parâmetro <code>MaxResults</code> não é válido.• As solicitações SOAP devem ser feitas por meio de uma conexão HTTPS.• A Aceleração de Transferências do Amazon S3 não é compatível com buckets com nomes que não estejam em conformidade com o DNS.• A Aceleração de Transferências do Amazon S3 não é compatível com buckets com ponto (.) no nome.• O endpoint da Aceleração de	

Código de status HTTP	Código de erro	Descrição do código de erro	
		<p>Transferências do Amazon S3 só permite solicitações no estilo virtual.</p> <ul style="list-style-type: none">• A Aceleração de Transferências do Amazon S3 não está configurada neste bucket.• A Aceleração de Transferências do Amazon S3 está desabilitada neste bucket.• A Aceleração de Transferências do Amazon S3 não é compatível com este bucket. Para obter ajuda, entre em contato com AWS Support.• A Aceleração de Transferências do Amazon S3 não pode ser habilitada neste bucket. Para obter ajuda, entre em contato com AWS Support.•	

Código de status HTTP	Código de erro	Descrição do código de erro	
		<p>Valores conflitantes fornecidos em cabeçalhos HTTP e parâmetros de consulta.</p> <ul style="list-style-type: none">• Valores conflitantes fornecidos nos cabeçalhos HTTP e nos campos do formulário POST.• Solicitação CopyObject feita em objetos maiores que 5 GB.	
	InvalidSOAPRequest	O corpo da solicitação SOAP não é válido.	
	InvalidStorageClass	A classe de armazenamento que você especificou não é válida.	
	InvalidTag	Sua solicitação contém uma entrada de tag que não é válida. Por exemplo, sua solicitação pode conter chaves duplicadas, chaves ou valores muito longos ou tags do sistema.	

Código de status HTTP	Código de erro	Descrição do código de erro	
	InvalidToken	O token fornecido foi malformado ou é inválido por algum outro motivo.	
	InvalidURI	Não foi possível analisar o URI especificado.	
	KeyTooLongError	A chave é muito longa.	
	MalformedACLError	O ACL fornecido não foi bem formado ou não foi validado com relação a nosso esquema publicado.	
	MalformedPOSTRequest	O corpo da solicitação POST não tem dados de formulários/de várias partes bem formados.	
	MalformedXML	O XML fornecido não foi bem formado ou não foi validado com relação a nosso esquema publicado.	
	MaxPostPreDataLengthExceededError	Os campos de solicitação POST anteriores ao arquivo de upload eram muito grandes.	

Código de status HTTP	Código de erro	Descrição do código de erro	
	MetadataTooLarge	Os cabeçalhos de metadados excedem o tamanho máximo permitido de metadados.	
	MissingRequestBodyError	Você enviou um documento XML vazio como uma solicitação.	
	MissingSecurityHeader	Falta um cabeçalho obrigatório na solicitação.	
	NoLoggingStatusForKey	Não existe um sub-recurso de status de registro em log para uma chave.	
	RequestHeaderSectionTooLarge	O cabeçalho da solicitação e os parâmetros de consulta usados para fazer a solicitação excedem os tamanhos máximos permitidos	
	UnexpectedContent	Essa solicitação contém um conteúdo incompatível.	
	UserKeyMustBeSpecified	A solicitação POST do bucket deve conter o nome do campo especificado. Se ele estiver especificado, confira a ordem dos campos.	

Código de status HTTP	Código de erro	Descrição do código de erro	
	IncorrectEndpoint	O bucket especificado existe em outra região. Direcione as solicitações ao endpoint correto.	
403 proibido	RequestTimeTooSkewed	A diferença entre a hora da solicitação e a hora do servidor é muito grande.	
	SignatureDoesNotMatch	A assinatura da solicitação que o servidor calculou não corresponde à assinatura que você forneceu. Verifique a chave de acesso secreta da AWS e o método de assinatura. Consulte mais informações em Assinar e autenticar as solicitações REST e em Autenticar solicitações SOAP .	
	NotSignedUp	Sua conta não está cadastrada no serviço do Amazon S3. É necessário o se cadastrar para poder usar o Amazon S3. Você pode se cadastrar no seguinte URL: https://aws.amazon.com/s3	

Código de status HTTP	Código de erro	Descrição do código de erro	
	InvalidSecurity	As credenciais de segurança fornecidas não são válidas.	
	InvalidPayer	Todo acesso a esse objeto foi desabilitado. Se precisar de assistência adicional, acesse Fale conosco .	
	InvalidAccessKeyId	O ID de chave de acesso da AWS fornecido não existe em nossos registros.	
	AccountProblem	Há um problema em sua Conta da AWS que impede a conclusão bem-sucedida da operação. Se precisar de assistência adicional, acesse Fale conosco .	
	UnauthorizedAccessError	Aplicável somente nas regiões da China. Retornado quando uma solicitação é feita para um bucket que não tem uma licença ICP. Consulte mais informações em ICP Recordal .	

Código de status HTTP	Código de erro	Descrição do código de erro	
404 Not Found (404 Não encontrado)	NoSuchUpload	O carregamento fracionado especificado não existe. O ID do carregamento pode não ser válido ou o carregamento fracionado pode ter sido anulado ou concluído.	
	NoSuchWebsiteConfiguration	O bucket especificado não tem uma configuração de site.	
405 Método não permitido	MethodNotAllowed	O método especificado não é permitido neste recurso.	
409 Conflito	BucketAlreadyExists	O nome do bucket solicitado não está disponível. O namespace do bucket é compartilhado por todos os usuários do sistema. Especifique um nome diferente e tente novamente.	
	InvalidBucketState	A solicitação não é válida para o estado atual do bucket.	

Código de status HTTP	Código de erro	Descrição do código de erro
	OperationAborted	Uma operação condicional conflitante está em andamento neste recurso. Tente novamente.
411 Comprimento obrigatório	MissingContentLength	Você deve fornecer o cabeçalho HTTP de tamanho do conteúdo.
412 Falha na pré-condição	RequestIsNotMultiPartContent	Uma solicitação POST de bucket deve ser do tipo dados de formulário/de várias partes.

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 só permite que você consulte um objeto por vez. Ele funciona em um objeto armazenado em formato CSV, JSON ou Apache Parquet. Ele também funciona em um objeto compactado com GZIP ou BZIP2 (somente para objetos CSV e JSON) e um objeto criptografado do lado do servidor. Você pode especificar o formato dos resultados como CSV ou JSON e determinar como os registros do resultado são delimitados.

Expressões SQL são passadas para o Amazon S3 na solicitação. O Amazon S3 Select é compatível com um subconjunto de SQL. Para obter mais informações sobre os elementos SQL compatíveis com o Amazon S3 Select, consulte [Referência SQL para o Amazon S3 Select](#).

É possível realizar consultas SQL usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI), a operação de API REST `SelectObjectContent` ou os SDKs da AWS.

Note

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 consultado for criptografado por meio da criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C), será necessário usar `https` e fornecer a chave de criptografia na solicitação.

Os seguintes limites se aplicam ao usar o Amazon S3 Select:

- O S3 Select pode consultar somente um objeto por solicitação.
- 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 consultar um objeto armazenado nas classes de armazenamento S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive ou Reduced Redundancy Storage (RRS). Também não é possível consultar um objeto armazenado no nível S3 Intelligent-Tiering Archive Access ou S3 Intelligent-Tiering Deep Archive Access. Para obter mais informações sobre classes de armazenamento, consulte [Uso de classes de armazenamento do Amazon S3](#).

Limitações adicionais se aplicam ao usar o Amazon S3 Select com um objeto Parquet:

- O Amazon S3 Select aceita somente a compactação colunar com GZIP ou Snappy. O Amazon S3 Select não aceita a compactação de objetos inteiros no caso de um objeto 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 consultado usando um objeto `InputSerialization`. Forneça detalhes sobre como os resultados serão retornados usando um objeto `OutputSerialization`. Inclua também a expressão SQL que o Amazon S3 usa para filtrar a solicitação.

Para obter mais informações sobre como criar uma solicitação do Amazon S3 Select, consulte [SelectObjectContent](#) na Referência de API 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 ultrapassa o intervalo de verificação, será processado pela consulta. Por exemplo, veja abaixo 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
```

Suponha que você esteja usando 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. A solicitação de intervalo de verificação retornará o resultado C, D porque 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) ou 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 verificação do Amazon S3 Select estão disponíveis para uso com a AWS CLI, a API do Amazon S3 e AWS SDKs. É possível usar o parâmetro `ScanRange` na solicitação de Seleção do Amazon S3 desse recurso. Para obter mais informações, consulte [SelectObjectContent](#) na Referência da API 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 a seguir.

Tópicos

- [Exemplos de uso do Amazon S3 Select em um objeto](#)
- [Referência SQL para o Amazon S3 Select](#)

Exemplos de uso do Amazon S3 Select em um objeto

É possível usar o S3 Select para selecionar conteúdo de um objeto usando o console do Amazon S3, a API REST e SDKs da AWS.

Para receber mais informações sobre as funções SQL compatíveis com o S3 Select, consulte [Funções SQL](#).

Uso do console do S3

Como selecionar o conteúdo de um objeto no 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. No painel de navegação à esquerda, escolha Buckets.
3. Escolha o bucket que contém o objeto do qual você deseja selecionar o conteúdo; depois, escolha o nome do objeto.
4. Escolha Ações de objeto e Consultar com o S3 Select.
5. Defina as Configurações de entrada, com base no formato dos dados de entrada.
6. Defina as Configurações de saída, com base no formato da saída que você deseja receber.
7. Para extrair registros do objeto escolhido, em Consulta SQL, insira os comandos SQL SELECT. Para obter mais informações sobre como escrever comandos SQL, consulte [Referência SQL para o Amazon S3 Select](#).
8. Depois de inserir consultas SQL, escolha Executar consulta SQL. Depois, em Resultados da consulta, é possível ver os resultados das consultas SQL.

Uso da API REST

É possível usar SDKs da AWS para selecionar conteúdo de um objeto. 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 [SelectObjectContent](#).

Uso dos AWS SDKs

É possível usar o Amazon S3 Select para selecionar parte do conteúdo de um objeto usando o método `selectObjectContent`. Se esse método for bem-sucedido, ele retornará 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 criar e testar um exemplo funcional, consulte [Testar exemplos de código Java no Amazon S3](#).

```
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)
            {
                setResultComplete.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 que usa o AWS SDK for JavaScript com a operação de API `SelectObjectContent` do S3 para selecionar registros dos arquivos JSON e CSV armazenados no Amazon S3, consulte a publicação de blog [Adição de suporte ao Amazon S3 Select no AWS SDK for JavaScript](#).

Python

Para obter um exemplo do Python sobre o uso de consultas SQL para pesquisar dados carregados no Amazon S3 como um arquivo de valor separado por vírgulas (CSV) usando o S3 Select, consulte a publicação do blog [Consultar dados sem servidores ou bancos de dados usando o Amazon S3 Select](#).

Referência SQL para o Amazon S3 Select

Essa referência contém uma descrição de elementos de linguagem de consulta estruturada (SQL) que são compatíveis com o Amazon S3 Select.

Tópicos

- [SELECT command](#)
- [Tipos de dados](#)
- [Operadores](#)

- [Palavras-chave reservadas](#)
- [Funções SQL](#)

SELECT command

O Amazon S3 Select só oferece suporte ao comando SQL SELECT. As seguintes cláusulas padrão ANSI são compatíveis com SELECT:

- SELECT lista
- FROMCláusula
- WHERECláusula
- LIMITCláusula

Note

As consultas do Amazon S3 Select não oferecem suporte a subconsultas ou junções no momento.

SELECT lista

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 projection1 AS column_alias_1, projection2 AS column_alias_2
```

O primeiro formulário de SELECT com o * (asterisco) retorna todas as linhas que passaram na cláusula WHERE, da maneira como estão. O segundo formulário de SELECT cria uma linha com expressões escalares de saída definidas pelo usuário *projection1* e *projection2* para cada coluna.

FROMCláusula

O Amazon S3 Select oferece suporte aos seguintes formatos de cláusula FROM:

```
FROM table_name
```



```
FROM table_name alias  
FROM table_name AS alias
```

Em cada forma da cláusula FROM, *table_name* é o S3Object que está sendo consultado. Os usuários provenientes de bancos de dados relacionais tradicionais podem pensar nisso como um esquema de banco de dados que contém várias visualizações em uma tabela.

Seguindo o SQL padrão, a cláusula FROM cria linhas filtradas na cláusula WHERE e projetadas na lista SELECT.

Para objetos JSON armazenados no 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. É possível especificar path usando uma das formas a seguir:

- Por nome (em um objeto): *.name* ou [*'name'*]
- Por índice (em uma matriz): [*index*]
- Por caractere curinga (em um objeto): *.**
- Por caractere curinga (em uma matriz): [***]

Note

- Essa forma da cláusula FROM funciona apenas com objetos JSON.
- Caracteres 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 execução da consulta), o Amazon S3 Select substituirá os valores MISSING por registros vazios.
- Funções agregadas (AVG, COUNT, MAX, MIN, and SUM) ignoram valores MISSING.
- Se não fornecer um alias ao usar um caractere curinga, você 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 caractere 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 caractere 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 caractere curinga. Portanto, `FROM S3Object` e `FROM S3Object[*].path` são cláusulas válidas, mas `FROM S3Object.path` não.

Example

Exemplos:

Exemplo 1

Este exemplo mostra resultados ao usar 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 teve correspondência com um registro. Portanto, o Amazon S3 Select 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 Amazon S3 Select 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 ao usar o seguinte conjunto de dados e consultas:

```
{ "created": "936864000", "dir_name": "important_docs", "files": [ { "name": "." },
  { "name": ".." }, { "name": ".aws" }, { "name": "downloads" } ], "owner": "Amazon
  S3" }
{ "created": "936864000", "dir_name": "other_docs", "files": [ { "name": "." },
  { "name": ".." }, { "name": "my stuff" }, { "name": "backup" } ], "owner": "User" }
```

```
SELECT d.dir_name, d.files FROM S3object[*] d
```

```
{"dir_name":"important_docs","files":[{"name":"."},{"name":".."},{"name":".aws"},
{"name":"downloads"}]}
{"dir_name":"other_docs","files":[{"name":"."},{"name":".."},{"name":"my stuff"},
{"name":"backup"}]}
```

```
SELECT _1.dir_name, _1.owner FROM S3object[*]
```

```
{"dir_name":"important_docs","owner":"Amazon S3"}
{"dir_name":"other_docs","owner":"User"}
```

WHERECláusula

A cláusula WHERE segue esta sintaxe:

```
WHERE condition
```

A cláusula WHERE filtra as linhas com base na *condition*. Uma condição é uma expressão com um valor booleano. Somente linhas para as quais a condição é avaliada como TRUE são retornadas no resultado.

LIMITCláusula

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 *number*.

Acesso ao 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 coluna Nth de uma linha com o nome da coluna *_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

- Documento — você pode acessar os campos do documentos JSON como *alias.name*. Também é possível acessar os campos aninhados, por exemplo, *alias.name1.name2.name3*.

- Lista: é possível 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]. É possível combinar elementos da lista de acesso com campos, por exemplo, *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 e nomes de atributo

Com o Amazon S3 Select, você pode usar aspas duplas para indicar que cabeçalhos de coluna (para objetos CSV) e atributos (para objetos JSON) fazem diferenciação entre letras maiúsculas e minúsculas. Sem as aspas duplas, os cabeçalhos e atributos de objeto não diferenciam letras maiúsculas de minúsculas. Um erro ocorre em casos de ambiguidade.

Os exemplos a seguir são 1) objetos do Amazon S3 no formato CSV com os cabeçalhos de coluna especificados e com `FileHeaderInfo` definido como "Use" para a solicitação de consulta; ou 2) objetos do Amazon S3 no formato JSON com os atributos especificados.

Exemplo 1: o objeto consultado tem o cabeçalho ou atributo `NAME`.

- A expressão a seguir retorna com êxito valores do objeto. Como não há aspas, a consulta não diferencia letras maiúsculas de minúsculas.

```
SELECT s.name from S3object s
```

- A expressão a seguir resulta em um erro 400 `MissingHeaderName`. Como há aspas, a consulta diferencia letras maiúsculas de minúsculas.

```
SELECT s."name" from S3object s
```

Exemplo 2: o objeto do Amazon S3 que está sendo consultado tem um cabeçalho ou atributo com `NAME` e outro cabeçalho/atributo com `name`.

- A expressão a seguir resulta em um erro 400 `AmbiguousFieldName`. Como não há aspas, a consulta não diferencia letras maiúsculas de minúsculas, mas há duas correspondências, então o erro é gerado.

```
SELECT s.name from S3object s
```

- A expressão a seguir retorna com êxito valores do objeto. Como há aspas, a consulta diferencia letras maiúsculas de minúsculas, portanto, não há ambiguidade.

```
SELECT s."NAME" from S3object s
```

Usar palavras-chave reservadas como termos definidos pelo usuário

O Amazon S3 Select possui 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 objetos 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 reservadas, consulte [Palavras-chave reservadas](#).

O exemplo a seguir é 1) um objeto do Amazon S3 no formato CSV com os cabeçalhos de coluna especificados, com `FileHeaderInfo` definido como "Use" para a solicitação de consulta ou 2) um objeto do Amazon S3 no formato JSON com os atributos especificados.

Exemplo: o objeto consultado tem o cabeçalho ou atributo nomeado como `CAST`, que é uma palavra-chave reservada.

- A expressão a seguir retorna com êxito valores do objeto. Como as aspas são usadas na consulta, o S3 Select usa o cabeçalho ou atributo definido pelo usuário.

```
SELECT s."CAST" from S3object s
```

- A expressão a seguir resulta em um erro de análise 400. Como nenhuma aspa é usada na consulta, `CAST` colide com uma 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 formulário *column_name* ou *alias.column_name*.

- ***unary_op expression***

Nesse caso, ***unary_op*** é um operador unário SQL.

- ***expression binary_op expression***

Nesse caso, ***binary_op*** é um operador binário SQL.

- ***func_name***

Nesse caso, ***func_name*** é o nome da função escalar a ser invocada.

- ***expression*** [NOT] BETWEEN ***expression*** AND ***expression***
- ***expression*** LIKE ***expression*** [ESCAPE ***expression***]

Tipos de dados

O Amazon S3 Select oferece suporte a vários tipos de dados primitivos.

Conversões de tipos de dados


A regra geral é seguir a função CAST se for definida. Se CAST não estiver definido, todos os dados de entrada serão tratados como uma string. Nesse caso, é necessário converter seus dados de entrada em tipos de dados relevantes quando necessário.

Para obter mais informações sobre a função CAST, consulte [CAST](#).

Tipos de dados compatíveis

O Amazon S3 Select oferece suporte ao conjunto a seguir de tipos de dados primitivos.

Name (Nome)	Descrição	Exemplos
bool	Um valor booleano, TRUE ou FALSE.	FALSE
int, integer	Um número inteiro assinado de 8 bytes no intervalo de -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807.	100000
string	Uma string de tamanho variável codificada por UTF8. O limite padrão é de um caractere. O limite máximo de caracteres é 2.147.483.647.	'xyz'
float	Um número de ponto flutuante de 8 bytes.	CAST(0.456 AS FLOAT)
decimal, numeric	Um número de base 10, com precisão máxima de 38 (ou seja, a quantidade máxima de dígitos significativos) e com	123.456


Name (Nome)	Descrição	Exemplos
	<p>uma escala no intervalo de -2^{31} a $2^{31}-1$ (ou seja, o expoente de base 10).</p> <div data-bbox="375 331 1247 556" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>O Amazon S3 Select ignora a escala e a precisão quando as duas são fornecidas ao mesmo tempo.</p> </div>	
timestamp	<p>Os carimbos de data/hora representam um momento específico, sempre incluem um deslocamento local e são capazes de oferecer precisão arbitrária.</p> <p>No formato de texto, os carimbos de data/hora seguem a nota W3C sobre formatos de data e hora, mas devem terminar com o literal T, se os carimbos de data/hora não forem 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. Os deslocamentos de horário local são necessários em carimbos de data/hora com hora e não são permitidos em valores de data.</p>	<pre>CAST('2007-04-05T14:30Z' AS TIMESTAMP)</pre>

Tipos compatíveis de Parquet

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 LIST do Parquet, 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 de 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 usam 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 do Parquet	Tipos de dados compatíveis
DATE	timestamp
DECIMAL	decimal, numeric
ENUM	string
INT(8)	int, integer
INT(16)	int, integer

Tipos do Parquet	Tipos de dados compatíveis
INT(32)	<code>int</code> , <code>integer</code>
INT(64)	<code>decimal</code> , <code>numeric</code>
LIST	Cada tipo de Parquet na lista é mapeado para o tipo de dados correspondente.
STRING	<code>string</code>
TIMESTAMP	<code>timestamp</code>

Operadores

O Amazon S3 Select é compatível 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, como indicado a seguir:

- +
- -
- *
- /
- %

Precedência do operador

A tabela a seguir mostra a precedência dos operadores em ordem decrescente.

Operador ou elemento	Capacidade de associação	Obrigatório
-	direita	menos unário

Operador ou elemento	Capacidade de associação	Obrigatório
*, /, %	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
OR	esquerda	disjunção lógica

Palavras-chave reservadas

Veja a seguir a lista de palavras-chave reservadas para o Amazon S3 Select. Essas palavras-chave incluem os nomes de função, tipos de dados, operadores e assim por diante, 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 é compatível com as funções SQL a seguir.

Tópicos

- [Funções agregadas](#)
- [Funções condicionais](#)
- [Funções de conversão](#)
- [Funções de data](#)
- [Funções de string](#)

Funções agregadas

O Amazon S3 Select é compatível com as seguintes funções agregadas.

Função	Tipo de argumento	Tipo de retorno
AVG(<i>expressão</i> <i>n</i>)	INT, FLOAT, DECIMAL	DECIMAL para um argumento INT, FLOAT ou um argumento de ponto flutuante, caso contrário, é igual ao tipo de dados do argumento.
COUNT	-	INT
MAX(<i>expressão</i> <i>n</i>)	INT, DECIMAL	O mesmo que o tipo de argumento.
MIN(<i>expressão</i> <i>n</i>)	INT, DECIMAL	O mesmo que o tipo de argumento.
SUM(<i>expressão</i> <i>n</i>)	INT, FLOAT, DOUBLE, DECIMAL	INT para um argumento INT, FLOAT ou um argumento de ponto flutuante; do contrário, é igual ao tipo de dados do argumento.

Exemplo de SUM

Para agregar os tamanhos totais de objetos de uma pasta em um [relatório do S3 Inventory](#), use uma expressão SUM.

O relatório do S3 Inventory a seguir é um arquivo CSV compactado com GZIP. Existem três colunas.

- A primeira coluna é o nome do bucket do S3 (*DOC-EXAMPLE-BUCKET*) para o qual o relatório do S3 Inventory se destina.
- A segunda coluna é o nome da chave do objeto que identifica exclusivamente o objeto no bucket.

O valor *example-folder/* na primeira linha é para a pasta *example-folder*. No Amazon S3, quando você cria uma pasta no bucket, o S3 cria um objeto de 0 byte com uma chave definida como o nome da pasta que você forneceu.

O valor *example-folder/object1* na segunda linha é para o objeto *object1* na pasta *example-folder*.

O valor *example-folder/object2* na terceira linha é para o objeto *object2* na pasta *example-folder*.

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

- A terceira coluna é o tamanho do objeto em bytes.

```
"DOC-EXAMPLE-BUCKET","example-folder/","0"  
"DOC-EXAMPLE-BUCKET","example-folder/object1","2011267"  
"DOC-EXAMPLE-BUCKET","example-folder/object2","1570024"
```

Para usar uma expressão SUM para calcular o tamanho total da pasta *example-folder*, execute a consulta SQL com o Amazon S3 Select.

```
SELECT SUM(CAST(_3 as INT)) FROM s3object s WHERE _2 LIKE 'example-folder/%' AND _2 !=  
'example-folder/';
```

Resultado da consulta:

```
3581291
```

Funções condicionais

O Amazon S3 Select é compatível com as seguintes funções condicionais.

Tópicos

- [CASE](#)
- [COALESCE](#)
- [NULLIF](#)

CASE

A expressão CASE é uma expressão condicional, semelhante às instruções `if/then/else` encontradas em outras linguagens. CASE é usado para especificar um resultado com 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 booliana e a instrução CASE retorna o primeiro CASE correspondente. Se nenhuma correspondência de CASE for encontrada entre as cláusulas WHEN, a ação na cláusula ELSE será 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.

A seguir está uma declaração CASE simples usada para corresponder às condições:

```
CASE expression WHEN value THEN result [WHEN...] [ELSE result] END
```

Veja a seguir uma declaração CASE pesquisada usada para avaliar 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 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 other.

```
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

COALESCE avalia os argumentos na ordem e retorna o primeiro valor não desconhecido, ou seja, o primeiro valor que não for nulo ou ausente. Essa função não propaga valores nulos e ausentes.

Sintaxe

```
COALESCE ( expression, expression, ... )
```

Parâmetros

expression

A expressão de destino na qual a função opera.

Exemplos

```
COALESCE(1)           -- 1
COALESCE(null)        -- null
COALESCE(null, null)  -- null
COALESCE(missing)     -- null
COALESCE(missing, missing) -- null
COALESCE(1, null)     -- 1
COALESCE(null, null, 1) -- 1
COALESCE(null, 'string') -- 'string'
COALESCE(missing, 1)  -- 1
```

NULLIF

Dadas as duas expressões, NULLIF retorna NULL se as duas forem avaliadas para o mesmo valor. Caso contrário, NULLIF retorna o resultado da avaliação da primeira expressão.

Sintaxe

```
NULLIF ( expression1, expression2 )
```

Parâmetros

expression1, *expression2*

As expressões de destino nas quais a função opera.

Exemplos

```
NULLIF(1, 1)           -- null
NULLIF(1, 2)           -- 1
NULLIF(1.0, 1)         -- null
NULLIF(1, '1')         -- 1
NULLIF([1], [1])       -- null
NULLIF(1, NULL)        -- 1
NULLIF(NULL, 1)        -- null
NULLIF(null, null)     -- null
NULLIF(missing, null)  -- null
NULLIF(missing, missing) -- null
```

Funções de conversão

O Amazon S3 Select é compatível com a seguinte função de conversão.

Tópicos

- [CAST](#)

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

expression

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](#).

Exemplos

```
CAST('2007-04-05T14:30Z' AS TIMESTAMP)
CAST(0.456 AS FLOAT)
```

Funções de data

O Amazon S3 Select é compatível com as seguintes funções de data.

Tópicos

- [DATE_ADD](#)
- [DATE_DIFF](#)
- [EXTRACT](#)
- [TO_STRING](#)
- [TO_TIMESTAMP](#)
- [UTCNOW](#)

DATE_ADD

Dada uma parte da data, uma quantidade e um carimbo de data/hora, DATE_ADD retorna um carimbo de data/hora 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

O valor a ser aplicado a um carimbo de data/hora atualizado. Os valores positivos para a *quantity* são adicionados à *date_part* do carimbo de data/hora e os valores negativos são subtraídos.

timestamp

O carimbo de data/hora de destino no qual a função opera.

Exemplos

```
DATE_ADD(year, 5, `2010-01-01T`)           -- 2015-01-01 (equivalent to
2015-01-01T)
DATE_ADD(month, 1, `2010T`)               -- 2010-02T (result will add precision
as necessary)
DATE_ADD(month, 13, `2010T`)              -- 2011-02T
DATE_ADD(day, -1, `2017-01-10T`)         -- 2017-01-09 (equivalent to
2017-01-09T)
DATE_ADD(hour, 1, `2017T`)                -- 2017-01-01T01:00-00:00
DATE_ADD(hour, 1, `2017-01-02T03:04Z`)   -- 2017-01-02T04:04Z
DATE_ADD(minute, 1, `2017-01-02T03:04:05.006Z`) -- 2017-01-02T03:05:05.006Z
DATE_ADD(second, 1, `2017-01-02T03:04:05.006Z`) -- 2017-01-02T03:04:06.006Z
```

DATE_DIFF

Dada uma parte da data e dois carimbos de data/hora válidos, DATE_DIFF 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 carimbos de data/hora deve ser comparada. Para a definição de *date_part*, consulte [DATE_ADD](#).

timestamp1

O primeiro carimbo de data/hora a ser comparado.

timestamp2

O segundo carimbo de data/hora a ser comparado.

Exemplos

```
DATE_DIFF(year, `2010-01-01T`, `2011-01-01T`)           -- 1
DATE_DIFF(year, `2010T`, `2010-05T`)                   -- 4 (2010T is equivalent to
2010-01-01T00:00:00.000Z)
DATE_DIFF(month, `2010T`, `2011T`)                     -- 12
DATE_DIFF(month, `2011T`, `2010T`)                     -- -12
DATE_DIFF(day, `2010-01-01T23:00`, `2010-01-02T01:00`) -- 0 (need to be at least 24h
apart to be 1 day apart)
```

EXTRACT

Dada uma parte da data e um carimbo de data/hora, EXTRACT retorna o valor da parte da data do carimbo de data/hora.

Sintaxe

```
EXTRACT( date_part FROM timestamp )
```

Parâmetros

date_part

Especifica que parte dos carimbos de data/hora deve ser extraída. Pode ser uma das partes a seguir:

- YEAR
- MONTH
- DAY
- HOUR
- MINUTE
- SECOND

- TIMEZONE_HOUR
- TIMEZONE_MINUTE

timestamp

O carimbo de data/hora de destino no qual a função opera.

Exemplos

```
EXTRACT(YEAR FROM `2010-01-01T`) -- 2010
EXTRACT(MONTH FROM `2010T`) -- 1 (equivalent to
2010-01-01T00:00:00.000Z)
EXTRACT(MONTH FROM `2010-10T`) -- 10
EXTRACT(HOUR FROM `2017-01-02T03:04:05+07:08`) -- 3
EXTRACT(MINUTE FROM `2017-01-02T03:04:05+07:08`) -- 4
EXTRACT(TIMEZONE_HOUR FROM `2017-01-02T03:04:05+07:08`) -- 7
EXTRACT(TIMEZONE_MINUTE FROM `2017-01-02T03:04:05+07:08`) -- 8
```

TO_STRING

Dado um carimbo de data/hora e um padrão de formato, TO_STRING retorna uma representação de string do carimbo de data/hora no formato especificado.

Sintaxe

```
TO_STRING ( timestamp time_format_pattern )
```

Parâmetros

timestamp

O carimbo de data/hora de destino no qual a função opera.

time_format_pattern

Uma string que possui as seguintes interpretações de caracteres especiais:

Formato	Exemplo	Descrição
yy	69	Ano com dois dígitos

Formato	Exemplo	Descrição
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: esse formato não é válido para uso com a função TO_TIMESTAMP .)
d	2	Dia do mês (1 a 31)

Formato	Exemplo	Descrição
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)
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)

Formato	Exemplo	Descrição
ss	05	Segundo do minuto preenchido com zeros (00 a 59)
S	0	Fração de um segundo (precisão: 0,1, intervalo: de 0,0 a 0,9)
SS	6	Fração de um segundo (precisão: 0,01, intervalo: de 0,0 a 0,99)
SSS	60	Fração de um segundo (precisão: 0,001, intervalo: de 0,0 a 0,999)
...
SSSSSSSSS	60000000	Fração de um segundo (precisão máxima: 1 nanossegundo, intervalo: de 0,0 a 0,999999999)

Formato	Exemplo	Descrição
n	60000000	Nano de um segundo
X	+07 ou Z	Deslocamento em horas ou Z se o deslocamento for 0
XX ou XXXX	+0700 ou Z	Deslocamento em horas e minutos ou Z se o deslocamento for 0
XXX ou XXXXX	+07:00 ou Z	Deslocamento em horas e minutos ou Z se o deslocamento for 0
x	7	Deslocamento em horas
xx ou xxxx	700	Deslocamento em horas e minutos
xxx ou 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`, 'MMMM d, y h:m a')  -- "July 20, 1969 8:18
PM"
TO_STRING(`1969-07-20T20:18Z`, 'y-MM-dd''T''H:m:ssX') --
"1969-07-20T20:18:00Z"
TO_STRING(`1969-07-20T20:18+08:00Z`, 'y-MM-dd''T''H:m:ssX') --
"1969-07-20T20:18:00Z"
TO_STRING(`1969-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXX') --
"1969-07-20T20:18:00+0800"
TO_STRING(`1969-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXXX') --
"1969-07-20T20:18:00+08:00"

```

TO_TIMESTAMP

Dada uma string, `TO_TIMESTAMP` converte-se em um carimbo de data/hora. `TO_TIMESTAMP` 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

`UTCNOW` retorna o tempo atual em UTC como um carimbo de data/hora.

Sintaxe

```
UTCNOW()
```

Parâmetros

UTCNOW não aceita parâmetros.

Exemplos

```
UTCNOW() -- 2017-10-13T16:02:11.123Z
```

Funções de string

O Amazon S3 Select é compatível com as seguintes funções de string.

Tópicos

- [CHAR_LENGTH, CHARACTER_LENGTH](#)
- [LOWER](#)
- [SUBSTRING](#)
- [TRIM](#)
- [UPPER](#)

CHAR_LENGTH, CHARACTER_LENGTH

CHAR_LENGTH (ou 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, LOWER 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, SUBSTRING 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 uma posição de índice de 1.

- Se `start < 1`, sem comprimento especificado, a posição do índice será definida como 1.
- Se `start for < 1`, sem comprimento especificado, a posição do índice será definida como `start + length - 1`.
- Se `start + length - 1 < 0`, uma string vazia será retornada.
- Se `start + length - 1 >= 0`, a substring começando na posição de 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 um espaço (' ').

Sintaxe

```
TRIM ( [[LEADING | TRAILING | BOTH remove_chars] FROM] string )
```

Parâmetros

string

A string de destino na qual a função opera.

LEADING | TRAILING | BOTH

Esse parâmetro indica se é necessário cortar os caracteres iniciais ou finais, ou ambos.

remove_chars

O conjunto de caracteres a ser removido. *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, UPPER 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!@#\$') -- 'ABCDEFGH!@#\$'
```

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.

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#). Para obter mais informações sobre o uso da funcionalidade Operações em Lote com a classe S3 Express One Zone e buckets de diretório, consulte [Usar a funcionalidade Operações em Lote com a classe S3 Express One Zone](#).

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](#).

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](#) 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 lotes 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](#).

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 AWS Elemental MediaConvert](#)

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](#)
- [Anexar políticas de permissões](#)

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](#) e [Copiar objetos usando o S3 Batch Operations](#).

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. Consulte mais informações e uma lista completa das chaves de condição específicas do Amazon S3 em [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço.

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.
- Se você enviar um manifesto de relatório de inventário criptografado com o AWS KMS, a política do IAM deverá incluir as permissões "kms:Decrypt" e "kms:GenerateDataKey" para o objeto manifest.json e todos os arquivos de dados CSV associados.

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](#).

```

{
  "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](#).

```

{
  "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 a funcionalidade Operações em Lote do Amazon S3, 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 `CreateJob`. Ela também fornece instruções para criar um trabalho de Operações em Lote usando o console do Amazon S3, a AWS Command Line Interface (AWS CLI) e o 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. Contudo que pelo menos

uma tarefa tenha sido invocada com êxito, a funcionalidade Operações em Lote do S3 gera um relatório para tarefas que foram concluídas, que falharam ou que foram canceladas. Para ter mais informações, consulte [Exemplos: relatórios de conclusão de operações em lote do S3](#).

Tópicos

- [Elementos da solicitação de trabalho de Operações em lote](#)
- [Especificar um manifesto](#)

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. A funcionalidade Operações em Lote possibilita a execução de uma operação em lote, com os mesmos resultados da operação executada individualmente em cada objeto.

Manifesto

O manifesto é uma lista de todos os objetos em que você deseja que a funcionalidade Operações em Lote do S3 execute a ação especificada. Você pode usar os seguintes métodos para especificar um manifesto para um trabalho de Operações em Lote:

- Crie manualmente sua própria lista de objetos personalizada em formato CSV.
- Escolha um relatório do [Inventário do Amazon S3](#) existente em formato CSV.
- Instruir a funcionalidade Operações em Lote para que gere um manifesto automaticamente com base nos critérios de filtro de objetos especificados ao criar um trabalho. Essa opção está disponível para trabalhos de replicação em lote criados no console do Amazon S3 ou para qualquer tipo de trabalho criado usando a AWS CLI, SDKs da AWS ou a API REST do Amazon S3.

Note

- Independentemente de como você especifica o manifesto, a lista em si deve ser armazenada em um bucket de uso geral. A funcionalidade Operações em Lote não pode importar manifestos existentes nem salvar manifestos gerados em buckets de

diretório. No entanto, os objetos descritos no manifesto podem ser armazenados em buckets de diretório. Para obter mais informações, consulte [Directory buckets](#).

- Se os objetos no manifesto estiverem em um bucket com versionamento, especificar os IDs de versão dos objetos instruirá a funcionalidade Operações em Lote a realizar a operação em uma versão específica. Se nenhum ID de versão for especificado, a funcionalidade Operações em Lote realizará a operação na versão mais recente dos objetos. Se o manifesto incluir um campo de ID de versão, você deverá fornecer um ID de versão para todos os objetos no manifesto.

Para ter mais informações, consulte [Especificar um manifesto](#).

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 a todos os trabalhos Restaurar (RestoreObject) uma prioridade 1, a todos os trabalhos de Copiar (CopyObject) uma prioridade 2 e a todos os trabalhos de Substituir listas de controle de acesso (ACLs) (PutObjectAcl) uma prioridade 3.


A funcionalidade Operações em Lote do S3 prioriza trabalhos em ordem numérica, mas o rigor dessa ordem não é garantido. Por isso, não use prioridades de trabalho para garantir que um trabalho comece ou termine antes de outro. Caso precise garantir uma ordem rigorosa, 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 CopyObject, o perfil do IAM deve ter a permissão `s3:GetObject` para o bucket de origem e a permissão `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 Manual do usuário do IAM.

Para obter mais informações sobre as permissões do Amazon S3, consulte [Ações de políticas para o Amazon S3](#).


 Note

Os trabalhos de Operações em Lote que realizam ações em buckets de diretório exigem permissões específicas. Para obter mais informações, consulte [AWS Identity and Access Management \(IAM\) for S3 Express One Zone](#).

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.

 Note

O relatório deve ser armazenado em um bucket de uso geral. A funcionalidade Operações em Lote não pode salvar relatórios em buckets de diretório. Para obter mais informações, consulte [Directory buckets](#).

- 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. Você pode usar etiquetas para identificar quem é responsável por um trabalho de Operações em Lote ou para controlar como os usuários interagem com os trabalhos 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. Por exemplo, você pode conceder a um usuário permissão para invocar a operação `CreateJob`, desde que o trabalho seja criado com a etiqueta `"Department=Finance"`.

Você pode criar trabalhos com tags anexadas a eles e pode adicionar tags a trabalhos depois de criá-los.

Para ter mais informações, consulte [the section called “Usar tags”](#).

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 Amazon S3 atue. Você pode fornecer um manifesto de uma das seguintes maneiras:

- Crie um arquivo de manifesto manualmente.
- Use um manifesto existente.
- Instrua a funcionalidade Operações em Lote para que gere um manifesto automaticamente com base nos critérios de filtro de objetos especificados ao criar um trabalho. Essa opção está disponível para trabalhos de replicação em lote criados no console do Amazon S3 ou para qualquer tipo de trabalho criado usando a AWS CLI, SDKs da AWS ou a API REST do Amazon S3.

Note

Independentemente de como você especifica o manifesto, a lista em si deve ser armazenada em um bucket de uso geral. A funcionalidade Operações em Lote não pode importar manifestos existentes nem salvar manifestos gerados em buckets de diretório. No entanto, os objetos descritos no manifesto podem ser armazenados em buckets de diretório. Para obter mais informações, consulte [Directory buckets](#).

Criar um arquivo de manifesto

Para criar um arquivo de manifesto manualmente, especifique a chave de objeto do manifesto, a ETag (etiqueta de entidade) e o ID da versão opcional em uma lista em formato CSV. O conteúdo do manifesto deve estar codificado em URL.

Por padrão, o Amazon S3 usa criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) automaticamente para criptografar um manifesto carregado para um bucket do Amazon S3. Os manifestos que usam a criptografia no servidor com chaves fornecidas pelo cliente (SSE-C) não são compatíveis. Manifestos que usam criptografia no lado do servidor com chaves do AWS Key Management Service (AWS KMS) (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.

Note

Se os objetos no manifesto estiverem em um bucket com versionamento, especificar os IDs de versão dos objetos instruirá a funcionalidade Operações em Lote a realizar a operação em uma versão específica. Se nenhum ID de versão for especificado, a funcionalidade Operações em Lote realizará a operação na versão mais recente dos objetos. Se o manifesto incluir um campo de ID de versão, você deverá fornecer um ID de versão para todos os objetos no manifesto.

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 em formato CSV que inclui IDs de versão.

```
Examplebucket,objectkey1,PZ9ibn9D51P6p298B7S9_ceqx1n5EJ0p
Examplebucket,objectkey2,YY_ouuAJByNW1LRBfFMfxMge7XQWxMBF
```



```
Examplebucket,objectkey3,jbo9_jhdPEyB4Rim0xWS0kU0EoNrU_oI
Examplebucket,photos/jpgs/objectkey4,6EqlikJJxLTsHsnbZbSRffn24_eh5Ny4
Examplebucket,photos/jpgs/newjersey/objectkey5,imHf3FAiRsvBW_EHB8G0u.NHunH01gVs
Examplebucket,object%20key%20with%20spaces,9HkPvDaZY5MVbMhn6TMn1YTb5ArQao3w
```

Especificar um arquivo de manifesto existente

É possível especificar um arquivo de manifesto na solicitação de criação de um trabalho usando um dos seguintes formatos:

- Relatório da funcionalidade Inventário Amazon S3: deve ser um relatório da funcionalidade Inventário Amazon S3 em formato 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](#). Se o relatório de inventário incluir IDs de versões, o recurso Operações em lote do S3 operará nas versões especificadas do objeto.

Note

- A funcionalidade Operações em Lote do S3 oferece suporte a relatórios de inventário em CSV criptografados com SSE-KMS.
- Se você enviar um manifesto de relatório de inventário criptografado com SSE-KMS, a política do IAM deverá incluir as permissões `"kms:Decrypt"` e `"kms:GenerateDataKey"` para o objeto `manifest.json` e todos os arquivos de dados CSV associados.

- 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

A funcionalidade Operações em Lote do S3 não oferece suporte a arquivos de manifesto em CSV criptografados com SSE-KMS.

⚠ Important

Ao usar um manifesto criado manualmente e um bucket com versionamento, recomendamos especificar os IDs de 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, os trabalhos podem levar muito tempo para serem executados, o que pode afetar a versão de um objeto em que o trabalho atua. Suponha que você substitua um objeto por uma nova versão enquanto um trabalho estiver em execução e não especifique o ID de versão desse objeto. Nesse caso, o Amazon S3 realizará a operação na versão mais recente do objeto, não na versão que existia quando você criou o trabalho. A única maneira de evitar esse comportamento é especificar o ID de versão do objeto listado no manifesto.

Gerar um manifesto automaticamente

Você pode instruir o Amazon S3 a gerar um manifesto automaticamente com base nos critérios de filtro de objetos especificados ao criar um trabalho. Essa opção está disponível para trabalhos de replicação em lote criados no console do Amazon S3 ou para qualquer tipo de trabalho criado usando a AWS CLI, SDKs da AWS ou a API REST do Amazon S3. Para obter mais informações sobre replicação em lote, consulte [Replicar objetos existentes com o S3 Batch Replication](#).

Para gerar um manifesto automaticamente, especifique os seguintes elementos como parte da solicitação de criação de trabalho:

- Informações sobre o bucket que contém seus objetos de origem, incluindo o proprietário do bucket e o nome do recurso da Amazon (ARN).
- Informações sobre a saída do manifesto, incluindo um sinalizador para criar um arquivo de manifesto, o proprietário do bucket de saída, o ARN, o prefixo, o formato do arquivo e o tipo de criptografia.
- Critérios opcionais para filtrar objetos por data de criação, nome da chave, tamanho, classe de armazenamento e etiquetas.

Critérios de filtro de objetos

Para filtrar a lista de objetos a serem incluídos em um manifesto gerado automaticamente, você pode especificar os critérios a seguir. Para obter mais informações, consulte [JobManifestGeneratorFilter](#) na Referência da API do Amazon S3.

CreatedAfter

Se fornecido, o manifesto gerado incluirá somente objetos de bucket de origem que foram criados após esse período.

CreatedBefore

Se fornecido, o manifesto gerado incluirá somente objetos de bucket de origem que foram criados antes desse período.

EligibleForReplication

Se fornecido, o manifesto gerado só incluirá os objetos que estiverem qualificados para replicação de acordo com a configuração de replicação no bucket de origem.

KeyNameConstraint

Se fornecido, o manifesto gerado incluirá somente os objetos do bucket de origem cujas chaves de objeto correspondam às restrições de string especificadas para `MatchAnySubstring`, `MatchAnyPrefix` e `MatchAnySuffix`.

`MatchAnySubstring`: se fornecido, o manifesto gerado incluirá objetos se a string especificada aparecer em qualquer lugar dentro da string da chave do objeto.

`MatchAnyPrefix`: se fornecido, o manifesto gerado incluirá objetos se a string especificada aparecer no início da string da chave do objeto.

`MatchAnySuffix`: se fornecido, o manifesto gerado incluirá objetos se a string especificada aparecer no final da string da chave do objeto.

MatchAnyStorageClass

Se fornecido, o manifesto gerado incluirá somente os objetos do bucket de origem que estiverem armazenados com a classe de armazenamento especificada.

ObjectReplicationStatuses

Se fornecido, o manifesto gerado só incluirá os objetos do bucket de origem que tenham um dos status de replicação especificados.

ObjectSizeGreaterThanBytes

Se fornecido, o manifesto gerado só incluirá os objetos do bucket de origem cujo tamanho de arquivo for maior que o número especificado de bytes.

ObjectSizeLessThanBytes

Se fornecido, o manifesto gerado só incluirá os objetos do bucket de origem cujo tamanho de arquivo for menor que o número especificado de bytes.

Note

Não é possível clonar a maioria dos trabalhos que têm manifestos gerados automaticamente. Os trabalhos de replicação em lote podem ser clonados, exceto quando usam os critérios de filtro de manifesto `KeyNameConstraint`, `MatchAnyStorageClass`, `ObjectSizeGreaterThanBytes` ou `ObjectSizeLessThanBytes`.

A sintaxe para especificar os critérios do manifesto varia de acordo com o método usado para criar o trabalho. Para ver exemplos, consulte [Criar uma tarefa](#).

Criar uma tarefa

Você pode criar trabalhos de Operações em Lote do S3 usando o console do Amazon S3, a AWS CLI, AWS SDKs ou a API REST do Amazon S3.

Para obter mais informações sobre como criar uma solicitação de trabalho, consulte [Elementos da solicitação de trabalho de Operações em lote](#).

Pré-requisitos

Antes de criar um trabalho de Operações em Lote, confirme se você configurou as permissões relevantes. Para ter mais informações, consulte [Conceder permissões para operações em lote do Amazon S3](#).

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. Na barra de navegação na parte superior da página, escolha o nome da Região da AWS exibida no momento. Em seguida, escolha a região onde você deseja criar o trabalho.

 Note

Para operações de cópia, você deve criar o trabalho na mesma região que o bucket de destino. Para todas as outras operações, você deve criar o trabalho na mesma região que os objetos arquivados no manifesto.

3. Selecione Operações em lotes no painel de navegação do console do Amazon S3.
4. Escolha Criar trabalho.
5. Visualize a Região da AWS onde você deseja criar o trabalho.
6. 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.

 Note

O console do Amazon S3 oferece suporte à geração automática de manifestos somente para trabalhos de replicação em lote. Para todos os outros tipos de trabalho, se você quiser que o Amazon S3 gere um manifesto automaticamente com base nos critérios de filtro especificados, deverá configurar o trabalho usando a AWS CLI, AWS SDKs ou a API REST do Amazon S3.

7. Escolha Próximo.
8. 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).
9. Preencha as informações para Configure additional options (Configurar opções adicionais) e, depois, selecione Next (Próximo).

10. Em Review (Revisão), verifique as configurações. Se precisar fazer alterações, escolha Previous (Anterior). Caso contrário, selecione Create job (Criar trabalho).

Usando a AWS CLI

Specify manifest

O exemplo a seguir mostra como criar um trabalho `S3PutObjectTagging` de Operações em Lote do S3 que atua em objetos listados em um arquivo de manifesto existente.

Como criar uma tarefa **`S3PutObjectTagging`** de operações em lote


1. Use os comandos a seguir para criar um perfil do AWS Identity and Access Management (IAM), depois crie uma política do IAM para atribuir as permissões relevantes. O perfil e a política a seguir concedem permissão ao Amazon S3 para adicionar etiquetas de objeto, que serão necessárias quando você criar o trabalho em uma etapa posterior.
 - a. Use o exemplo de comando a seguir para criar um perfil do IAM para usar a funcionalidade Operações em Lote. Para usar esse exemplo de comando, substitua **`S3BatchJobRole`** pelo nome que você deseja dar ao perfil.

```
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. Use o exemplo de comando a seguir para criar uma política do IAM com as permissões necessárias e anexe-a ao perfil do IAM que você criou na etapa anterior. Para obter

mais informações sobre as permissões necessárias, consulte [Conceder permissões para operações em lote do Amazon S3](#).

 Note

Os trabalhos de Operações em Lote que realizam ações em buckets de diretório exigem permissões específicas. Para obter mais informações, consulte [AWS Identity and Access Management \(IAM\) for S3 Express One Zone](#).

Para usar esse exemplo de comando, substitua *user input placeholders* da seguinte forma:

- Substitua *S3BatchJobRole* pelo nome do seu perfil do IAM. Garanta que esse nome corresponda ao nome que você usou anteriormente.
- Substitua *PutObjectTaggingBatchJobPolicy* pelo nome que você deseja atribuir à política do IAM.
- Substitua *DOC-EXAMPLE-DESTINATION-BUCKET* pelo nome do bucket que contém os objetos aos quais você deseja aplicar etiquetas.
- Substitua *DOC-EXAMPLE-MANIFEST-BUCKET* pelo nome do bucket que contém o manifesto.
- Substitua *DOC-EXAMPLE-REPORT-BUCKET* pelo nome do bucket ao qual você deseja que o relatório de conclusão seja entregue.

```
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:::DOC-EXAMPLE-DESTINATION-BUCKET/*"
```

```

    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-BUCKET/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET/*"
      ]
    }
  ]
}'

```

2. Use o exemplo de comando a seguir para criar um trabalho S3PutObjectTagging.

O arquivo `manifest.csv` fornece uma lista de buckets e valores de chave de objeto. O trabalho aplica as etiquetas específicas aos objetos identificados no manifesto. O ETag é o ETag do objeto `manifest.csv`, que você pode obter no console do Amazon S3. Essa solicitação especifica o parâmetro `no-confirmation-required` para que você possa executar o trabalho sem precisar confirmá-lo com o comando `update-job-status`. Para obter mais informações, consulte [create-job](#) na Referência de comandos da AWS CLI.

Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações. Substitua *IAM-role* pelo ARN do perfil do IAM que você criou anteriormente.

```

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:::DOC-EXAMPLE-REPORT-
BUCKET", "Prefix": "final-reports",
"Format": "Report_CSV_20180820", "Enabled": true, "ReportScope": "AllTasks"}' \
--priority 42 \
--role-arn IAM-role \
--client-request-token $(uuidgen) \
--description "job description" \
--no-confirmation-required

```

Em resposta, o Amazon S3 retorna um ID de trabalho (por exemplo, `00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c`). Você precisará do ID do trabalho para identificar, monitorar e modificar o trabalho.

Generate manifest

O exemplo a seguir mostra como criar um trabalho `S3DeleteObjectTagging` de Operações em Lote do S3 que gera automaticamente um manifesto com base em critérios de filtro de objetos. Esses critérios incluem a data de criação, o nome da chave, o tamanho, a classe de armazenamento e etiquetas.

Como criar uma tarefa `S3DeleteObjectTagging` de operações em lote

1. Use os comandos a seguir para criar um perfil do AWS Identity and Access Management (IAM), depois crie uma política do IAM para atribuir as permissões. O perfil e a política a seguir concedem permissão ao Amazon S3 para excluir etiquetas de objeto, que serão necessárias quando você criar o trabalho em uma etapa posterior.
 - a. Use o exemplo de comando a seguir para criar um perfil do IAM para usar a funcionalidade Operações em Lote. Para usar esse exemplo de comando, substitua `S3BatchJobRole` pelo nome que você deseja dar ao perfil.

```


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. Use o exemplo de comando a seguir para criar uma política do IAM com as permissões necessárias e anexe-a ao perfil do IAM que você criou na etapa anterior. Para obter mais informações sobre as permissões necessárias, consulte [Conceder permissões para operações em lote do Amazon S3](#).

 Note

Os trabalhos de Operações em Lote que realizam ações em buckets de diretório exigem permissões específicas. Para obter mais informações, consulte [AWS Identity and Access Management \(IAM\) for S3 Express One Zone](#).

Para usar esse exemplo de comando, substitua *user input placeholders* da seguinte forma:

- Substitua *S3BatchJobRole* pelo nome do seu perfil do IAM. Garanta que esse nome corresponda ao nome que você usou anteriormente.
- Substitua *DeleteObjectTaggingBatchJobPolicy* pelo nome que você deseja atribuir à política do IAM.
- Substitua *DOC-EXAMPLE-DESTINATION-BUCKET* pelo nome do bucket que contém os objetos aos quais você deseja aplicar etiquetas.
- Substitua *DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET* pelo nome do bucket onde deseja salvar o manifesto.

- Substitua *DOC-EXAMPLE-REPORT-BUCKET* pelo nome do bucket ao qual você deseja que o relatório de conclusão seja entregue.

```
aws iam put-role-policy \  
  --role-name S3BatchJobRole \  
  --policy-name DeleteObjectTaggingBatchJobPolicy \  
  --policy-document '{  
    "Version":"2012-10-17",  
    "Statement":[  
      {  
        "Effect":"Allow",  
        "Action":[  
          "s3:DeleteObjectTagging",  
          "s3:DeleteObjectVersionTagging"  
        ],  
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET/*"  
      },  
      {  
        "Effect":"Allow",  
        "Action":[  
          "s3:PutInventoryConfiguration"  
        ],  
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET"  
      },  
      {  
        "Effect": "Allow",  
        "Action": [  
          "s3:GetObject",  
          "s3:GetObjectVersion",  
          "s3:ListBucket"  
        ],  
        "Resource": [  
          "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET",  
          "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET/*"  
        ]  
      },  
      {  
        "Effect":"Allow",  
        "Action":[  
          "s3:PutObject",  
          "s3:ListBucket"  
        ],  
      },  
    ]  
  }'
```

```

    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET",
      "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET/*",
      "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET/*"
    ]
  }
]
}'

```

- Use o exemplo de comando a seguir para criar o trabalho `S3DeleteObjectTagging`.

Neste exemplo, os valores na seção `--report` especificam o bucket, o prefixo, o formato e o escopo do relatório de trabalho que será gerado. A seção `--manifest-generator` especifica informações sobre o bucket de origem que contém os objetos sobre os quais o trabalho atuará, informações sobre a lista de saída do manifesto que será gerada para o trabalho e critérios de filtro para restringir o escopo de objetos a serem incluídos no manifesto por data de criação, restrições de nome, tamanho e classe de armazenamento. O comando também especifica a prioridade do trabalho, o perfil do IAM e a Região da AWS.

Para obter mais informações, consulte [create-job](#) na Referência de comandos da AWS CLI.

Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações. Substitua *IAM-role* pelo ARN do perfil do IAM que você criou anteriormente.

```

aws s3control create-job \
  --account-id 012345678901 \
  --operation '{
    "S3DeleteObjectTagging": {}
  }' \
  --report '{
    "Bucket":"arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET",
    "Prefix":"reports",
    "Format":"Report_CSV_20180820",
    "Enabled":true,
    "ReportScope":"AllTasks"
  }' \
  --manifest-generator '{
    "S3JobManifestGenerator": {
      "ExpectedBucketOwner": "012345678901",
      "SourceBucket": "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET",

```

```

"EnableManifestOutput": true,
"ManifestOutputLocation": {
  "ExpectedManifestBucketOwner": "012345678901",
  "Bucket": "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET",
  "ManifestPrefix": "prefix",
  "ManifestFormat": "S3InventoryReport_CSV_20211130"
},
"Filter": {
  "CreatedAfter": "2023-09-01",
  "CreatedBefore": "2023-10-01",
  "KeyNameConstraint": {
    "MatchAnyPrefix": [
      "prefix"
    ],
    "MatchAnySuffix": [
      "suffix"
    ]
  },
  "ObjectSizeGreaterThanOrEqualToBytes": 100,
  "ObjectSizeLessThanBytes": 200,
  "MatchAnyStorageClass": [
    "STANDARD",
    "STANDARD_IA"
  ]
}
}' \
--priority 2 \
--role-arn IAM-role \
--region us-east-1

```

Em resposta, o Amazon S3 retorna um ID de trabalho (por exemplo, `00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c`). Você precisará desse ID de trabalho para identificar, monitorar e modificar o trabalho.

Usando a AWS SDK for Java

Specify manifest

O exemplo a seguir mostra como criar um trabalho `S3PutObjectTagging` de Operações em Lote do S3 que atua em objetos listados em um arquivo de manifesto existente. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

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:::DOC-EXAMPLE-REPORT-BUCKET";
        String uuid = UUID.randomUUID().toString();

        ArrayList tagSet = new ArrayList<S3Tag>();
        tagSet.add(new S3Tag().withKey("keyOne").withValue("ValueOne"));

        try {
            JobOperation jobOperation = new JobOperation()
                .withS3PutObjectTagging(new S3SetObjectTaggingOperation()
                    .withTagSet(tagSet)
                );

            JobManifest manifest = new JobManifest()
                .withSpec(new JobManifestSpec()
                    .withFormat("S3BatchOperations_CSV_20180820")
                    .withFields(new String[][]{
                        {"Bucket", "Key"}
                    })
                )
                .withLocation(new JobManifestLocation()
                    .withObjectArn("arn:aws:s3:::my_manifests/manifest.csv")
                );
        }
    }
}
```

```

        .withETag("60e460c9d1046e73f7dde5043ac3ae85"));
    JobReport jobReport = new JobReport()
        .withBucket(reportBucketName)
        .withPrefix("reports")
        .withFormat("Report_CSV_20180820")
        .withEnabled(true)
        .withReportScope("AllTasks");

    AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
        .withCredentials(new ProfileCredentialsProvider())
        .withRegion(US_WEST_2)
        .build();

    s3ControlClient.createJob(new CreateJobRequest()
        .withAccountId(accountId)
        .withOperation(jobOperation)
        .withManifest(manifest)
        .withReport(jobReport)
        .withPriority(42)
        .withRoleArn(iamRoleArn)
        .withClientRequestToken(uuid)
        .withDescription("job description")
        .withConfirmationRequired(false)
    );

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

```

Generate manifest

O exemplo a seguir mostra como criar um trabalho `s3PutObjectCopy` de Operações em Lote do S3 que gera automaticamente um manifesto com base em critérios de filtro de objetos, incluindo a data de criação, o nome da chave e o tamanho. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

Example

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.CreateJobRequest;
import com.amazonaws.services.s3control.model.CreateJobResult;
import com.amazonaws.services.s3control.model.JobManifestGenerator;
import com.amazonaws.services.s3control.model.JobManifestGeneratorFilter;
import com.amazonaws.services.s3control.model.JobOperation;
import com.amazonaws.services.s3control.model.JobReport;
import com.amazonaws.services.s3control.model.KeyNameConstraint;
import com.amazonaws.services.s3control.model.S3JobManifestGenerator;
import com.amazonaws.services.s3control.model.S3ManifestOutputLocation;
import com.amazonaws.services.s3control.model.S3SetObjectTaggingOperation;
import com.amazonaws.services.s3control.model.S3Tag;

import java.time.Instant;
import java.util.Date;
import java.util.UUID;
import java.util.ArrayList;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class test {
    public static void main(String[] args) {
        String accountId = "012345678901";
        String iamRoleArn = "arn:aws:iam::012345678901:role/ROLE";
        String sourceBucketName = "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET";
        String reportBucketName = "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET";
        String manifestOutputBucketName = "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-
OUTPUT-BUCKET";
        String uuid = UUID.randomUUID().toString();
        long minimumObjectSize = 100L;

        ArrayList<S3Tag> tagSet = new ArrayList<>();
        tagSet.add(new S3Tag().withKey("keyOne").withValue("ValueOne"));

        ArrayList<String> prefixes = new ArrayList<>();
```



```
    prefixes.add("s3KeyStartsWith");

    try {
        JobOperation jobOperation = new JobOperation()
            .withS3PutObjectTagging(new S3SetObjectTaggingOperation()
                .withTagSet(tagSet)
            );
        S3ManifestOutputLocation manifestOutputLocation = new
S3ManifestOutputLocation()
            .withBucket(manifestOutputBucketName)
            .withManifestPrefix("manifests")
            .withExpectedManifestBucketOwner(accountId)
            .withManifestFormat("S3InventoryReport_CSV_20211130");

        JobManifestGeneratorFilter jobManifestGeneratorFilter = new
JobManifestGeneratorFilter()
            .withEligibleForReplication(true)
            .withKeyNameConstraint(
                new KeyNameConstraint()
                    .withMatchAnyPrefix(prefixes))
            .withCreatedBefore(Date.from(Instant.now()))
            .withObjectSizeGreaterThanBytes(minimumObjectSize);

        S3JobManifestGenerator s3JobManifestGenerator = new
S3JobManifestGenerator()
            .withEnableManifestOutput(true)
            .withManifestOutputLocation(manifestOutputLocation)
            .withFilter(jobManifestGeneratorFilter)
            .withSourceBucket(sourceBucketName);

        JobManifestGenerator jobManifestGenerator = new
JobManifestGenerator()
            .withS3JobManifestGenerator(s3JobManifestGenerator);

        JobReport jobReport = new JobReport()
            .withBucket(reportBucketName)
            .withPrefix("reports")
            .withFormat("Report_CSV_20180820")
            .withEnabled(true)
            .withReportScope("AllTasks");

        AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
            .withCredentials(new ProfileCredentialsProvider())
            .withRegion(US_WEST_2)
```

```
        .build();

        CreateJobResult createJobResult = s3ControlClient.createJob(new
CreateJobRequest()

        .withAccountId(accountId)
        .withOperation(jobOperation)
        .withManifestGenerator(jobManifestGenerator)
        .withReport(jobReport)
        .withPriority(42)
        .withRoleArn(iamRoleArn)
        .withClientRequestToken(uuid)
        .withDescription("job description")
        .withConfirmationRequired(true)

        );

        System.out.println("Created job " + createJobResult.getJobId());

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

Uso da API REST

Você pode usar a API REST para criar um trabalho do Batch Operations. Para obter mais informações, consulte [CreateJob](#) na Referência da API do Amazon Simple Storage Service.

Respostas do trabalho

Se a solicitação CreateJob 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 pela AWS CLI, por AWS SDKs ou pela API REST, é possível definir a funcionalidade Operações em Lote do S3 para que comece a processar o trabalho automaticamente. O trabalho é executado assim que fica pronto em vez de aguardar o processamento de trabalhos de prioridade mais alta.

Ao criar um trabalho por meio do console do Amazon S3, você deve examinar os detalhes e confirmar se deseja executá-lo antes que a funcionalidade Operações em Lote comece a processá-lo. 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](#).

Para obter mais informações sobre como copiar objetos no Amazon S3, bem como parâmetros obrigatórios e opcionais, consulte [Copiar e mover objetos](#) 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](#).
- 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](#)
- [Uso de um manifesto CSV armazenado na conta de origem para copiar objetos entre Contas da AWS](#)
- [Uso do S3 Batch Operations para criptografar objetos com chaves de bucket do S3](#)

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”](#).

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](#).

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 ou sobre como criptografar um arquivo de lista de inventário, consulte [Configurar o inventário do Amazon S3](#).

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](#).

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 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:::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](#).

Para obter informações sobre a criação de um trabalho usando o console, consulte [Criar um trabalho de operações em lote do S3](#).

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"](#).

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](#).

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:::user/ConsoleUserCreatingJob",
          "arn:aws:iam:::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:GetObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3::ObjectSourceBucket/*"
    }
  ]
}
```

4. 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](#).

Para obter informações sobre a criação de um trabalho usando o console, consulte [Criar um trabalho de operações em lote do S3](#).

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](#) e [Configurando seu bucket para usar uma chave de bucket do S3 com SSE-KMS para novos objetos](#).

Os tópicos abordados neste exemplo incluem:

Tópicos

- [Pré-requisitos](#)
- [Etapa 1: obter a lista de objetos usando o Amazon S3 Inventory](#)
- [Etapa 2: Filtrar sua lista de objetos com S3 Select](#)
- [Etapa 3: Configurar e executar o trabalho do S3 Batch Operations](#)
- [Resumo](#)

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](#)
- [Criar um trabalho de operações em lote do S3](#)
- [Operações suportadas pelo S3 Batch Operations](#)
- [Gerenciar trabalhos de operações em lote do S3](#)

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](#).

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 ter mais informações, consulte [Conceder permissões para o Inventário do S3 e análises do S3](#).

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. (Opcional) Na seção Campos adicionais: opcional, selecione Criptografia e quaisquer outros campos de relatórios de seu interesse. 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. Depois de 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 do S3. Se quiser que todos os objetos do bucket sejam criptografados com chaves de bucket do S3, 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](#) 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.

As Operações em Lote do S3 precisam do bucket, da chave e do ID da versão como entradas para executar o trabalho, além do campo pelo qual pesquisar, que é o `BucketKeyStatus`. Você não precisa do campo de ID da versão, mas é útil especificá-lo quando você opera em um bucket versionado. Para ter mais informações, consulte [Trabalhar com objetos em um bucket com versionamento habilitado](#).

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 a 3 para todos os objetos sem chave de bucket do S3 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,TLYESLn1mXD5c4Bwi0IinqFrktddkoL
batchoperationsdemo,0200147%7Ethumb.jpg,amufzfMi_fEw0Rs99rxR_HrDF1E.l3Y0
batchoperationsdemo,0301420%7Ethumb.jpg,9qGU2SEscL.C.c_sK89trmXYIwooABSh
batchoperationsdemo,0401524%7Ethumb.jpg,ORnEWNuB1QhHrrYAGFsZhbyvEYJ3DUor
batchoperationsdemo,200907200065HQ
%7Ethumb.jpg,d8LgvIVjbDR5mUVwW6pu9ahTfReyn5V4
batchoperationsdemo,200907200076HQ
%7Ethumb.jpg,XUT25d7.gK40u_GmnupdaZg3BVx2jN40
batchoperationsdemo,201103190002HQ
%7Ethumb.jpg,z.2sVRh0myqVi0BuIrnqWlsRPQdb7q0S
```

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](#)
- [Configurar sua função do IAM do Batch Operations](#)
- [Ative as chaves de bucket do S3 para um bucket existente](#)
- [Criar seu trabalho do Batch Operations](#)
- [Executar seu trabalho do Batch Operations](#)
- [Pontos importantes](#)

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. Selecione Edit 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 de seu bucket de origem.
- b. Substitua *DESTINATION_BUCKET_FOR_COPY* pelo nome de seu 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 Perfis e escolha Criar perfil.
2. Selecione AWS service (Serviço da AWS), S3 e 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 ao perfil do IAM que você criará na etapa de criação do trabalho das Operações em Lote posteriormente. Para ter mais informações, consulte [Conceder permissões para operações em lote do Amazon S3](#).

```
{
  "Sid": "AddIamPermissions",
  "Effect": "Allow",
  "Action": [
    "iam:GetRole",
    "iam:PassRole"
  ],
  "Resource": "arn:aws:iam:::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 uma chave de bucket do S3.
3. Escolha Properties (Propriedades).
4. Em Default encryption (Criptografia padrão), escolha Edit (Editar).
5. Em Tipo de criptografia, você pode escolher entre Chaves gerenciadas do Amazon S3 (SSE-S3) e Chave do AWS Key Management Service (SSE-KMS).
6. Se você escolheu chave do AWS Key Management Service (SSE-KMS), em AWS KMS key, poderá especificar a chave do AWS KMS por meio de uma das opções a seguir.
 - Para selecionar entre uma lista de chaves do KMS disponíveis, selecione Escolher entre suas chaves do AWS KMS. Na lista de chaves disponíveis, selecione uma chave do KMS de criptografia simétrica na mesma região que seu bucket. A chave gerenciada pela AWS (aws/s3) e as chaves gerenciadas pelo cliente são exibidas na lista.
 - Para inserir o ARN da chave do KMS, selecione Inserir ARN da chave do AWS KMS e insira o ARN da chave do KMS no campo exibido.
 - Para criar uma chave gerenciada pelo cliente no console do AWS KMS, selecione Criar uma chave do KMS.
7. Em Bucket key (Chave do bucket), escolha Enable (Habilitar) e, em seguida, escolha Save changes (Salvar alterações).


Agora que a chave de bucket do S3 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 Operações em Lote do Amazon S3.

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 o recurso

de chaves de bucket do S3 habilitado, a operação de cópia aplicará chaves de bucket do S3 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 Próximo.
7. Para a criptografia no lado do servidor, siga estas etapas:
 - a. Em Criptografia do lado do servidor, selecione uma das seguintes opções:
 - Para manter as configurações de bucket para a criptografia padrão de objetos no lado do servidor ao armazená-los no Amazon S3, selecione Não especificar uma chave de criptografia. Desde que o destino do bucket tenha o recurso de chaves de bucket do S3 ativado, a operação de cópia aplicará uma chave de bucket do S3 ao bucket de destino.

 Note

Se a política de bucket para o destino especificado exigir que os objetos sejam criptografados antes de armazená-los no Amazon S3, você deverá especificar uma chave de criptografia. Caso contrário, ocorrerá uma falha na cópia de objetos no destino.

- Para criptografar objetos antes de armazená-los no Amazon S3, selecione Especificar uma chave de criptografia.
- b. Em Configurações de criptografia, se você selecionar Especificar uma chave de criptografia, deverá escolher Usar configurações do bucket de destino para criptografia padrão ou Substituir configurações do bucket de destino para criptografia padrão.
 - c. Se você selecionar Substituir configurações do bucket de destino para criptografia padrão, deverá definir as configurações de criptografia a seguir.
 - i. Em Tipo de criptografia, você deve escolher Chaves gerenciadas do Amazon S3 (SSE-S3) ou Chave do AWS Key Management Service (SSE-KMS). A SSE-S3 usa uma das criptografias de bloco mais fortes: o padrão de criptografia avançada de 256 bits (AES-256) para criptografar cada objeto. A SSE-KMS oferece mais controle sobre sua chave. Para obter mais informações, consulte [Usar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#) e [Usar criptografia do lado do servidor com o AWS KMS \(SSE-KMS\)](#).

- ii. Se você selecionar chave do AWS Key Management Service (SSE-KMS), em AWS KMS key, poderá especificar sua AWS KMS key por meio de uma das opções a seguir.
 - Para escolher em uma lista de chaves do KMS disponíveis, selecione Escolher entre suas AWS KMS keys e, depois, escolha uma chave do KMS de criptografia simétrica na mesma região que seu bucket. A chave gerenciada pela AWS (aws/s3) e as chaves gerenciadas pelo cliente são exibidas na lista.
 - Para inserir o ARN da chave do KMS, selecione Inserir ARN da chave do AWS KMS e insira o ARN da chave do KMS no campo exibido.
 - Para criar uma chave gerenciada pelo cliente no console do AWS KMS, selecione Criar uma chave do KMS.
 - iii. Em Bucket key (Chave do bucket), escolha Enable (Ativar). A operação de cópia aplicará uma chave do bucket do S3 no bucket de destino.
8. 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).
 9. Na seção Permissions (Permissões), certifique-se de escolher a função do IAM do Batch Operations definida anteriormente. Escolha Next (Próximo).
 10. 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](#).

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, AWS SDKs ou a API REST do Amazon S3. 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](#).

Pontos importantes

Considere os seguintes problemas ao usar as Operações em Lote do S3 para criptografar objetos com chaves de bucket do S3:

- Você será cobrado por trabalhos, objetos e solicitações das Operações em Lote do S3, além de quaisquer encargos associados à operação que as Operações em Lote do S3 executa em seu nome, como 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 do bucket do S3 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](#).
- 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 em seu 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. Depois, você aplicou o recurso de chave de bucket em objetos não criptografados usando as Operações em

Lote do S3 para copiar dados existentes em um bucket com chave de bucket do S3 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](#).

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](#).

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](#)
- [Criar uma função do Lambda para uso com operações em lote do S3](#)
- [Criar um trabalho de operações em lote do S3 que chama uma função do Lambda](#)
- [Fornecer informações em nível de tarefa em manifestos do Lambda](#)
- [Tutorial de uso do recurso Operações em Lote do S3](#)

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.

Usar o Lambda e a funcionalidade Operações em Lote do Amazon S3 com buckets de diretório

Os buckets de diretório são um tipo de bucket do Amazon S3 projetado para workloads ou aplicações críticas à performance que requerem latência consistente inferior a dez milissegundos. Para obter mais informações, consulte [Directory buckets](#).

Há requisitos especiais para usar a funcionalidade Operações em Lote do Amazon S3 para invocar funções do Lambda que atuam em buckets de diretório. Por exemplo, você deve estruturar a solicitação do Lambda usando um esquema JSON atualizado e especificar [InvocationSchemaVersion 2.0](#) ao criar o trabalho. Esse esquema atualizado permite que você especifique pares opcionais de chave e valor para [UserArguments](#), que você pode usar para modificar determinados parâmetros de funções do Lambda existentes. Para obter mais informações, consulte [Automate object processing in Amazon S3 directory buckets with S3 Batch Operations and AWS Lambda](#) no blog de armazenamento da AWS.

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
Succeeded	A tarefa foi concluída normalmente. Se você solicitou um relatório de conclusão do trabalho, a string do resultado da tarefa será incluída no relatório.
TemporaryFailure	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.
PermanentFailure	A tarefa sofreu uma falha permanente. Se você solicitou um relatório de conclusão do trabalho, a tarefa será marcada como Failed 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.

Exemplo 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"
    }
  ]
}
```

Exemplo 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.

Exemplo Solicitação

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": "dGFza2lkZ29lc2hlcmUK",
      "s3Key": "customerImage1.jpg",
      "s3VersionId": "1",
      "s3BucketArn": "arn:aws:s3:us-east-1:0123456788:awsexamplebucket1"
    }
  ]
}

```

Exemplo Resposta

Veja a seguir um exemplo do JSON de uma resposta para a função do Lambda.

```
{
```

```
"invocationSchemaVersion": "1.0",
"treatMissingKeysAs" : "PermanentFailure",
"invocationId" : "YXNkbGZqYWRmaiBhc2RmdW9hZHNmZGpmaGFzbGtkaGZza2RmaAo",
"results": [
  {
    "taskId": "dGFza2lkZ29lc2hlcmUK",
    "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](#) e [Operações suportadas pelo S3 Batch Operations](#).

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,"Pre
--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,
}
```

Tutorial de uso do recurso Operações em Lote do S3

O tutorial a seguir apresenta procedimentos completos para algumas tarefas do recurso Operações em Lote com o Lambda.

- [Tutorial: Vídeos de transcodificação em lote com operações em lote do S3, AWS Lambda e AWS Elemental MediaConvert](#)

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](#) 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](#).

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](#).

Para obter mais detalhes sobre a marcação de objetos, consulte [Substituir todas as tags de objeto](#) 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 \(ACL\)](#).

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](#).

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

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 com operações em lote

A operação Restaurar inicia solicitações de restauração dos objetos arquivados do Amazon S3 listados no manifesto. Os objetos a seguir devem ser restaurados para 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 do S3 para receber notificações quando os objetos estiverem disponíveis no Amazon S3. Para obter mais informações, consulte [Notificações de eventos do Amazon S3](#).

Para criar um trabalho para 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 para iniciar a restauração voltados para objetos do S3 Glacier Flexible Retrieval e do S3 Glacier Deep Archive exigem que você defina `ExpirationInDays` como 1 ou superior.

Important

Não defina `ExpirationInDays` ao criar trabalhos para iniciar a restauração de um objeto do S3 voltados a objetos dos níveis S3 Intelligent-Tiering Archive Access e Deep Archive Access. Os objetos no nível S3 Intelligent-Tiering Archive Access não estão sujeitos a uma data de validade da restauração; portanto, especificar `ExpirationInDays` provoca uma falha na solicitação de restauração.

GlacierJobTier

O Amazon S3 pode restaurar objetos usando um destes três diferentes níveis de recuperação: EXPEDITED, STANDARD e BULK. No entanto, o recurso Operações em lote do S3 suporta apenas os níveis de recuperação STANDARD e BULK. Para receber mais informações sobre as diferenças entre os níveis de recuperação, consulte [Opções de recuperação de arquivamento](#).

Para receber mais informações sobre preço para cada nível, consulte a seção Solicitações e recuperações de dados na página [Definição de preço do Amazon S3](#).

Diferenças na restauração por meio do S3 Glacier e do S3 Intelligent-Tiering

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 o término do valor especificado no argumento `ExpirationInDays`. Depois que essa cópia temporária for excluída, será preciso enviar uma solicitação de restauração adicional para acessar o objeto.
- Ao restaurar objetos arquivados do S3 Intelligent-Tiering, não especifique o argumento `ExpirationInDays`. Quando você restaura um objeto do nível S3 Intelligent-Tiering Archive Access ou Deep Archive Access, o objeto faz a transição de volta ao nível S3 Intelligent-Tiering Frequent Access. Após no mínimo 90 dias consecutivos sem acesso, o objeto transita automaticamente para o nível Archive Access. Depois de um período mínimo de 180 dias consecutivos sem acesso, o objeto é movido automaticamente ao nível Deep Archive Access.
- Os trabalhos de operações em lote podem operar em objetos das classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive ou em objetos dos níveis S3 Intelligent-Tiering Archive Access e Deep Archive Access. As operações em lote não podem operar nos dois tipos de objetos arquivados no mesmo trabalho. Para restaurar objetos de ambos os tipos, você deve criar trabalhos de Operações em lote separados.

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 à solicitação de restauração já em andamento, o valor `ExpirationInDays` desse trabalho é o mesmo, e o valor `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 validade do objeto restaurado para coincidir com o valor `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 esse trabalho (especificada por `ExpirationInDays`) é diferente da duração da restauração especificada na solicitação de restauração em andamento.
- O nível de restauração para esse trabalho (especificado pelo valor `GlacierJobTier`) é o mesmo ou é mais lento do que o nível especificado na solicitação de restauração em andamento.

Limitações

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.

Para obter mais informações sobre a restauração de objetos, consulte [Restaurar um objeto arquivado](#).

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”](#).

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 “Considerações sobre a funcionalidade Bloqueio de Objetos”](#).

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”](#). Para ver um exemplo de AWS SDK for Java, consulte [the section called “Usar operações em lote com retenção de bloqueio de objetos”](#).

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 “Considerações sobre a funcionalidade Bloqueio de Objetos”](#).

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](#).

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](#)
- [Invocar função do AWS Lambda](#)
- [Substituir todas as tags de objeto](#)

- [Excluir todas as tags de objeto](#)
- [Substituir lista de controle de acesso](#)
- [Restaurar objetos com operações em lote](#)
- [Retenção do Bloqueio de objetos do S3](#)
- [Retenção legal do Bloqueio de objetos do S3](#)
- [Replicar objetos existentes com o S3 Batch Replication](#)

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](#)
- [Listar trabalhos](#)
- [Visualizar detalhes do trabalho](#)
- [Atribuir prioridade aos trabalhos](#)

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
```

Para obter mais informações e exemplos, consulte [list-jobs](#) na Referência de comandos da AWS CLI.

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. É possível ver detalhes dos trabalhos que ainda não foram concluídos ou dos que foram concluídos nos últimos 90 dias. 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. Para visualizar os detalhes de um trabalho, você pode usar o console do Amazon S3 ou a AWS Command Line Interface (AWS CLI).

Ter a descrição de um trabalho de Operações em Lote do S3 no console do Amazon S3

Como visualizar a descrição de um trabalho de Operações em Lote 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 ID do trabalho específico para visualizar os detalhes.

Ter a descrição de um trabalho de Operações em Lote do S3 no AWS CLI

O exemplo a seguir obtém a descrição de um trabalho de Operações em Lote do S3 usando a AWS CLI. Para usar o comando a seguir, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control describe-job \  
--region us-west-2 \  
--account-id acct-id \  
--job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

Para obter mais informações e exemplos, consulte [describe-job](#) na Referência de comandos da AWS CLI.

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).

Usando 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](#).

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](#)
- [Atualizar status do trabalho](#)
- [Notificações e registro em log](#)
- [Monitorar falhas nos trabalhos](#)
- [Relatórios de conclusão](#)
- [Monitorar um trabalho de operações em lote do S3 no Amazon EventBridge por meio do AWS CloudTrail](#)
- [Exemplos: relatórios de conclusão de operações em lote do S3](#)

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 <code>Preparing</code> 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.	<p>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.</p> <p>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.</p>
Suspended	<p>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.</p>	Depois de confirmar que deseja executar o trabalho, o status muda para Ready.

Status	Descrição	Transições
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 concluí-los.
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 DescribeJobs 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.

Status	Descrição	Transições
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.

Status	Descrição	Transições
Failing	O trabalho está passando para o estado Failed.	O trabalho passa automaticamente para Failed quando o estado Failing é concluído.
Failed	O trabalho apresentou falha e não está mais em execução. Para obter mais informações sobre falhas em trabalhos, consulte Monitorar falhas nos trabalhos .	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](#).

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](#).

Example

```
package aws.example.s3control;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import com.amazonaws.auth.profile.ProfileCredentialsProvider;  
import com.amazonaws.services.s3control.AWSS3Control;  
import com.amazonaws.services.s3control.AWSS3ControlClient;  
import com.amazonaws.services.s3control.model.UpdateJobStatusRequest;  
  
import static com.amazonaws.regions.Regions.US_WEST_2;  
  
public class UpdateJobStatus {  
    public static void main(String[] args) {  
        String accountId = "Account ID";  
        String jobId = "00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c";  
  
        try {  
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()  
                .withCredentials(new ProfileCredentialsProvider())  
                .withRegion(US_WEST_2)  
                .build();  
  
            s3ControlClient.updateJobStatus(new UpdateJobStatusRequest()  
                .withAccountId(accountId)
```

```
        .withJobId(jobId)
        .withRequestedJobStatus("Ready"));

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

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](#).

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 excedeu o limite de falhas da tarefa, você poderá identificar a causa das falhas. Por exemplo, você pode ter incluído por acidente alguns objetos no manifesto que não existem no bucket especificado. Depois de corrigir os erros, você pode reenviar o trabalho.

Note

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](#).

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](#)
- [Exemplos: relatórios de conclusão de operações em lote do S3](#)

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](#)
- [Regra do EventBridge para rastrear eventos de trabalhos de operações em lote do S3](#)

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.

Exemplo 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 [Criação de regras do EventBridge que reagem aos eventos](#). 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 S3

```
{
  "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. Contudo 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.

Exemplo 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": "22ee037f3515975f7719699e5c416eaa",
      "Key": "job-f8fb9d89-a3aa-461d-bddc-ea6a1b131955/results/
b2ddad417e94331e9f37b44f1faf8c7ed5873f2e.csv"
    }
  ],
  "ReportSchema": "Bucket, Key, VersionId, TaskStatus, ErrorCode, HTTPStatusCode,
ResultMessage"
}
```

Exemplo 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 marcada 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': 'nXNaClIMxEJzWNmeMNQV2KpjbaCJLn00GoXWZpuV0FS/
iQYWxb3QtTvzX9SVfx2lA3oTKLwImKw=', 'RequestId': '3ED5852152014362', 'HTTPHeaders':
{'content-length': '234', 'x-amz-id-2': 'nXNaClIMxEJzWNmeMNQV2KpjbaCJLn00GoXWZpuV0FS/
iQYWxb3QtTvzX9SVfx2lA3oTKLwImKw=', 'x-amz-copy-source-version-id':
'xVR78haVKlRnurYofbTfYr3ufYbktF8h', 'server': 'AmazonS3', 'x-amz-request-id':
'3ED5852152014362', 'date': 'Fri, 05 Apr 2019 17:35:39 GMT', 'content-type':
'application/xml'}}}"
awsexamplebucket1,image_17763,,succeeded,200,, "{u'CopySourceVersionId':
'6Hj0USim4Wj6BTcbxToXW44pSZ.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': '6Hj0USim4Wj6BTcbxToXW44pSZ.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': 'F9ooZ0gpE5g9sNgBZxjdiPHqB4+0DNWgj3qbsir
+sKai4fv7rQEcF2fBN1VeeFc2WH45a9ygb2g=', 'RequestId': '8D9CA56A56813DF3', 'HTTPHeaders':
{'content-length': '234', 'x-amz-id-2': 'F9ooZ0gpE5g9sNgBZxjdiPHqB4+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”](#).

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”](#).

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
```

- 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
```

- 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-5dc7a8fb90808fc5d546218"
  }
}
EOF
```

- 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.getJobId();
}
```

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.

1. 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
```

2. 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](#). Consulte mais informações em [Actions, resources, and condition keys for Amazon S3](#) na Referência de autorização do serviço.

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](#)
- [Marcar trabalhos de operações em lote por estágio e impor limites na prioridade do trabalho](#)

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 etiquetas a usuários 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 etiqueta department=Finance ao usuário.

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/departament}`", que é substituída pela etiqueta de departamento do usuário no momento da avaliação da política. A condição é satisfeita quando o valor fornecido para a tag de departamento na solicitação ("`aws:RequestTag/departament`") 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": "s3:CreateJob",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/departament": "${aws:PrincipalTag/
departament}"
        }
      }
    },
    {
      "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](#).

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](#)
- [Definir a retenção do Bloqueio de objetos usando o Operações em lote](#)
- [Usar operações em lote do S3 com o modo de conformidade de retenção do bloqueio de objetos do S3](#)
- [Usar operações em lote do S3 com modo de governança de retenção de bloqueio de objetos do S3](#)
- [Usar operações em lote do S3 para desativar a retenção legal do bloqueio de objetos do S3](#)

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”](#) e [the section called “Retenção legal do Bloqueio de objetos”](#).

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 “Especificar um manifesto”](#).

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”](#).

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 1.º de janeiro de 2025. 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 **retain until date** do modo **COMPLIANCE** para 15 de janeiro de 2025

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 2025";

```

Usar o AWS SDK for Java

Example Defina o modo de retenção como COMPLIANCE e o prazo de retenção até 1.º de janeiro de 2025.

```

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("01/01/2025");

final JobOperation jobOperation = new JobOperation()
    .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
        .withRetention(new S3Retention()
            .withMode(S3ObjectLockRetentionMode.COMPLIANCE)
            .withRetainUntilDate(janFirst)));

final String roleArn = "arn:aws:iam::123456789012:role/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 2025")
    .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 COMPLIANCE do modo `retain until date` para 15 de janeiro de 2025.

```
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("15/01/2025");

    final JobOperation jobOperation = new JobOperation()
      .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
        .withRetention(new S3Retention()
          .withMode(S3ObjectLockRetentionMode.COMPLIANCE)
          .withRetainUntilDate(jan15th)));

    final String roleArn = "arn:aws:iam::123456789012:role/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 2025")
        .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

Exemplo 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 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":"2025-01-30T00:00:00",
      "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";
```

Exemplo 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.

Mostra como aplicar a governança de retenção do Bloqueio de Objetos do S3 com `retain until` date definido como 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.

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 2025

```
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("30/01/2025");

    final JobOperation jobOperation = new JobOperation()
        .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
            .withRetention(new S3Retention()
                .withMode(S3ObjectLockRetentionMode.GOVERNANCE)
```

```

        .withRetainUntilDate(jan30th)));

final String roleArn = "arn:aws:iam::123456789012:role/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 AWS Elemental MediaConvert](#)

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.

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Tópicos

- [Ferramentas de monitoramento](#)
- [Opções de registro em log para o Amazon S3](#)
- [Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail](#)
- [Registrar em log as solicitações com registro em log de acesso ao servidor](#)
- [Monitoramento de métricas com o Amazon CloudWatch](#)
- [Notificações de eventos do Amazon S3](#)

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](#).
- **AWS CloudTrail Monitoramento 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](#).

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](#).

- 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 obter 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, perfis ou Serviços 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 que você use o CloudTrail para registrar em log ações por bucket e objeto para seus 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](#)
- [Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail](#)

A tabela a seguir lista as principais propriedades dos logs do CloudTrail e dos logs de acesso ao servidor do Amazon S3. Analise a tabela e as observações para garantir que o CloudTrail atenda aos seus requisitos de segurança.

Propriedades de log	AWS CloudTrail	Logs do servidor do Amazon S3
Pode ser encaminhado para outros sistemas (Amazon CloudWatch Logs, Amazon CloudWatch Events)	Sim	No
Entregue logs para mais de um destino (por exemplo, envie os mesmos logs para dois buckets diferentes)	Sim	No
Ative logs para um subconjunto de objetos (prefixo)	Sim	No
Entrega de logs entre contas (bucket de origem e de destino pertencentes a contas diferentes)	Sim	No
Validação de integridade do arquivo de log usando assinatura digital ou hashing	Sim	No
Padrão ou opção de criptografia para arquivos de log	Sim	No
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	No

Propriedades de log	AWS CloudTrail	Logs do servidor do Amazon S3
Campos para parâmetros de bloqueio de objetos, propriedades selecionadas do Amazon S3 para registros de log	Sim	No
Campos de Object Size, Total Time, Turn-Around Time e HTTP Referer para registros de log	Não	Sim
Transições, expirações e restaurações do ciclo de vida	Não	Sim
Registro de chaves em uma operação de exclusão em lote	Não	Sim
Falhas de autenticação ¹	Não	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
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

Propriedades de log	AWS CloudTrail	Logs do servidor do Amazon S3
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 quando a conta não tem acesso total ao objeto na solicitação. Para ter mais informações, consulte [Ações em nível de objeto do Amazon S3 em cenários entre contas](#).
3. O S3 não é compatível com a entrega de logs do CloudTrail ou logs de acesso ao servidor ao solicitante ou ao proprietário do bucket para solicitações de endpoint da VPC quando a política de endpoint da VPC as nega.

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 realizadas por um usuário, um perfil ou um AWS service (Serviço da AWS). O CloudTrail captura todas as chamadas de API para o Amazon S3 como eventos. As chamadas capturadas incluem aquelas do console do Amazon S3 e as chamadas de código para as operações de API do Amazon S3. Ao fazer uso das informações coletadas pelo CloudTrail, é possível determinar a solicitação feita ao Amazon S3, o endereço IP que a realizou, quando ela foi feita e detalhes adicionais.

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 credenciais de usuário.
- Se a solicitação foi feita em nome de um usuário do Centro de Identidade 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).

O CloudTrail está ativo em sua Conta da AWS e você tem acesso automático ao Histórico de eventos do CloudTrail. O Histórico de eventos do CloudTrail fornece um registro visualizável, pesquisável, baixável e imutável dos últimos 90 dias de eventos de gerenciamento gravados em uma Região da AWS. Para obter mais informações, consulte [Trabalhar com histórico de eventos do CloudTrail](#) no Guia do usuário do AWS CloudTrail. Não há cobranças do CloudTrail pela visualização do Histórico de eventos.

Para obter um registro contínuo de eventos em sua Conta da AWS nos últimos 90 dias, crie uma trilha ou um armazenamento de dados de eventos do [CloudTrail Lake](#).

Trilhas do CloudTrail

Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket Amazon S3. As trilhas criadas usando o AWS Management Console são de várias regiões. Só é possível criar uma trilha de região única ou de várias regiões usando a AWS CLI. Criar uma trilha de várias regiões é uma prática recomendada, pois você captura atividades em todas as Regiões da AWS da conta. Se você criar uma trilha de região única, poderá visualizar somente os eventos registrados na Região da AWS da trilha. Para obter mais informações sobre trilhas, consulte [Criar uma trilha para a Conta da AWS](#) e [Criar uma trilha para uma organização](#) no Guia do usuário do AWS CloudTrail.

Uma cópia dos seus eventos de gerenciamento em andamento pode ser entregue no bucket do Amazon S3 sem nenhum custo via CloudTrail com a criação de uma trilha. No entanto, há cobranças de armazenamento do Amazon S3. Para obter mais informações sobre o preço do CloudTrail, consulte [AWS CloudTrail Preço do](#). Para receber informações sobre a definição de preço do Amazon S3, consulte [Definição de preço do Amazon S3](#).

Armazenamentos de dados de eventos do CloudTrail Lake

O CloudTrail Lake permite executar consultas baseadas em SQL em seus eventos. O CloudTrail Lake converte eventos existentes em formato JSON baseado em linhas para o formato [Apache ORC](#). O ORC é um formato colunar de armazenamento otimizado para recuperação rápida de dados. Os eventos são agregados em armazenamentos de dados de eventos, que são coleções imutáveis de eventos baseados nos critérios selecionados com a aplicação de [seletores de eventos avançados](#). Os seletores que você aplica a um armazenamento de dados de eventos controlam quais eventos persistem e estão disponíveis para você consultar. Para obter mais

informações sobre o CloudTrail Lake, consulte [Trabalhar com o AWS CloudTrail Lake](#), no Guia do usuário do AWS CloudTrail.

Os armazenamentos de dados de eventos e consultas do CloudTrail Lake incorrem em custos. Ao criar um armazenamento de dados de eventos, você escolhe a [opção de preço](#) que deseja usar para ele. A opção de preço determina o custo para a ingestão e para o armazenamento de eventos, e o período de retenção padrão e máximo para o armazenamento de dados de eventos. Para obter mais informações sobre o preço do CloudTrail, consulte [AWS CloudTrail Preço do](#).

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.

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 ao 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”](#).

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](#).

Você também pode usar logs do CloudTrail junto com o Amazon 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](#).

Note

O S3 não é compatível com a entrega de logs do CloudTrail ao solicitante ou ao proprietário do bucket para solicitações de endpoint da VPC quando a política de endpoint da VPC as nega.

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](#).

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	GetBucketAc1
SetBucketAccessControlPolicy	PutBucketAc1
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](#)
- [Entradas de arquivo de log do CloudTrail para Amazon S3 e S3 on Outposts](#)
- [Habilitar o log de eventos do CloudTrail para buckets e objetos do S3](#)
- [Identificar solicitações do Amazon S3 usando o CloudTrail](#)

Eventos do CloudTrail no Amazon S3

Important

O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).

Esta seção fornece informações sobre os eventos que o S3 registra em log no CloudTrail.

Eventos de dados do Amazon S3 no CloudTrail

Os [eventos de dados](#) fornecem informações sobre as operações de recursos realizadas em um recurso (por exemplo, leitura ou gravação em um objeto do Amazon S3). Elas também são conhecidas como operações de plano de dados. Eventos de dados geralmente são atividades de alto volume. Por padrão, o CloudTrail não registra eventos de dados em log. O Histórico de eventos do CloudTrail não registra eventos de dados.

Há cobranças adicionais para eventos de dados. Para obter mais informações sobre o preço do CloudTrail, consulte [AWS CloudTrailPreço do](#).

É possível registrar em log eventos de dados para os tipos de recurso do Amazon S3 usando o console do CloudTrail, a AWS CLI ou as operações de API do CloudTrail. Para obter mais

informações sobre como registrar eventos de dados em log, consulte [Registrar eventos de dados com o AWS Management Console](#) e [Registrar eventos de dados com a AWS Command Line Interface](#) no Guia do usuário do AWS CloudTrail.

A tabela a seguir lista o tipo de recurso do Amazon S3 para o qual você pode registrar eventos de dados em log. A coluna Tipo de evento de dados (console) mostra o valor a ser escolhido na lista Tipo de evento de dados no console do CloudTrail. A coluna do valor `resources.type` mostra o valor de `resources.type` que você especificaria ao configurar seletores de eventos avançados usando a AWS CLI ou as APIs do CloudTrail. A coluna APIs de dados registradas no CloudTrail mostra as chamadas de API registradas no CloudTrail para o tipo de recurso.

Tipo de evento de dados (console)	valor <code>resources.type</code>	APIs de dados registradas no CloudTrail
S3	AWS::S3::Object	<ul style="list-style-type: none"> • AbortMultipartUpload • CompleteMultipartUpload • CopyObject • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • DeleteObjects • GetObject • GetObjectAcl • GetObjectAttributes • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • GetObjectTorrent • HeadObject • ListMultipartUploads • ListObjectVersions • ListObjects • ListParts

Tipo de evento de dados (console)	valor resources.type	APIs de dados registradas no CloudTrail
		<ul style="list-style-type: none">• PutObject• PutObjectAcl• PutObjectLegalHold• PutObjectRetention• PutObjectTagging• RestoreObject• SelectObjectContent• UploadPart• UploadPartCopy

Tipo de evento de dados (console)	valor resources.type	APIs de dados registradas no CloudTrail
Ponto de acesso do S3	AWS::S3::Access Point	<ul style="list-style-type: none"> • AbortMultipartUpload • CompleteMultipartUpload • CopyObject (somente cópias da mesma região) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetBucketAcl • GetBucketCors • GetBucketLocation • GetBucketNotificationConfiguration • GetBucketPolicy • GetObject • GetObjectAcl • GetObjectAttributes • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • HeadBucket • HeadObject • ListMultipartUploads • ListObjects • ListObjectsV2 • ListObjectVersions • ListParts • Presign • PutObject

Tipo de evento de dados (console)	valor resources.type	APIs de dados registradas no CloudTrail
		<ul style="list-style-type: none">• PutObjectLegalHold• PutObjectRetention• PutObjectAcl• PutObjectTagging• RestoreObject• UploadPart• UploadPartCopy (somente cópias da mesma região)

Tipo de evento de dados (console)	valor resources.type	APIs de dados registradas no CloudTrail
S3 Object Lambda	AWS::S3ObjectLambda::AccessPoint	<ul style="list-style-type: none">• AbortMultipartUpload• CompleteMultipartUpload• CopyObject (somente cópias da mesma região)• CreateMultipartUpload• DeleteObject• DeleteObjectTagging• GetObject• GetObjectAcl• GetObjectLegalHold• GetObjectRetention• GetObjectTagging• HeadObject• ListMultipartUploads• ListObjects• ListObjectVersions• ListParts• PutObject• PutObjectLegalHold• PutObjectRetention• PutObjectAcl• PutObjectTagging• RestoreObject• UploadPart• WriteGetObjectResponse

Tipo de evento de dados (console)	valor resources.type	APIs de dados registradas no CloudTrail
S3 Outposts	AWS::S3Outposts::Object	<ul style="list-style-type: none"> • AbortMultipartUpload • CompleteMultipartUpload • CopyObject (somente cópias da mesma região) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetObject • GetObjectTagging • HeadObject • ListMultipartUploads • ListObjects • ListObjectsV2 • ListParts • PutObject • PutObjectTagging • UploadPart • UploadPartCopy

É possível configurar seletores de eventos avançados para filtrar os campos `eventName`, `readOnly` e `resources.ARN` para registrar somente os eventos que são importantes para você. Para obter mais informações sobre esses campos, consulte [AdvancedFieldSelector](#) na Referência de API do AWS CloudTrail.

Eventos de gerenciamento do Amazon S3 no CloudTrail

O Amazon S3 registra em log todas as operações do ambiente de gerenciamento como eventos de gerenciamento. Consulte mais informações sobre as operações de API do S3 na [Referência de API 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 de buckets incluem eventos como `CreateBucket`, `DeleteBucket`, `PutBucketLifecycle`, `PutBucketPolicy` e assim por diante. 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 AWS service (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. Os nomes dos eventos do CloudTrail diferem do nome da ação da API. Por exemplo, `DeletePublicAccessBlock` é `DeleteAccountPublicAccessBlock`.

- [DeleteAccountPublicAccessBlock](#)
- [GetAccountPublicAccessBlock](#)
- [PutAccountPublicAccessBlock](#)

Ações de buckets do Amazon S3 rastreadas pelo registro em log do CloudTrail

Por padrão, o CloudTrail registra em log as ações nos buckets de uso geral. 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.

Esta seção lista as ações no 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. Em alguns casos, o

nome do evento do CloudTrail é diferente do nome da ação da API. Por exemplo, `PutBucketLifecycleConfiguration` é `PutBucketLifecycle`.

- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteBucketAnalyticsConfiguration](#)
- [DeleteBucketCors](#)
- [DeleteBucketEncryption](#)
- [DeleteBucketIntelligentTieringConfiguration](#)
- [DeleteBucketInventoryConfiguration](#)
- [DeleteBucketLifecycle](#)
- [DeleteBucketMetricsConfiguration](#)
- [DeleteBucketOwnershipControls](#)
- [DeleteBucketPolicy](#)
- [DeleteBucketPublicAccessBlock](#)
- [DeleteBucketReplication](#)
- [DeleteBucketTagging](#)
- [GetAccelerateConfiguration](#)
- [GetBucketAcl](#)
- [GetBucketAnalyticsConfiguration](#)
- [GetBucketCors](#)
- [GetBucketEncryption](#)
- [GetBucketIntelligentTieringConfiguration](#)
- [GetBucketInventoryConfiguration](#)
- [GetBucketLifecycle](#)
- [GetBucketLocation](#)
- [GetBucketLogging](#)
- [GetBucketMetricsConfiguration](#)
- [GetBucketNotification](#)
- [GetBucketObjectLockConfiguration](#)
- [GetBucketOwnershipControls](#)

- [GetBucketPolicy](#)
- [GetBucketPolicyStatus](#)
- [GetBucketPublicAccessBlock](#)
- [GetBucketReplication](#)
- [GetBucketRequestPayment](#)
- [GetBucketTagging](#)
- [GetBucketVersioning](#)
- [GetBucketWebsite](#)
- [HeadBucket](#)
- [ListBuckets](#)
- [PutAccelerateConfiguration](#)
- [PutBucketAcl](#)
- [PutBucketAnalyticsConfiguration](#)
- [PutBucketCors](#)
- [PutBucketEncryption](#)
- [PutBucketIntelligentTieringConfiguration](#)
- [PutBucketInventoryConfiguration](#)
- [PutBucketLifecycle](#)
- [PutBucketLogging](#)
- [PutBucketMetricsConfiguration](#)
- [PutBucketNotification](#)
- [PutBucketObjectLockConfiguration](#)
- [PutBucketOwnershipControls](#)
- [PutBucketPolicy](#)
- [PutBucketPublicAccessBlock](#)
- [PutBucketReplication](#)
- [PutBucketRequestPayment](#)
- [PutBucketTagging](#)
- [PutBucketVersioning](#)
- [PutBucketWebsite](#)

Além dessas operações de API, também é possível usar a ação de objetos [OPTIONS object](#). 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 do bucket da classe S3 Express One Zone (endpoint de API regional) monitoradas pelo registro em log do CloudTrail

Por padrão, o CloudTrail registra em log as ações nos buckets de diretório como eventos de gerenciamento. O `eventsSource` para os eventos de gerenciamento do CloudTrail para a classe S3 Express One Zone é `s3express.amazonaws.com`.

Note

Para a classe S3 Express One Zone, o registro em log do CloudTrail de operações de API do endpoint zonal (nível de objeto ou plano de dados) (por exemplo, `PutObject` ou `GetObject`) não é compatível.

Veja a seguir as operações de API de endpoint regional que são registradas em log no CloudTrail.

- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteBucketPolicy](#)
- [GetBucketPolicy](#)
- [PutBucketPolicy](#)
- [ListDirectoryBuckets](#)

Para ter mais informações, consulte [Práticas recomendadas de segurança para a classe S3 Express One Zone](#).

Ações em nível de objeto do Amazon S3 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 logs ao solicitante (a conta que fez a chamada de API), exceto em alguns casos de acesso negado em que as entradas de log são editadas ou omitidas. Para estabelecer acesso entre contas, considere os exemplos nesta seção.

Note

Os exemplos supõem que os logs do CloudTrail estejam configurados adequadamente.

Exemplo 1: CloudTrail entrega logs ao proprietário do bucket

O CloudTrail entrega os logs de acesso ao proprietário do bucket mesmo que o proprietário do bucket não tenha permissão para a mesma operação de API do objeto. Considere o seguinte cenário entre contas:

- A conta A é proprietária do bucket.
- A conta B (o solicitante) tenta acessar um objeto nesse bucket.
- A conta C é proprietária do objeto. A conta C pode ou não ser igual à conta A.

Note

O CloudTrail sempre entrega os logs de API de objetos ao solicitante (conta B). Além disso, o CloudTrail também entrega os mesmo logs ao proprietário do bucket (conta A) mesmo quando o proprietário do bucket (conta C) não é proprietário do objeto (conta C) nem tem permissões para as mesmas operações de API sobre esse objeto.

Exemplo 2: CloudTrail não prolifera os endereços de e-mail usados na definição de ACLs de objeto

Considere o seguinte cenário entre contas:

- A conta A é proprietária do 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 \(ACL\)](#).

O solicitante recebe 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 endereço de 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

Important

O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).

Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a operação solicitada, a data e a hora da operação da API, os parâmetros de solicitação etc. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada de chamadas de API pública, portanto 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](#)
- [Exemplo: entradas de arquivo de log do Amazon S3 no Outposts](#)

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 [GET Service](#), [PutBucketAcl](#) e [GetBucketVersioning](#).

```
{
  "Records": [
    {
      "eventVersion": "1.03",
      "userIdentity": {
```

```

    "type": "IAMUser",
    "principalId": "111122223333",
    "arn": "arn:aws:iam::111122223333:user/myUserName",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "myUserName"
  },
  "eventTime": "2019-02-01T03:18:19Z",
  "eventSource": "s3.amazonaws.com",
  "eventName": "ListBuckets",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "[]",
  "requestParameters": {
    "host": [
      "s3.us-west-2.amazonaws.com"
    ]
  },
  "responseElements": null,
  "additionalEventData": {
    "SignatureVersion": "SigV2",
    "AuthenticationMethod": "QueryString",
    "aclRequired": "Yes"
  },
  "requestID": "47B8E8D397DCE7A6",
  "eventID": "cdc4b7ed-e171-4cef-975a-ad829d4123e8",
  "eventType": "AwsApiCall",
  "recipientAccountId": "444455556666",
  "tlsDetails": {
    "tlsVersion": "TLSv1.2",
    "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
    "clientProvidedHostHeader": "s3.amazonaws.com"
  }
},
{
  "eventVersion": "1.03",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "111122223333",
    "arn": "arn:aws:iam::111122223333:user/myUserName",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "myUserName"
  },

```



```

"eventTime": "2019-02-01T03:22:33Z",
"eventSource": "s3.amazonaws.com",
"eventName": "PutBucketAcl",
"awsRegion": "us-west-2",
"sourceIPAddress": "",
"userAgent": "[]",
"requestParameters": {
  "bucketName": "",
  "AccessControlPolicy": {
    "AccessControlList": {
      "Grant": {
        "Grantee": {
          "xsi:type": "CanonicalUser",
          "xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
          "ID":
"d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
        },
        "Permission": "FULL_CONTROL"
      }
    },
    "xmlns": "http://s3.amazonaws.com/doc/2006-03-01/",
    "Owner": {
      "ID":
"d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
    }
  },
  "host": [
    "s3.us-west-2.amazonaws.com"
  ],
  "acl": [
    ""
  ]
},
"responseElements": null,
"additionalEventData": {
  "SignatureVersion": "SigV4",
  "CipherSuite": "ECDHE-RSA-AES128-SHA",
  "AuthenticationMethod": "AuthHeader"
},
"requestID": "BD8798EACDD16751",
"eventID": "607b9532-1423-41c7-b048-ec2641693c47",
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333",
"tlsDetails": {

```

```

        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "clientProvidedHostHeader": "s3.amazonaws.com"
    }
},
{
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "111122223333",
        "arn": "arn:aws:iam::111122223333:user/myUserName",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "myUserName"
    },
    "eventTime": "2019-02-01T03:26:37Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "GetBucketVersioning",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "",
    "userAgent": "[]",
    "requestParameters": {
        "host": [
            "s3.us-west-2.amazonaws.com"
        ],
        "bucketName": "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",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "clientProvidedHostHeader": "s3.amazonaws.com"
    }
}

```

```
}
}
]
}
```

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](#). 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": "29xXQBv20
+x0HKItvzY1suLv1i6A52E0z0X159fpfsItYd58JhXwKxXAXI4IQkp6",
    "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:YYY:outpost/op-01ac5d28a6a232904/
bucket/path/upload.sh"
    },
    {

```

```
    "accountId": "222222222222",
    "type": "AWS::S3Outposts::Bucket",
    "ARN": "arn:aws:s3-outposts:us-east-1:YYY: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 ajuda você a criar 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 registrar em log eventos de dados de uma trilha no CloudTrail, você pode optar por usar seletores de eventos avançados ou básicos. Ao criar uma trilha no console do CloudTrail usando seletores de eventos avançados, na seção de eventos de dados, é possível selecionar Registrar todos os eventos no Modelo de seletor de log para registrar em log todos os eventos no nível de objeto. Ao criar uma trilha no console do CloudTrail usando seletores de eventos básicos, na seção de eventos de dados, marque a caixa de seleção Selecionar todos os buckets do S3 em sua conta para registrar em log todos os eventos no nível de objeto.

Note

- É uma prática recomendada criar uma configuração de ciclo de vida para o bucket de evento de dados do AWS CloudTrail. Configure a configuração 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 uma configuração de ciclo de vida em um bucket](#).
- Para obter mais informações sobre o formato do registro em log, consulte [Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail](#).
- 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](#).

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 de buckets do Amazon S3 rastreadas pelo registro em log do CloudTrail](#). 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 Eventos de dados do AWS CloudTrail, selecione 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

1. Faça login no AWS Management Console e abra o console do CloudTrail em <https://console.aws.amazon.com/cloudtrail/>.
2. No painel de navegação à esquerda, selecione Trilhas.
3. Escolha o nome da trilha que você criou para registrar eventos para o bucket.
4. Na página de detalhes da trilha, escolha Parar o registro no canto superior direito.
5. Na caixa de diálogo exibida, selecione Parar o registro.

Para obter informações sobre como habilitar o registro no nível do objeto ao criar um bucket do S3, consulte [Criação de um bucket](#).

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](#)
- [Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail](#)
- [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”](#).

Tópicos

- [Identificar solicitações feitas ao Amazon S3 em um log do CloudTrail](#)
- [Identificar solicitações de Signature Version 2 do Amazon S3 usando o CloudTrail](#)
- [Identificar o acesso a objetos do S3 usando o CloudTrail](#)

Identificar solicitações feitas ao Amazon S3 em um log 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://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/Region/yyyy/mm/dd
```

Eventos registrados pelo CloudTrail são armazenados como objetos JSON compactados no formato gzipped 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 [Criar a tabela de logs do AWS CloudTrail no Athena usando projeção de partições](#) no Guia do usuário do Amazon Athena.

Identificar solicitações de Signature Version 2 do Amazon S3 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 (suspensão). 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: AWS CloudTrail para Amazon S3 adiciona novos campos para auditoria de segurança aprimorada](#) no AWS re:Post.

Note

Os eventos do CloudTrail para o Amazon S3 incluem a versão da assinatura 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 GET, PUT e DELETE,

you must enable CloudTrail data events. (This resource is disabled by default.)

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 Version 2 usando logs de acesso do Amazon S3](#).

Tópicos

- [Exemplos de consulta do Athena para identificar solicitações do Amazon S3 Signature versão 2](#)
- [Particionar dados do Signature versão 2](#)

Exemplos de consulta do Athena para identificar solicitações do Amazon S3 Signature versão 2

Example : Selecione todos os eventos do Signature Version 2 e imprima somente **EventTime**, **S3_Action**, **Request_Parameters**, **Region**, **SourceIP** e **UserAgent**

Na consulta do Athena a seguir, substitua *s3_cloudtrail_events_db.cloudtrail_table* pelos seus 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_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_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_table_partitioned(  
  eventversion STRING,  
  userIdentity STRUCT<  
    type:STRING,  
    principalid:STRING,  
    arn:STRING,  
    accountid:STRING,  
    invokedby:STRING,  
    accesskeyid:STRING,  
    userName:STRING,  
    sessioncontext:STRUCT<  
      attributes:STRUCT<  
        mfaauthenticated:STRING,  
        creationdate:STRING>,  
      sessionIssuer:STRUCT<  
        type:STRING,  
        principalId:STRING,  
        arn:STRING,  
        accountId:STRING,  
        userName:STRING>  
    >  
  >,  
  eventTime STRING,  
  eventSource STRING,  
  eventName STRING,  
  awsRegion STRING,  
  sourceIpAddress STRING,  
  userAgent STRING,  
  errorCode STRING,  
  errorMessage STRING,  
  requestParameters STRING,  
  responseElements STRING,  
  additionalEventData STRING,  
  requestId STRING,  
  eventId STRING,  
  resources ARRAY<STRUCT<ARN:STRING,accountId: STRING,type:STRING>>,  
  >
```

```

    eventType STRING,
    apiVersion STRING,
    readOnly STRING,
    recipientAccountId STRING,
    serviceEventDetails STRING,
    sharedEventID STRING,
    vpcEndpointId STRING
)
PARTITIONED BY (region string, year string, month string, day string)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://DOC-EXAMPLE-BUCKET1/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_table_partitioned ADD
  PARTITION (region= 'us-east-1', year= '2019', month= '02', day= '19') LOCATION
  's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/us-east-1/2019/02/19/'
  PARTITION (region= 'us-west-1', year= '2019', month= '02', day= '19') LOCATION
  's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/us-west-1/2019/02/19/'
  PARTITION (region= 'us-west-2', year= '2019', month= '02', day= '19') LOCATION
  's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/us-west-2/2019/02/19/'
  PARTITION (region= 'ap-southeast-1', year= '2019', month= '02', day= '19') LOCATION
  's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/ap-southeast-1/2019/02/19/'
  PARTITION (region= 'ap-southeast-2', year= '2019', month= '02', day= '19') LOCATION
  's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/ap-southeast-2/2019/02/19/'
  PARTITION (region= 'ap-northeast-1', year= '2019', month= '02', day= '19') LOCATION
  's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/ap-northeast-1/2019/02/19/'
  PARTITION (region= 'eu-west-1', year= '2019', month= '02', day= '19') LOCATION
  's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/eu-west-1/2019/02/19/'
  PARTITION (region= 'sa-east-1', year= '2019', month= '02', day= '19') LOCATION
  's3://DOC-EXAMPLE-BUCKET1/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_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 seus logs 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 PUT de objetos para o Amazon S3 de um log de eventos do AWS CloudTrail.

Tópicos

- [Exemplos de consulta do Athena para identificar solicitações de acesso a objetos do Amazon S3](#)

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_table* pelos seus detalhes do Athena e modifique o intervalo de datas, conforme necessário.

Example : Selecione todos os eventos que tenham solicitações **PUT** de acesso a objetos 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_table  
WHERE  
  eventName = 'PutObject'  
  AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example : Selecione todos os eventos que tenham solicitações **GET** de acesso a objetos 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_table  
WHERE  
  eventName = 'GetObject'  
  AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example : Selecione todos os eventos de solicitante anônimo em um bucket em determinado período e imprima 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_table
```

```
WHERE
  userIdentity.accountId = 'anonymous'
  AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example : Identifique todas as solicitações que exigiram uma ACL para autorização

O exemplo de consulta do Amazon Athena a seguir mostra como identificar todas as solicitações feitas aos seus buckets do S3 que exigiram uma lista de controle de acesso (ACL) para autorização. Se a solicitação exigiu uma ACL para autorização, o valor de `aclRequired` em `additionalEventData` é `Yes`. Se nenhuma ACL foi necessária, `aclRequired` não está presente. Você pode usar essas informações para migrar essas permissões de ACL para as políticas de bucket apropriadas. Depois de criar essas políticas de bucket, você pode desativar as ACLs para esses buckets. Para obter mais informações sobre como desativar as ACLs, consulte [Pré-requisitos para desabilitar ACLs](#).

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  userIdentity.arn as userArn,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  json_extract_scalar(requestParameters, '$.key') as object,
  json_extract_scalar(additionalEventData, '$.aclRequired') as aclRequired
FROM
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  json_extract_scalar(additionalEventData, '$.aclRequired') = 'Yes'
  AND eventTime BETWEEN '2022-05-10T00:00:00Z' and '2022-08-10T00:00:00Z'
```

Note

- 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 ou endereços IP inesperados ou não autorizados e identificar solicitações anônimas aos seus buckets.
- 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](#).

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. Essas informações também podem ajudar você a conhecer sua base de clientes e entender a cobrança 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 detalhes, consulte [Habilitar o log de acesso ao servidor do Amazon S3](#).

1. Forneça o nome do bucket de destino (também conhecido como bucket-alvo). 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. O bucket de destino não pode ter uma configuração de período de retenção padrão da funcionalidade Bloqueio de Objetos do S3. O bucket de destino também não pode ter a opção Pagamento pelo solicitante habilitada.

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 serão criados para os logs que forem gravados no bucket, o que criará um ciclo infinito de logs.

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 especifique um prefixo de destino (também conhecido como prefixo-alvo) para todas as chaves de objeto de log. Quando você especifica um prefixo, todos os nomes de objeto de log começam com uma string comum, o que facilita a identificação dos objetos de log.

2. (Opcional) Atribua um prefixo de destino a todas as chaves de objeto de log do Amazon S3. O prefixo de destino (também conhecido como prefixo-alvo) 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, por exemplo:

```
logs/2013-11-01-21-32-16-E568B2907131C0C0
```

Se você especificar o valor de prefixo `logs`, o objeto de log será exibido da seguinte forma:

```
logs2013-11-01-21-32-16-E568B2907131C0C0
```

Os [prefixos](#) também são úteis para distinguir entre buckets de origem quando vários buckets são registrados em log no mesmo bucket de destino.

Esse prefixo também ajuda ao excluir os logs. Por exemplo, você pode definir uma regra de configuração do ciclo de vida para que o Amazon S3 exclua objetos com um prefixo específico. Para ter mais informações, consulte [Exclusão de arquivos de log do Amazon S3](#).

3. (Opcional) Defina permissões 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 usar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos do S3 a fim de desabilitar as listas de controle de acesso (ACLs), você não pode conceder permissões nas concessões de destino (também chamadas de concessões-alvo) que usam ACLs. Porém, você pode atualizar a política do bucket de destino para conceder acesso a outros. Para obter mais informações, consulte [Gerenciamento de identidade e acesso para o Amazon S3](#) e [Permissões para entrega de logs](#).
4. (Opcional) Defina um formato de chave de objeto de log para os arquivos de log. Você tem duas opções para o formato da chave de objeto de log (também conhecido como formato da chave de objeto-alvo):

- **Particionamento não baseado em data:** esse é o formato original da chave de objeto de log. Se você escolher esse formato, o formato da chave de arquivo de log será exibido da seguinte forma:

```
[DestinationPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Por exemplo, se você especificar `logs/` como prefixo, os objetos de log serão nomeados assim:

```
logs/2013-11-01-21-32-16-E568B2907131C0C0
```

- **Particionamento baseado em data:** se você escolher o particionamento baseado em data, poderá escolher a hora do evento ou a hora de entrega do arquivo de log como a fonte de data usada no formato do log. Esse formato facilita a consulta dos logs.

Se você escolher o particionamento baseado em data, o formato da chave de arquivo de log será exibido da seguinte forma:

```
[DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Por exemplo, se você especificar `logs/` como prefixo-alvo, os objetos de log serão nomeados assim:

```
logs/123456789012/us-west-2/DOC-EXAMPLE-SOURCE-BUCKET/2023/03/01/2023-03-01-21-32-16-E568B2907131C0C0
```

Para a entrega na hora de entrega, a hora nos nomes de arquivo de log corresponde à hora de entrega dos arquivos de log.

Para a entrega na hora do evento, o ano, o mês e o dia correspondem ao dia em que o evento ocorreu, e a hora, os minutos e os segundos são definidos como `00` na chave. Os logs entregues nesses arquivos de log são somente para um dia específico.

Se você estiver configurando os logs por meio da AWS Command Line Interface (AWS CLI), AWS SDKs ou API REST do Amazon S3, use `TargetObjectKeyFormat` para especificar o formato da chave de objeto de log. Para especificar o particionamento não baseado

em data, use `SimplePrefix`. Para especificar o particionamento baseado em data, use `PartitionedPrefix`. Se você usar `PartitionedPrefix`, use `PartitionDateSource` para especificar `EventTime` ou `DeliveryTime`.

Para `SimplePrefix`, o formato da chave de arquivo de log aparece da seguinte forma:

```
[TargetPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Para `PartitionedPrefix` com hora do evento ou hora da entrega, o formato da chave de arquivo de log aparece da seguinte forma:

```
[TargetPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/  
[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Formato da chave de objeto de log

O Amazon S3 usa os seguintes formatos de chave de objeto para os objetos de log carregados no bucket de destino:

- Particionamento não baseado em data: esse é o formato original da chave de objeto de log. Se você escolher esse formato, o formato da chave de arquivo de log será exibido da seguinte forma:

```
[DestinationPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

- Particionamento baseado em data: se você escolher o particionamento baseado em data, poderá escolher a hora do evento ou a hora de entrega do arquivo de log como a fonte de data usada no formato do log. Esse formato facilita a consulta dos logs.

Se você escolher o particionamento baseado em data, o formato da chave de arquivo de log será exibido da seguinte forma:

```
[DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/  
[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Na chave de objeto de log, `YYYY`, `MM`, `DD`, `hh`, `mm` e `ss` são os dígitos do ano, mês, dia, hora, minutos e segundos (respectivamente). 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.

Como os logs são entregues?

O Amazon S3 coleta periodicamente os registros de log de acesso, consolida-os em arquivos de log e faz upload desses arquivos no bucket de destino como objetos de log. Caso o registro em log esteja habilitado em diversos buckets de origem que identifiquem o mesmo bucket de destino, esse bucket de destino terá os 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 do bucket de destino para conceder acesso à entidade principal de serviço do registro em log (`logging.s3.amazonaws.com`) para entrega de logs de acesso. Você também pode conceder acesso para a entrega de logs de acesso ao grupo de entrega de logs do S3 por meio da lista de controle de acesso (ACL) do bucket. No entanto, não é recomendável conceder acesso ao grupo de entrega de logs do S3 usando a ACL do bucket.

Ao habilitar o registro em log de acesso ao servidor e conceder acesso para a entrega de logs de acesso por meio da política do bucket de destino, atualize a política para permitir acesso `s3:PutObject` para a entidade principal de serviço do registro em log. Se você usar o console do Amazon S3 para habilitar o registro em log de acesso ao servidor, o console atualizará automaticamente a política do bucket de destino para conceder essas permissões à entidade principal de serviço do registro em log. Para obter mais informações sobre como conceder permissões para logs de acesso ao servidor, consulte [Permissões para entrega de logs](#).

Note

Os logs de acesso ao servidor não serão entregues ao solicitante ou ao proprietário do bucket para solicitações do endpoint da nuvem privada virtual (VPC) se a política do endpoint da VPC negar essas solicitações.

Configuração aplicada pelo proprietário do bucket em S3 Object Ownership

Se o bucket de destino usar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos, as ACLs serão desabilitadas e não afetarão mais as permissões. É necessário atualizar a política do bucket de destino para conceder acesso à entidade principal de serviço do registro em log. Para obter informações sobre o Object Ownership, consulte [Conceder acesso ao grupo de entrega de logs do S3 para registro em log de acesso ao servidor](#).

Entrega de logs do servidor de melhor esforço

Os registros de log de acesso ao 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. É possível que você até veja uma duplicação de um registro de log. A finalidade dos logs do servidor é proporcionar uma ideia da natureza do tráfego no bucket. Embora a perda ou a duplicação de registros de log sejam acontecimentos raros, esteja ciente de que o registro em log do servidor não tem como objetivo ser uma contabilidade completa de todas as solicitações.

Devido à natureza de melhor esforço do registro em log do servidor, os relatórios de uso podem incluir uma ou mais solicitações de acesso que não aparecem em um log do servidor entregue. Você pode encontrar esses relatórios de uso em Cost & usage reports no console do AWS Billing and Cost Management.

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 registro em log para um bucket, algumas solicitações feitas na hora seguinte poderão ser registradas, enquanto outras não. Suponha que você altere o bucket de destino para registro em log do bucket A para o bucket B. Durante a hora seguinte, alguns logs poderão continuar sendo entregues ao bucket A, 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](#)
- [Formato dos logs de acesso ao servidor do Amazon S3](#)
- [Exclusão de arquivos de log do Amazon S3](#)
- [Usar logs de acesso ao servidor do Amazon S3 para identificar solicitações](#)

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. Essas informações também podem ajudar você a conhecer sua base de clientes e entender a cobrança 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 os logs de acesso de um bucket de origem a um bucket de destino (também conhecido como bucket-alvo) de sua escolha. O bucket de destino deve estar nas mesmas Região da AWS e Conta da AWS que o bucket de origem.

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](#).

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 estimamos as cobranças pela transferência de dados por entrega de arquivo 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, a entrega de

logs para o bucket de origem causará um loop infinito de logs e não é recomendada. Para um gerenciamento de logs mais simples, recomendamos que você salve logs de acesso em um bucket diferente. Para ter mais informações, consulte [Como faço para habilitar a entrega de logs?](#).

- Os buckets do S3 com a funcionalidade Bloqueio de Objetos do S3 habilitada não podem ser usados como buckets de destino para logs de acesso ao servidor. O bucket de destino não pode ter uma configuração de período de retenção padrão.
- O bucket de destino não pode ter a opção Pagamento pelo solicitante habilitada.
- Você poderá usar a [criptografia de bucket padrão](#) no bucket de destino somente se usar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3), que usa o Advanced Encryption Standard de 256 bits (AES-256). A criptografia do lado do servidor padrão com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS) não é compatível.

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.

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. Para a entrega dos logs de acesso, você deve conceder à entidade principal de serviço do registro em log (`logging.s3.amazonaws.com`) acesso ao bucket de destino.

Para conceder permissões ao Amazon S3 para a entrega de logs, você pode usar uma política de bucket ou listas de controle de acesso (ACLs) de bucket, dependendo das configurações da Propriedade de Objetos do S3 do bucket de destino. No entanto, recomendamos usar uma política de bucket em vez de ACLs.

Configuração aplicada pelo proprietário do bucket em S3 Object Ownership

Se o bucket de destino usar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos, as ACLs serão desabilitadas e não afetarão mais as permissões. Nesse caso, é necessário atualizar a política do bucket de destino para conceder acesso à entidade principal de serviço do registro em log. Não é possível atualizar sua ACL de bucket para conceder acesso ao grupo de entrega de logs do S3. Você também não pode incluir concessões de destino (também conhecidas como concessões-alvo) 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 registro em log de acesso ao servidor](#). Para obter informações sobre o Object Ownership, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#). Quando você cria um bucket, as ACLs são desabilitadas por padrão.

Conceder acesso usando uma política de bucket

Para conceder acesso usando uma política de bucket no bucket de destino, atualize a política de bucket para conceder a permissão `s3:PutObject` à entidade principal de serviço do registro em log. Se você usar o console do Amazon S3 para habilitar o registro em log de acesso ao servidor, o console atualizará automaticamente a política do bucket de destino para conceder essa permissão à entidade principal de serviço do registro em log. Se você habilitar o registro em log de acesso ao servidor de maneira programática, poderá atualizar manualmente a política do bucket de destino para conceder acesso à entidade principal de serviço do registro em log.

Para ver um exemplo de política de bucket que concede acesso à entidade principal de serviço do registro em log, consulte [the section called “Conceder permissões à entidade principal do serviço de log usando uma política de bucket”](#).


Conceder acesso usando ACLs de bucket

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. No entanto, não é recomendável conceder acesso ao grupo de entrega de logs do S3 usando ACLs de bucket. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#). 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 registro em log de acesso ao servidor](#). Para ver um exemplo de ACL que concede acesso à entidade principal de serviço do registro em log, consulte [the section called “Conceder permissões ao grupo de entrega de logs usando a ACL de bucket”](#).

Conceder permissões à entidade principal do serviço de log usando uma política de bucket

Este exemplo de política de bucket concede a permissão `s3:PutObject` à entidade principal de serviço do registro em log (`logging.s3.amazonaws.com`). Para usar essa política de bucket, substitua os *user input placeholders* por suas próprias informações. Na política a seguir, ***DOC-EXAMPLE-DESTINATION-BUCKET*** é o bucket de destino em que os logs de acesso ao servidor

serão entregues e *DOC-EXAMPLE-SOURCE-BUCKET* é o bucket de origem. *EXAMPLE-LOGGING-PREFIX* é o prefixo de destino opcional (também conhecido como prefixo-alvo) que você deseja usar para os objetos de log. *SOURCE-ACCOUNT-ID* é a Conta da AWS à qual o bucket de origem pertence.

 Note

Se houver declarações Deny na política de bucket, verifique se elas não estão impedindo que o Amazon S3 entregue logs de acesso.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3ServerAccessLogsPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "logging.s3.amazonaws.com"
      },
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET/EXAMPLE-LOGGING-PREFIX",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET"
        },
        "StringEquals": {
          "aws:SourceAccount": "SOURCE-ACCOUNT-ID"
        }
      }
    }
  ]
}
```

Conceder permissões ao grupo de entrega de logs usando a ACL de bucket

Note

Com base em uma prática recomendada de segurança, o Amazon S3 desabilita as listas de controle de acesso (ACLs) por padrão em todos os novos buckets. Para obter mais informações sobre permissões de ACL no console do Amazon S3, consulte [Configurar ACLs](#).

Embora esse método não seja recomendado, é possível conceder permissões ao grupo de entrega de logs usando uma ACL de bucket. Porém, se o bucket de destino usar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos, você não poderá definir ACLs de bucket ou de objeto. Você também não pode incluir concessões de destino (também conhecidas como concessões-alvo) na configuração [PutBucketLogging](#). Em vez disso, é necessário usar uma política de bucket para conceder acesso à entidade principal do serviço de log (`logging.s3.amazonaws.com`). Para ter mais informações, consulte [Permissões para entrega de logs](#).

Na ACL de bucket, o grupo de entrega de logs é representado pelo seguinte URL:

```
http://acs.amazonaws.com/groups/s3/LogDelivery
```

Para conceder as permissões WRITE e READ_ACP (leitura de 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](#).

⚠ 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 modelo do CloudFormation. Fazer isso é importante porque só é possível conceder essas permissões criando uma ACL para o bucket, mas não é possível criar ACLs personalizadas para buckets no CloudFormation. Você só pode usar ACLs predefinidas com o CloudFormation.

Para habilitar o log de acesso ao servidor

Para habilitar o registro em log de acesso ao servidor usando o console do Amazon S3, a API REST do Amazon S3, os AWS SDKs e a AWS CLI, use os procedimentos a seguir.

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 Registro em log de acesso ao servidor, selecione Habilitar.
6. Em Bucket de destino, especifique um bucket e um prefixo opcional. Se você especificar um prefixo, recomendamos incluir uma barra (/) depois dele para facilitar a localização dos logs.

ℹ Note

Especificar um prefixo com uma barra (/) simplifica 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, da seguinte forma:

```
logs/2013-11-01-21-32-16-E568B2907131C0C0
```

Se você especificar o valor de prefixo logs, o objeto de log será exibido da seguinte forma:

```
logs2013-11-01-21-32-16-E568B2907131C0C0
```

7. Em Formato da chave do objeto de log, execute uma das seguintes ações:

- Para escolher o particionamento não baseado em data, escolha [DestinationPrefix][AAAA]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString].
- Para escolher o particionamento baseado em data, escolha [DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[AAAA]/[MM]/[DD]/[AAAA]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString], depois escolha Hora do evento do S3 ou Hora de entrega do arquivo de log.

8. Escolha Salvar alterações.

Quando você habilita o registro em log de acesso ao servidor em um bucket, o console habilita o registro em log no bucket de origem e atualiza a política do bucket de destino para conceder a permissão `s3:PutObject` à entidade principal de serviço do registro em log (`logging.s3.amazonaws.com`). Para obter mais informações sobre esta política de bucket, consulte [Conceder permissões à entidade principal do serviço de log usando uma política de bucket](#).

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?](#).

Para ter mais informações, consulte [Visualização das propriedades de um bucket do S3](#).

Uso da API REST

Para habilitar o registro em log, envie uma solicitação [PutBucketLogging](#) para adicionar a configuração de registro em log no bucket de origem. A solicitação especifica o bucket de destino (também conhecido como bucket-alvo) e, opcionalmente, o prefixo a ser usado com todas as chaves de objeto de log.

O exemplo a seguir identifica `DOC-EXAMPLE-DESTINATION-BUCKET` como o bucket de destino e `Logs/` como o prefixo.

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

```
<LoggingEnabled>
  <TargetBucket>DOC-EXAMPLE-DESTINATION-BUCKET</TargetBucket>
  <TargetPrefix>Logs/</TargetPrefix>
</LoggingEnabled>
</BucketLoggingStatus>
```

O exemplo a seguir identifica *DOC-EXAMPLE-DESTINATION-BUCKET* como o bucket de destino, *Logs/* como o prefixo e *EventTime* como o formato da chave de objeto de log.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>DOC-EXAMPLE-DESTINATION-BUCKET</TargetBucket>
    <TargetPrefix>Logs/</TargetPrefix>
    <TargetObjectKeyFormat>
      <PartitionedPrefix>
        <PartitionDateSource>EventTime</PartitionDateSource>
      </PartitionedPrefix>
    </TargetObjectKeyFormat>
  </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 (também conhecidas como concessões-alvo) para conceder permissões a outros usuários para que eles possam acessar os logs. Para ter mais informações, consulte [PutBucketLogging](#).

Note

Se o bucket de destino usar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos, você não poderá usar concessões de destino para conceder permissões a outros usuários. Para conceder permissões a outros, é possível atualizar a política de bucket no bucket de destino. Para ter mais informações, consulte [Permissões para entrega de logs](#).

Para recuperar a configuração de registro em log de um bucket, use a operação de API [GetBucketLogging](#).

Para excluir a configuração de registro em log, envie uma solicitação `PutBucketLogging` com um `BucketLoggingStatus` vazio:

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

Para habilitar o registro em log em um bucket, é possível usar a API do Amazon S3 ou as bibliotecas wrapper do AWS SDK.

Uso dos AWS SDKs

O exemplo a seguir habilita o registro em log em um bucket. Você deve criar dois buckets, um de origem e um de destino (alvo). Os exemplos atualizam primeiro a ACL do bucket de destino. Depois, eles concedem ao grupo de entrega de logs as permissões necessárias para gravar logs no bucket de destino e habilitam o registro em log no bucket de origem.

Este exemplo não funcionará em buckets de destino que usam a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos.

Se o bucket de destino (alvo) usar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos, você não poderá definir ACLs de bucket ou de objeto. Também não é possível incluir concessões de destino (alvo) 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 ter mais informações, consulte [Permissões para entrega de logs](#).

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
using System;
using System.IO;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;
using Microsoft.Extensions.Configuration;
```

```
/// <summary>
/// This example shows how to enable logging on an Amazon Simple Storage
/// Service (Amazon S3) bucket. You need to have two Amazon S3 buckets for
/// this example. The first is the bucket for which you wish to enable
/// logging, and the second is the location where you want to store the
/// logs.
/// </summary>
public class ServerAccessLogging
{
    private static IConfiguration _configuration = null!;

    public static async Task Main()
    {
        LoadConfig();

        string bucketName = _configuration["BucketName"];
        string logBucketName = _configuration["LogBucketName"];
        string logObjectKeyPrefix = _configuration["LogObjectKeyPrefix"];
        string accountId = _configuration["AccountId"];

        // If the AWS Region defined for your default user is different
        // from the Region where your Amazon S3 bucket is located,
        // pass the Region name to the Amazon S3 client object's constructor.
        // For example: RegionEndpoint.USWest2 or RegionEndpoint.USEast2.
        IAmazonS3 client = new AmazonS3Client();

        try
        {
            // Update bucket policy for target bucket to allow delivery of
logs to it.
            await SetBucketPolicyToAllowLogDelivery(
                client,
                bucketName,
                logBucketName,
                logObjectKeyPrefix,
                accountId);

            // Enable logging on the source bucket.
            await EnableLoggingAsync(
                client,
                bucketName,
                logBucketName,
                logObjectKeyPrefix);
        }
    }
}
```

```

        catch (AmazonS3Exception e)
        {
            Console.WriteLine($"Error: {e.Message}");
        }
    }

    /// <summary>
    /// This method grants appropriate permissions for logging to the
    /// Amazon S3 bucket where the logs will be stored.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client which will be
used
    /// to apply the bucket policy.</param>
    /// <param name="sourceBucketName">The name of the source bucket.</param>
    /// <param name="logBucketName">The name of the bucket where logging
    /// information will be stored.</param>
    /// <param name="logPrefix">The logging prefix where the logs should be
delivered.</param>
    /// <param name="accountId">The account id of the account where the
source bucket exists.</param>
    /// <returns>Async task.</returns>
    public static async Task SetBucketPolicyToAllowLogDelivery(
        IAmazonS3 client,
        string sourceBucketName,
        string logBucketName,
        string logPrefix,
        string accountId)
    {
        var resourceArn = @$"arn:aws:s3:::" + logBucketName + "/" +
logPrefix + @$""";

        var newPolicy = @"{
            ""Statement"": [{
                ""Sid"": ""S3ServerAccessLogsPolicy"",
                ""Effect"": ""Allow"",
                ""Principal"": { ""Service"":
""logging.s3.amazonaws.com"" },
                ""Action"": [""s3:PutObject""],
                ""Resource"": ["" + resourceArn + @""],
                ""Condition"": {
                    ""ArnLike"": { ""aws:SourceArn"":
""arn:aws:s3:::" + sourceBucketName + @"" },
                    ""StringEquals"": { ""aws:SourceAccount"": "" +
accountId + @"" }
                }
            }
        }";
    }
}

```



```

        }
    }
}";
    Console.WriteLine($"The policy to apply to bucket {logBucketName} to
enable logging:");
    Console.WriteLine(newPolicy);

    PutBucketPolicyRequest putRequest = new PutBucketPolicyRequest
    {
        BucketName = logBucketName,
        Policy = newPolicy,
    };
    await client.PutBucketPolicyAsync(putRequest);
    Console.WriteLine("Policy applied.");
}

/// <summary>
/// This method enables logging for an Amazon S3 bucket. Logs will be
stored
/// in the bucket you selected for logging. Selected prefix
/// will be prepended to each log object.
/// </summary>
/// <param name="client">The initialized Amazon S3 client which will be
used
/// to configure and apply logging to the selected Amazon S3 bucket.</
param>
/// <param name="bucketName">The name of the Amazon S3 bucket for which
you
/// wish to enable logging.</param>
/// <param name="logBucketName">The name of the Amazon S3 bucket where
logging
/// information will be stored.</param>
/// <param name="logObjectKeyPrefix">The prefix to prepend to each
/// object key.</param>
/// <returns>Async task.</returns>
public static async Task EnableLoggingAsync(
    IAmazonS3 client,
    string bucketName,
    string logBucketName,
    string logObjectKeyPrefix)
{
    Console.WriteLine($"Enabling logging for bucket {bucketName}.");
    var loggingConfig = new S3BucketLoggingConfig
    {

```

```
        TargetBucketName = logBucketName,
        TargetPrefix = logObjectKeyPrefix,
    };

    var putBucketLoggingRequest = new PutBucketLoggingRequest
    {
        BucketName = bucketName,
        LoggingConfig = loggingConfig,
    };
    await client.PutBucketLoggingAsync(putBucketLoggingRequest);
    Console.WriteLine($"Logging enabled.");
}

/// <summary>
/// Loads configuration from settings files.
/// </summary>
public static void LoadConfig()
{
    _configuration = new ConfigurationBuilder()
        .SetBasePath(Directory.GetCurrentDirectory())
        .AddJsonFile("settings.json") // Load settings from .json file.
        .AddJsonFile("settings.local.json", true) // Optionally, load
local settings.
        .Build();
}
}
```

- Para obter detalhes da API, consulte [PutBucketLogging](#) na Referência da API AWS SDK for .NET.

Java

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.BucketLoggingStatus;
import software.amazon.awssdk.services.s3.model.LoggingEnabled;
import software.amazon.awssdk.services.s3.model.PartitionedPrefix;
import software.amazon.awssdk.services.s3.model.PutBucketLoggingRequest;
import software.amazon.awssdk.services.s3.model.TargetObjectKeyFormat;
```

```
// Class to set a bucket policy on a target S3 bucket and enable server access
logging on a source S3 bucket.
public class ServerAccessLogging {
    private static S3Client s3Client;

    public static void main(String[] args) {
        String sourceBucketName = "SOURCE-BUCKET";
        String targetBucketName = "TARGET-BUCKET";
        String sourceAccountId = "123456789012";
        String targetPrefix = "logs/";

        // Create S3 Client.
        s3Client = S3Client.builder().
            region(Region.US_EAST_2)
            .build();

        // Set a bucket policy on the target S3 bucket to enable server access
logging by granting the
        // logging.s3.amazonaws.com principal permission to use the PutObject
operation.
        ServerAccessLogging serverAccessLogging = new ServerAccessLogging();
        serverAccessLogging.setTargetBucketPolicy(sourceAccountId, sourceBucketName,
targetBucketName);

        // Enable server access logging on the source S3 bucket.
        serverAccessLogging.enableServerAccessLogging(sourceBucketName,
targetBucketName,
            targetPrefix);

    }

    // Function to set a bucket policy on the target S3 bucket to enable server
access logging by granting the
    // logging.s3.amazonaws.com principal permission to use the PutObject operation.
    public void setTargetBucketPolicy(String sourceAccountId, String
sourceBucketName, String targetBucketName) {
        String policy = "{\n" +
            "    \"Version\": \"2012-10-17\",\n" +
            "    \"Statement\": [\n" +
            "        {\n" +
            "            \"Sid\": \"S3ServerAccessLogsPolicy\",\n" +
            "            \"Effect\": \"Allow\",\n" +
            "            \"Principal\": {\"Service\": \"logging.s3.amazonaws.com
\n\"},\n" +
```

```

        "        \"Action\": [\n" +
        "            \"s3:PutObject\"\n" +
        "        ],\n" +
        "        \"Resource\": \"arn:aws:s3:::\" + targetBucketName + "/*
\", \n" +
        "        \"Condition\": {\n" +
        "            \"ArnLike\": {\n" +
        "                \"aws:SourceArn\": \"arn:aws:s3:::\" +
sourceBucketName + "\"\n" +
        "            },\n" +
        "            \"StringEquals\": {\n" +
        "                \"aws:SourceAccount\": \"\" + sourceAccountId +
"\" \n" +
        "            }\n" +
        "        }\n" +
        "    ]\n" +
        "};
    s3Client.putBucketPolicy(b -> b.bucket(targetBucketName).policy(policy));
}

// Function to enable server access logging on the source S3 bucket.
public void enableServerAccessLogging(String sourceBucketName, String
targetBucketName,
    String targetPrefix) {
    TargetObjectKeyFormat targetObjectKeyFormat =
TargetObjectKeyFormat.builder()

.partitionedPrefix(PartitionedPrefix.builder().partitionDataSource("EventTime").build())
    .build();
    LoggingEnabled loggingEnabled = LoggingEnabled.builder()
        .targetBucket(targetBucketName)
        .targetPrefix(targetPrefix)
        .targetObjectKeyFormat(targetObjectKeyFormat)
        .build();
    BucketLoggingStatus bucketLoggingStatus = BucketLoggingStatus.builder()
        .loggingEnabled(loggingEnabled)
        .build();
    s3Client.putBucketLogging(PutBucketLoggingRequest.builder()
        .bucket(sourceBucketName)
        .bucketLoggingStatus(bucketLoggingStatus)
        .build());
}

```

```
}
```

Usando a 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 os logs de acesso do Amazon S3 sejam entregues 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 (alvo) usar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos, você não poderá definir ACLs de bucket ou de objeto. Também não é possível incluir concessões de destino (alvo) 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](#).

Example : Habilite logs de acesso com cinco buckets em duas regiões

Neste exemplo, você tem estes cinco buckets:

- 1-DOC-EXAMPLE-BUCKET1-us-east-1
- 2-DOC-EXAMPLE-BUCKET1-us-east-1
- 3-DOC-EXAMPLE-BUCKET1-us-east-1
- 1-DOC-EXAMPLE-BUCKET1-us-west-2
- 2-DOC-EXAMPLE-BUCKET1-us-west-2

Note

A etapa final do procedimento a seguir fornece exemplos de scripts bash que você pode usar para criar buckets de registro em log e habilitar o registro em log de acesso ao servidor nesses buckets. Para usar esses scripts, você deve criar os arquivos `policy.json` e `logging.json`, conforme descrito no procedimento a seguir.

1. Crie dois buckets de destino do registro em log nas regiões Oeste dos EUA (Oregon) e Leste dos EUA (N. da Virgínia) e dê a eles os seguintes nomes:

- DOC-EXAMPLE-BUCKET1-logs-us-east-1
 - DOC-EXAMPLE-BUCKET1-logs-us-west-2
2. Posteriormente nessas etapas, você habilitará o registro em log de acesso ao servidor da seguinte forma:
- Logs de 1-DOC-EXAMPLE-BUCKET1-us-east-1 no bucket do S3 DOC-EXAMPLE-BUCKET1-logs-us-east-1 com o prefixo 1-DOC-EXAMPLE-BUCKET1-us-east-1
 - Logs de 2-DOC-EXAMPLE-BUCKET1-us-east-1 no bucket do S3 DOC-EXAMPLE-BUCKET1-logs-us-east-1 com o prefixo 2-DOC-EXAMPLE-BUCKET1-us-east-1
 - Logs de 3-DOC-EXAMPLE-BUCKET1-us-east-1 no bucket do S3 DOC-EXAMPLE-BUCKET1-logs-us-east-1 com o prefixo 3-DOC-EXAMPLE-BUCKET1-us-east-1
 - Logs de 1-DOC-EXAMPLE-BUCKET1-us-west-2 no bucket do S3 DOC-EXAMPLE-BUCKET1-logs-us-west-2 com o prefixo 1-DOC-EXAMPLE-BUCKET1-us-west-2
 - Logs de 2-DOC-EXAMPLE-BUCKET1-us-west-2 no bucket do S3 DOC-EXAMPLE-BUCKET1-logs-us-west-2 com o prefixo 2-DOC-EXAMPLE-BUCKET1-us-west-2
3. Para cada bucket de registro em log de destino, conceda permissões para a entrega de logs de acesso ao servidor usando uma ACL de bucket ou uma política de bucket:
- Atualizar a política de bucket (recomendado): para conceder permissões à entidade principal do serviço de log, use o comando `put-bucket-policy` a seguir. Substitua *DOC-EXAMPLE-DESTINATION-BUCKET-Logs* pelo nome de seu bucket de destino.

```
aws s3api put-bucket-policy --bucket DOC-EXAMPLE-DESTINATION-BUCKET-Logs --policy file://policy.json
```

`Policy.json` é um documento JSON na pasta atual que contém a política de bucket a seguir. Para usar essa política de bucket, substitua os *user input placeholders* por suas próprias informações. Na política a seguir, *DOC-EXAMPLE-DESTINATION-BUCKET-Logs* é o bucket de destino em que os logs de acesso ao servidor serão entregues e *DOC-EXAMPLE-SOURCE-BUCKET* é o bucket de origem. *SOURCE-ACCOUNT-ID* é a Conta da AWS à qual o bucket de origem pertence.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```

    {
      "Sid": "S3ServerAccessLogsPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "logging.s3.amazonaws.com"
      },
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET-Logs/*",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET"
        },
        "StringEquals": {
          "aws:SourceAccount": "SOURCE-ACCOUNT-ID"
        }
      }
    }
  ]
}

```

- Atualizar a ACL de bucket: para conceder permissões ao grupo de entrega de logs do S3, use o comando `put-bucket-acl` a seguir. Substitua *DOC-EXAMPLE-DESTINATION-BUCKET-Logs* pelo nome do bucket de destino (alvo).

```

aws s3api put-bucket-acl --bucket DOC-EXAMPLE-DESTINATION-BUCKET-Logs --grant-write URI=http://acs.amazonaws.com/groups/s3/LogDelivery --grant-read-acp URI=http://acs.amazonaws.com/groups/s3/LogDelivery

```

4. Depois, crie um arquivo `logging.json` que contenha a configuração de registro em log (com base em um dos três exemplos a seguir). Depois de criar o arquivo `logging.json`, você pode aplicar a configuração de registro em log usando o comando `put-bucket-logging` a seguir. Substitua *DOC-EXAMPLE-DESTINATION-BUCKET-Logs* pelo nome do bucket de destino (alvo).

```

aws s3api put-bucket-logging --bucket DOC-EXAMPLE-DESTINATION-BUCKET-Logs --bucket-logging-status file://logging.json

```

Note

Em vez de usar esse comando `put-bucket-logging` para aplicar a configuração de registro em log em cada bucket de destino, você pode usar um dos scripts bash fornecidos na próxima etapa. Para usar esses scripts, você deve criar os arquivos `policy.json` e `logging.json`, conforme descrito nesse procedimento.

O arquivo `logging.json` é um documento JSON na pasta atual que contém a configuração do registro em log. Se um bucket de destino usar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos, a configuração de registro em log não poderá conter concessões de destino (alvo). Para ter mais informações, consulte [Permissões para entrega de logs](#).

Example : **logging.json** sem concessões de destino (alvo)

O exemplo de arquivo `logging.json` a seguir não contém concessões de destino (alvo). Portanto, você pode aplicar essa configuração a um bucket de destino (alvo) que usa a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos.

```
{
  "LoggingEnabled": {
    "TargetBucket": "DOC-EXAMPLE-DESTINATION-BUCKET-logs",
    "TargetPrefix": "DOC-EXAMPLE-DESTINATION-BUCKET/"
  }
}
```

Example : **logging.json** com concessões de destino (alvo)

O exemplo de arquivo `logging.json` a seguir contém concessões de destino (alvo).

Se o bucket de destino usar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos, você não poderá incluir concessões de destino (alvo) na configuração [PutBucketLogging](#). Para ter mais informações, consulte [Permissões para entrega de logs](#).

```
{
```



```

    "LoggingEnabled": {
      "TargetBucket": "DOC-EXAMPLE-DESTINATION-BUCKET-logs",
      "TargetPrefix": "DOC-EXAMPLE-DESTINATION-BUCKET/",
      "TargetGrants": [
        {
          "Grantee": {
            "Type": "AmazonCustomerByEmail",
            "EmailAddress": "user@example.com"
          },
          "Permission": "FULL_CONTROL"
        }
      ]
    }
  }
}

```

Example : **logging.json** com o formato da chave de objeto de log definido como a hora do evento do S3

O arquivo `logging.json` a seguir altera o formato da chave de objeto de log para a hora do evento do S3. Para obter mais informações sobre como definir o formato da chave de objeto de log, consulte [the section called “Como faço para habilitar a entrega de logs?”](#).

```

{
  "LoggingEnabled": {
    "TargetBucket": "DOC-EXAMPLE-DESTINATION-BUCKET-logs",
    "TargetPrefix": "DOC-EXAMPLE-DESTINATION-BUCKET/",
    "TargetObjectKeyFormat": {
      "PartitionedPrefix": {
        "PartitionDateSource": "EventTime"
      }
    }
  }
}

```

5. Use um script bash a seguir para adicionar o registro em log de acesso a todos os buckets em sua conta. Substitua *DOC-EXAMPLE-DESTINATION-BUCKET-logs* pelo nome do bucket de destino (alvo) e substitua *us-west-2* pelo nome da região em que os buckets estão localizados.

Note

Esse script só 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='DOC-EXAMPLE-DESTINATION-BUCKET-Logs'
region='us-west-2'

# Create the logging bucket.
aws s3 mb s3://$loggingBucket --region $region

aws s3api put-bucket-policy --bucket $loggingBucket --policy file://policy.json

# List the buckets in this account.
buckets="$(aws s3 ls | awk '{print $3}')"

# Put a bucket logging configuration on each bucket.
for bucket in $buckets
do
    # This if statement excludes the logging bucket.
    if [ "$bucket" != "$loggingBucket" ] ; then
        continue;
    fi
    printf '{
        "LoggingEnabled": {
            "TargetBucket": "%s",
            "TargetPrefix": "%s/"
        }
    }' "$loggingBucket" "$bucket" > logging.json
    aws s3api put-bucket-logging --bucket $bucket --bucket-logging-status file://
logging.json
    echo "$bucket done"
done

rm logging.json
```

```
echo "Complete"
```

Example – Conceda acesso com ACLs de bucket e adicione log para os buckets de sua conta

```
loggingBucket='DOC-EXAMPLE-DESTINATION-BUCKET-logs'
region='us-west-2'

# Create the logging bucket.
aws s3 mb s3://$loggingBucket --region $region

aws s3api put-bucket-acl --bucket $loggingBucket --grant-write URI=http://
acs.amazonaws.com/groups/s3/LogDelivery --grant-read-acp URI=http://
acs.amazonaws.com/groups/s3/LogDelivery

# List the buckets in this account.
buckets="$(aws s3 ls | awk '{print $3}')"

# Put a bucket logging configuration on each bucket.
for bucket in $buckets
do
    # This if statement excludes the logging bucket.
    if [ "$bucket" != "$loggingBucket" ] ; then
        continue;
    fi
    printf '{
        "LoggingEnabled": {
            "TargetBucket": "%s",
            "TargetPrefix": "%s/"
        }
    }' "$loggingBucket" "$bucket" > logging.json
    aws s3api put-bucket-logging --bucket $bucket --bucket-logging-status file://
logging.json
    echo "$bucket done"
done

rm logging.json

echo "Complete"
```

Verificar a configuração dos logs de acesso ao servidor

Depois de habilitar o registro em log de acesso ao servidor, execute as seguintes etapas:

- Acesse o bucket de destino e verifique se os arquivos de log estão sendo entregues. Depois de configurar os logs de acesso, pode levar mais de uma hora para que todas as solicitações sejam registradas e entregues adequadamente. Também é possível verificar automaticamente a entrega de logs usando métricas de solicitação do Amazon S3 e configurando alarmes do Amazon CloudWatch para essas métricas. Para ter mais informações, consulte [Monitoramento de métricas com o Amazon CloudWatch](#).
- Verifique se você consegue abrir e ler o conteúdo dos arquivos de log.

Para obter informações sobre solução de problemas do registro de acesso ao servidor, consulte [Solução de problemas do registro em log de acessos ao servidor](#).

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 as seguintes finalidades:

- Realizar auditorias de segurança e acesso
- Aprender sobre sua base de clientes
- 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
DOC-EXAMPLE-BUCKET1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 3E57427F3EXAMPLE
REST.GET.VERSIONING - "GET /DOC-EXAMPLE-BUCKET1?versioning HTTP/1.1" 200 - 113 - 7 -
"- "S3Console/0.4" - s91zHYrFp76ZVxRcpX9+5cjAnEH2R0uNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/
```

```
XV/VLi31234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader DOC-EXAMPLE-BUCKET1.s3.us-
west-1.amazonaws.com TLSV1.2 arn:aws:s3:us-west-1:123456789012:accesspoint/example-AP
Yes
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
DOC-EXAMPLE-BUCKET1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 891CE47D2EXAMPLE
REST.GET.LOGGING_STATUS - "GET /DOC-EXAMPLE-BUCKET1?logging HTTP/1.1" 200 -
242 - 11 - "-" "S3Console/0.4" - 9vKBE6vMhrNiWHZmb2L0mX0cqPgZQ0I5XLnCtZNPxev+Hf
+7tpT6sxDwDty4LHBUOZJG96N1234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader DOC-
EXAMPLE-BUCKET1.s3.us-west-1.amazonaws.com TLSV1.2 - -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
DOC-EXAMPLE-BUCKET1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be A1206F460EXAMPLE
REST.GET.BUCKETPOLICY - "GET /DOC-EXAMPLE-BUCKET1?policy HTTP/1.1" 404
NoSuchBucketPolicy 297 - 38 - "-" "S3Console/0.4" - BNaBsXZQQDbssi6xMBdBU2sLt
+Yf5kZDmeBUP35sFoKa3sLLeM78iwEIWxs99CRUrbS4n11234= SigV4 ECDHE-RSA-AES128-GCM-SHA256
AuthHeader DOC-EXAMPLE-BUCKET1.s3.us-west-1.amazonaws.com TLSV1.2 - Yes
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
DOC-EXAMPLE-BUCKET1 [06/Feb/2019:00:01:00 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 7B4A0FABBEXAMPLE
REST.GET.VERSIONING - "GET /DOC-EXAMPLE-BUCKET1?versioning HTTP/1.1" 200 -
113 - 33 - "-" "S3Console/0.4" - Ke1bUcazaN1jWuU1PJaxF64cQVpUEhoZKEG/hmy/gijN/
I1DeWqDfFvnpybFeseEME/u7ME1234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader DOC-
EXAMPLE-BUCKET1.s3.us-west-1.amazonaws.com TLSV1.2 - -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
DOC-EXAMPLE-BUCKET1 [06/Feb/2019:00:01:57 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
DD6CC733AEXAMPLE REST.PUT.OBJECT s3-dg.pdf "PUT /DOC-EXAMPLE-BUCKET1/
s3-dg.pdf HTTP/1.1" 200 - - 4406583 41754 28 "-" "S3Console/0.4" -
10S62Zv81kBW7BB6SX4XJ48o6kpc16LPwEoizZQQxJd5qDSCCTLX0TgS37kYUBKQW3+bPdrG1234= SigV4
ECDHE-RSA-AES128-SHA AuthHeader DOC-EXAMPLE-BUCKET1.s3.us-west-1.amazonaws.com TLSV1.2
- Yes
```

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](#)
- [Registro em log adicional para operações de cópia](#)

- [Informações personalizadas do log de acesso](#)
- [Considerações de programação para o formato do log de acesso ao servidor extensível](#)

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 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

```
DOC-EXAMPLE-BUCKET1
```

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 IP aparente do solicitante. Os proxies e os firewalls intermediários podem obscurecer o endereço IP 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 retornará o nome do usuário do IAM do solicitante com o Usuário raiz da conta da AWS ao 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](#).

Exemplo de registro

```
REST.PUT.OBJECT
```

Chave

A parte da chave (nome do objeto) da solicitação.

Exemplo de registro

```
/photos/2019/08/puppy.jpg
```

Request-URI

A parte Request-URI da mensagem de solicitação HTTP.

Entrada de exemplo

```
"GET /DOC-EXAMPLE-BUCKET1/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 [Código de erro](#) do Amazon S3, 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 (ms) em que a solicitação esteve em trânsito 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 `Referer` 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.example.com/webservices"
```

User-Agent

O valor do cabeçalho do `User-Agent` 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

```
3HL4kqtJvjVBH40Nıjfkđ
```

ID do host

O `x-amz-id-2` ou o ID de solicitação estendida do Amazon S3.

Exemplo de registro

```
s91zHYıFp76ZVxRcpX9+5cjAnEH2R0uNkd2BHfIa6UkFVdtjf5mKR3/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 cifra de Secure Sockets Layer (SSL) que foi negociada para uma solicitação HTTPS, ou um - 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 (URL pré-assinado) 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 herdados. Você poderá ver esses endpoints nos logs de acesso ao servidor ou nos logs do AWS CloudTrail. Para ter mais informações, consulte [Endpoints herdados](#). Para obter uma lista completa de regiões e endpoints do Amazon S3, consulte [Endpoints e cotas do Amazon S3](#) na 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: TLSv1.1, TLSv1.2, TLSv1.3 ou - se o TLS não foi usado.

Exemplo de registro

```
TLSv1.2
```

ARN 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 malformatado ou se não for usado, o campo apresentará um -. Para obter mais informações sobre pontos de acesso, consulte [Usar pontos de acesso](#). Para obter mais informações sobre ARNs, consulte [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
```

aclRequired

Uma string que indica se a solicitação exigiu uma lista de controle de acesso (ACL) para autorização. Se a solicitação exigiu uma ACL para autorização, a string é Yes. Se nenhuma ACL foi necessária, a string é -. Para obter mais informações sobre ACLs, consulte [Visão geral da lista de controle de acesso \(ACL\)](#). Para obter mais informações sobre como usar o campo

ac1Required para desativar ACLs, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

Exemplo de registro

```
Yes
```

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 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

```
DOC-EXAMPLE-BUCKET1
```

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 IP aparente do solicitante. Os proxies e os firewalls intermediários podem obscurecer o endereço IP 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 retornará o nome do usuário do IAM do solicitante junto com o Usuário raiz da conta da AWS ao 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 (nome do objeto) 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 Request-URI da mensagem de solicitação HTTP.

Exemplo de registro

```
"GET /DOC-EXAMPLE-BUCKET1/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](#) 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 (ms) em que a solicitação esteve em trânsito 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 `Referer` 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.example.com/webservices"
```

User-Agent

O valor do cabeçalho do User-Agent 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 tiver especificado um parâmetro `versionId` como parte da fonte da cópia.

Entrada de exemplo

```
3HL4kqtJvjVBH40Nrfkd
```

ID do host

O `x-amz-id-2` ou o ID de solicitação estendida do Amazon S3.

Exemplo de registro

```
s91zHYrFp76ZVxRcpX9+5cjAnEH2R0uNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Versão do Signature

A versão de assinatura, `SigV2` ou `SigV4`, que foi usada para autenticar a solicitação, ou um - para solicitações não autenticadas.

Exemplo de registro

```
SigV4
```

Pacote de criptografia

A cifra de Secure Sockets Layer (SSL) que foi negociada para uma solicitação HTTPS, ou um - 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 strings de consulta (URL pré-assinados) ou um - para solicitações não autenticadas.

Exemplo de registro

```
AuthHeader
```

Cabeçalho de host

O endpoint que foi usado para conectar ao Amazon S3.

Exemplo de registro

```
s3.us-west-2.amazonaws.com
```

Algumas regiões mais antigas oferecem suporte a endpoints herdados. Você poderá ver esses endpoints nos logs de acesso ao servidor ou nos logs do AWS CloudTrail. Para ter mais informações, consulte [Endpoints herdados](#). Para obter uma lista completa de regiões e endpoints do Amazon S3, consulte [Endpoints e cotas do Amazon S3](#) na 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: TLSv1.1, TLSv1.2, TLSv1.3 ou - se o TLS não foi usado.

Exemplo de registro

```
TLSv1.2
```

ARN 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 malformatado ou se não for usado, o campo apresentará um -. Para obter mais informações sobre pontos de acesso, consulte [Usar pontos de acesso](#). Para obter mais informações sobre ARNs, consulte [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
```

aclRequired

Uma string que indica se a solicitação exigiu uma lista de controle de acesso (ACL) para autorização. Se a solicitação exigiu uma ACL para autorização, a string é Yes. Se nenhuma ACL foi necessária, a string é -. Para obter mais informações sobre ACLs, consulte [Visão geral da lista de controle de acesso \(ACL\)](#). Para obter mais informações sobre como usar o campo aclRequired para desativar ACLs, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

Exemplo de registro

```
Yes
```

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/DOC-EXAMPLE-BUCKET1/photos/2019/08/puppy.jpg?x-user=johndoe" funciona da mesma forma que a solicitação para "s3.amazonaws.com/DOC-EXAMPLE-BUCKET1/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, certifique-se de que qualquer código que analise logs de acesso ao servidor consiga lidar com os campos finais que talvez não entenda.

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. Use a configuração da funcionalidade Ciclo de Vida do Amazon S3 para definir regras de modo que o Amazon S3 organize esses objetos automaticamente para exclusão ao final de suas vidas.

É possível definir uma configuração de ciclo de vida para um subconjunto de objetos no bucket do S3 usando um prefixo compartilhado. 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, suponha que os objetos de log tenham 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](#).

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](#).

Usar logs de acesso ao servidor do Amazon S3 para identificar solicitações

Você pode identificar solicitações do Amazon S3 usando os logs de acesso ao servidor do Amazon S3.

Note

- Para identificar as solicitações do Amazon S3, recomendamos que você use eventos de dados do AWS CloudTrail em vez de logs de acesso ao servidor do Amazon S3. Os eventos de dados do CloudTrail são mais fáceis de configurar e contêm mais informações. Para ter mais informações, consulte [Identificar solicitações do Amazon S3 usando o CloudTrail](#).
- Dependendo de quantas solicitações de acesso você receber, a análise dos logs poderá exigir mais recursos ou tempo do que o uso dos eventos de dados do CloudTrail.

Tópicos

- [Consultar logs de acesso para solicitações usando o Amazon Athena](#)
- [Identificar solicitações do Signature Version 2 usando logs de acesso do Amazon S3](#)
- [Identificar solicitações de acesso a objetos usando logs de acesso do Amazon S3](#)

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. Substitua os *user input placeholders* utilizados nos exemplos a seguir por suas próprias informações.

Note

Para especificar um local do Amazon S3 em uma consulta do Athena, é necessário fornecer um URI do S3 para o bucket ao qual os logs são entregues. Esse URI deve incluir o nome e o prefixo do bucket no seguinte formato: `s3://DOC-EXAMPLE-BUCKET1-logs/prefix/`

1. Abra o console do Athena em <https://console.aws.amazon.com/athena/>.
2. No Editor de consultas, execute um comando semelhante ao seguinte. Substitua `s3_access_logs_db` pelo nome que você deseja atribuir ao banco de dados.

```
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. Substitua *s3_access_logs_db.mybucket_logs* pelo nome que você deseja atribuir à tabela. 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,  
  `bytessent` BIGINT,  
  `objectsize` BIGINT,  
  `totaltime` STRING,  
  `turnaroundtime` STRING,  
  `referrer` STRING,  
  `useragent` STRING,  
  `versionid` STRING,  
  `hostid` STRING,  
  `sigv` STRING,  
  `ciphersuite` STRING,  
  `authtype` STRING,  
  `endpoint` STRING,  
  `tlsversion` STRING,  
  `accesspointarn` STRING,  
  `aclrequired` STRING)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.serde2.RegexSerDe'
```

```

WITH SERDEPROPERTIES (
  'input.regex'='([ ]*) ([ ]*) \\[(.?)\\] ([ ]*) ([ ]*) ([ ]*) ([ ]*)
([ ]*) (\\"[^\\"]*"|\\'| -) (-|[0-9]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*)
(\\"[^\\"]*"|\\'| -) ([ ]*)(?: ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*)
([ ]*))?.*$'
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET1-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 determinado período

```
SELECT coalesce(SUM(bytesent), 0) AS bytesenttotal
FROM s3_access_logs_db.mybucket_logs
WHERE remoteip='192.0.2.1'
      AND parse_datetime(requestdatetime, 'dd/MMM/yyyy:HH:mm:ss Z')
      BETWEEN parse_datetime('2022-06-01', 'yyyy-MM-dd')
      AND parse_datetime('2022-07-01', 'yyyy-MM-dd');
```

Note

Para reduzir o tempo de retenção dos logs, você pode criar uma configuração da funcionalidade Ciclo de Vida do S3 para o bucket de logs de acesso ao servidor. Crie regras de configuração 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 ter mais informações, consulte [Definir uma configuração de ciclo de vida em um bucket](#).

Identificar solicitações do Signature Version 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 Version 2 usando os logs de acesso do Amazon S3.

Note

Para identificar as solicitações do Signature Version 2, recomendamos que você use eventos de dados do AWS CloudTrail em vez de logs de acesso ao servidor do Amazon S3. Os eventos de dados do CloudTrail são mais fáceis de configurar e contêm mais informações

do que os logs de acesso ao servidor. Para ter mais informações, consulte [Identificar solicitações de Signature Version 2 do Amazon S3 usando o CloudTrail](#).

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 a objetos 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 PUT de objetos para o Amazon S3 de um log de acesso ao servidor.

Example : Mostrar todos os solicitantes que estão enviando solicitações **PUT** de objetos em determinado período

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
FROM s3_access_logs_db
WHERE operation='REST.PUT.OBJECT' AND
parse_datetime(requestdatetime, 'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
```

O exemplo de consulta do Amazon Athena a seguir mostra como obter todas as solicitações GET de objetos para o Amazon S3 do log de acesso ao servidor.

Example : Mostrar todos os solicitantes que estão enviando solicitações **GET** de objetos em determinado período

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
```



```
FROM s3_access_logs_db
WHERE operation='REST.GET.OBJECT' AND
parse_datetime(requestdatetime, 'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
```

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 determinado período

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
FROM s3_access_logs_db.mybucket_logs
WHERE requester IS NULL AND
parse_datetime(requestdatetime, 'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
```

A consulta do Amazon Athena a seguir mostra como identificar todas as solicitações feitas aos seus buckets do S3 que exigiram uma lista de controle de acesso (ACL) para autorização. Você pode usar essas informações para migrar essas permissões de ACL para as políticas de bucket apropriadas e desativar as ACLs. Depois de criar essas políticas de bucket, você pode desativar as ACLs para esses buckets. Para obter mais informações sobre como desativar as ACLs, consulte [Pré-requisitos para desabilitar ACLs](#).

Example : Identifique todas as solicitações que exigiram uma ACL para autorização

```
SELECT bucket_name, requester, key, operation, aclrequired, requestdatetime
FROM s3_access_logs_db
WHERE aclrequired = 'Yes' AND
parse_datetime(requestdatetime, 'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2022-05-10:00:00:00', 'yyyy-MM-dd:HH:mm:ss')
AND parse_datetime('2022-08-10:00:00:00', 'yyyy-MM-dd:HH:mm:ss')
```

Note

- 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 ou endereços IP inesperados ou não autorizados e identificar solicitações anônimas aos seus buckets.
- Essa consulta recupera somente informações do momento no qual o registro estava habilitado.
- Se você estiver usando logs do AWS CloudTrail, consulte [Identificar o acesso a objetos do S3 usando o CloudTrail](#).

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](#).

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](#).
- 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](#).
- 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](#).

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 da AWS de destino e o número total de operações com replicação falhada. 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 [Monitorar o progresso com métricas de replicação e Notificações de eventos do S3](#) ou [Atendimento aos requisitos de conformidade usando o Controle do tempo de replicação do S3 \(S3 RTC\)](#).

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](#).

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 sobre o CloudWatch, consulte [O que é o Amazon CloudWatch?](#) no Guia do usuário do Amazon CloudWatch. Você pode precisar de algumas configurações adicionais para os alarmes do CloudWatch, dependendo dos seus casos de uso. Por exemplo, você pode usar a expressão matemática de métricas para criar um alarme. Para obter mais informações, consulte [Usar métricas do CloudWatch](#), [Usar matemática de métricas](#), [Usar de alarmes do Amazon CloudWatch](#) e [Criar um alarme do CloudWatch com base em uma expressão matemática de métricas](#) no Guia do usuário do Amazon CloudWatch.

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, consulte os tópicos a seguir.

Tópicos

- [Métricas e dimensões](#)
- [Acessar métricas do CloudWatch](#)

- [Configurações de métricas do CloudWatch](#)

Métricas e dimensões

As métricas e dimensões que o Amazon S3 envia para o Amazon CloudWatch estão listadas nas tabelas a seguir.

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.

Tópicos

- [Métricas de armazenamento diárias do Amazon S3 para buckets no CloudWatch](#)
- [Métricas de solicitação do Amazon S3 no CloudWatch](#)
- [Métricas de replicação do S3 no CloudWatch](#)
- [Métricas do S3 Storage Lens no CloudWatch](#)
- [Métricas de solicitação do S3 Object Lambda no CloudWatch](#)
- [Métricas do Amazon S3 on Outposts no CloudWatch](#)
- [Dimensões do Amazon S3 no CloudWatch](#)
- [Dimensões da funcionalidade Replicação do S3 no CloudWatch](#)
- [Dimensões do S3 Storage Lens no CloudWatch](#)
- [Dimensões de solicitação do S3 Object Lambda no CloudWatch](#)

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
BucketSizeBytes	<p>A quantidade de dados, em bytes, armazenados em um bucket nas seguintes classes de armazenamento:</p> <ul style="list-style-type: none"> • S3 Standard (STANDARD) • S3 Intelligent-Tiering (INTELLIGENT_TIERING) • S3 Standard-Infrequent Access (STANDARD_IA) • S3 One Zone-Infrequent Access (ONEZONE_IA) • Reduced Redundancy Storage (RRS) (REDUCED_REDUNDANCY) • S3 Glacier Instant Retrieval (GLACIER_IR) • S3 Glacier Deep Archive (DEEP_ARCHIVE) • S3 Glacier Flexible Retrieval (GLACIER) • S3 Express One Zone (EXPRESS_ONEZONE) <p>O valor é calculado somando o tamanho de todos os objetos e metadados (como nomes de buckets) no bucket (objetos atuais e não atuais), incluindo o tamanho de todas as partes de todos os carregamentos fracionados incompletos do bucket.</p> <p>Filtros de tipo de armazenamento válidos: StandardStorage , IntelligentTieringFAStorage , IntelligentTieringIAStorage , IntelligentTieringAAStorage , IntelligentTieringAIAStorage , IntelligentTieringDAASStorage , StandardIAStorage , StandardIASizeOverhead , StandardIAObjectOverhead , OneZoneIAStorage , OneZoneIASizeOverhead , ReducedRedundancyStorage , GlacierInstantRetrievalSizeOverhead , GlacierInstantRetrievalStorage , GlacierStorage , GlacierStagingStorage , GlacierObjectOverhead , GlacierS3ObjectOverhead , DeepArchiveStorage , DeepArchiveObjectOverhead</p>

Métrica	Descrição
	<p>verhead , DeepArchiveS3ObjectOverhead , DeepArchiveStagingStorage e ExpressOneZone (consulte a dimensão StorageType)</p> <p>Unidade: bytes</p> <p>Estatística válida: média</p>
NumberOfObjects	<p>O número total de objetos armazenados em um bucket de uso geral para todas as classes de armazenamento. Esse valor é calculado contando todos os objetos do bucket, que incluem objetos atuais e não atuais, marcadores de exclusão e o número total de partes de todos os carregamentos fracionados incompletos do bucket. Para buckets de diretório com objetos na classe de armazenamento S3 Express One Zone, esse valor é calculado contando todos os objetos do bucket, mas não inclui uploads múltiplos incompletos no bucket.</p> <p>Filtros de tipo de armazenamento válidos: AllStorageTypes (consulte a dimensão StorageType)</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. Essas métricas incluem solicitações não faturáveis (no caso de solicitações GET de CopyObject e Replication).

Note

As métricas de solicitação do Amazon S3 no CloudWatch não são compatíveis com buckets de diretório.

Métrica	Descrição
AllRequests	<p>O número total de solicitações HTTP feitas em um bucket do Amazon S3, independentemente do tipo. Se você estiver usando uma configuração de métricas com um filtro, então essa métrica só retornará as solicitações HTTP que atendam aos requisitos do filtro.</p> <p>Unidades: contagem</p> <p>Estatística válida: soma</p>
GetRequests	<p>O número de solicitações HTTP GET feitas para objetos em um bucket do Amazon S3. Isso não inclui operações de lista. Essa métrica é incrementada para a origem de cada solicitação CopyObject .</p> <p>Unidades: contagem</p> <p>Estatística válida: soma</p> <div data-bbox="472 968 1507 1234" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"><p> Note</p><p>Solicitações paginadas orientadas a listas, como ListMulti partUploads, ListParts e ListObjectVersions, não estão incluídas nessa métrica.</p></div>
PutRequests	<p>O número de solicitações HTTP PUT feitas para objetos em um bucket do Amazon S3. Essa métrica é incrementada para o destino de cada solicitação CopyObject .</p> <p>Unidades: contagem</p> <p>Estatística válida: soma</p>
DeleteRequests	<p>O número de solicitações HTTP DELETE feitas para objetos em um bucket do Amazon S3. Essa métrica também inclui as solicitações DeleteObjects. Essa métrica exibe o número de solicitações feitas, não o número de objetos excluídos.</p> <p>Unidades: contagem</p>

Métrica	Descrição
	Estatística válida: soma
HeadRequests	<p>O número de solicitações HTTP HEAD feitas para um bucket do Amazon S3.</p> <p>Unidades: contagem</p> <p>Estatística válida: soma</p>
PostRequests	<p>O número de solicitações HTTP POST feitas para um bucket do Amazon S3.</p> <p>Unidades: contagem</p> <p>Estatística válida: soma</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>As solicitações DeleteObjects e SelectObjectContent não estão incluídas nessa métrica.</p> </div>
SelectRequests	<p>O número de solicitações SelectObjectContent do Amazon S3 feitas para objetos em um bucket do Amazon S3.</p> <p>Unidades: contagem</p> <p>Estatística válida: soma</p>
SelectBytesScanned	<p>O número de bytes de dados verificados com solicitações SelectObjectContent do Amazon S3 em um bucket do Amazon S3.</p> <p>Unidades: 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
SelectBytesReturned	<p>O número de bytes de dados retornados com solicitações SelectObjectContent do Amazon S3 em um bucket do Amazon S3.</p> <p>Unidades: 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 carregados para solicitações feitas a um bucket do Amazon S3, em que a solicitação inclui um corpo.</p> <p>Unidades: 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	<p>O número de solicitações com 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 Average (Média) mostra a taxa de erros, e a estatística Sum (Soma) mostra a contagem desse tipo de erro, durante cada período.</p> <p>Unidades: contagem</p> <p>Estatísticas válidas: média (relatórios por solicitação), soma (relatórios por período), mín., máx., contagem de amostras</p>
5xxErrors	<p>O número de solicitações com 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 Average (Média) mostra a taxa de erros, e a estatística Sum (Soma) mostra a contagem desse tipo de erro, durante cada período.</p> <p>Unidades: contagem</p> <p>Estatísticas válidas: média (relatórios por solicitação), soma (relatórios por período), mín., máx., contagem de amostras</p>
FirstByte Latency	<p>O tempo por solicitação desde o recebimento da solicitação completa por um bucket do Amazon S3 até o momento em que a resposta começa a ser retornada.</p> <p>Unidade: milissegundos</p> <p>Estatísticas válidas: Average (Média), Sum (Soma), Min (Mínimo), Max (Máximo) (o mesmo que p100), Sample Count (Contagem de amostras), qualquer percentil entre p0,0 e p100</p>

Métrica	Descrição
TotalRequestLatency	<p>O tempo por solicitação decorrido do primeiro byte recebido até o último byte enviado para um bucket do Amazon S3. Essa métrica inclui o tempo necessário para receber o corpo da solicitação e enviar o corpo da resposta, que não está incluído em FirstByteLatency .</p> <p>Unidade: milissegundos</p> <p>Estatísticas válidas: Average (Média), Sum (Soma), Min (Mínimo), Max (Máximo) (o mesmo que p100), Sample Count (Contagem de amostras), qualquer percentil entre p0,0 e p100</p>

Métricas de replicação do 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	<p>O número máximo de segundos em que a Região da AWS de destino da replicação está atrás da Região da AWS de origem de uma determinada regra de replicação.</p> <p>Unidade: segundos</p> <p>Estatísticas válidas: Max</p>

Métrica	Descrição
BytesPendingReplication	<p>O número total de bytes de objetos com replicação pendente para uma determinada regra de replicação.</p> <p>Unidade: bytes</p> <p>Estatísticas válidas: Max</p>
OperationSPendingReplication	<p>O número de operações com replicação pendente para uma determinada regra de replicação.</p> <p>Unidades: contagem</p> <p>Estatísticas válidas: Max</p>
OperationSFailedReplication	<p>O número de operações com falha na replicação para determinada regra de replicação.</p> <p>Unidades: contagem</p> <p>Estatísticas válidas: soma (número total de operações com falha), média (taxa de falha), contagem de amostras (número total de operações de replicação)</p>

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. 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 que foram atualizados para métricas e recomendações avançadas.

Para obter uma lista das métricas do S3 Storage Lens publicadas no CloudWatch, consulte [Glossário de métricas de lente de armazenamento do Amazon S3](#). Para obter uma lista completa de dimensões, consulte [Dimensões](#).

Métricas de solicitação do S3 Object Lambda no CloudWatch

O S3 Object Lambda inclui as métricas de solicitação a seguir.

Métrica	Descrição
AllRequests	<p>O número total de solicitações HTTP feitas para um bucket do Amazon S3 usando um ponto de acesso do Object Lambda.</p> <p>Unidades: contagem</p> <p>Estatística válida: soma</p>
GetRequests	<p>O número de solicitações HTTP GET feitas para objetos usando um ponto de acesso do Object Lambda. Essa métrica não inclui operações de lista.</p> <p>Unidades: contagem</p> <p>Estatística válida: soma</p>
BytesUploaded	<p>O número de bytes carregados para um bucket do Amazon S3 usando um ponto de acesso do Object Lambda, para solicitações que incluem um corpo.</p> <p>Unidades: 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>
PostRequests	<p>O número de solicitações HTTP POST feitas para um bucket do Amazon S3 usando um ponto de acesso do Object Lambda.</p> <p>Unidades: contagem</p> <p>Estatística válida: soma</p>
PutRequests	<p>O número de solicitações HTTP PUT feitas para objetos em um bucket do Amazon S3 usando um ponto de acesso do Object Lambda.</p> <p>Unidades: contagem</p> <p>Estatística válida: soma</p>

Métrica	Descrição
DeleteRequests	<p>O número de solicitações HTTP DELETE feitas para objetos em um bucket do Amazon S3 usando um ponto de acesso do Object Lambda. Essa métrica inclui as solicitações DeleteObjects. Essa métrica exibe o número de solicitações feitas, não o número de objetos excluídos.</p> <p>Unidades: contagem</p> <p>Estatística válida: soma</p>
BytesDownloaded	<p>O número de bytes baixados para solicitações feitas a um bucket do Amazon S3 usando um ponto de acesso do Object Lambda, em que a resposta inclui um corpo.</p> <p>Unidades: 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>
FirstByte Latency	<p>O tempo por solicitação desde o recebimento da solicitação completa por um bucket do Amazon S3 por meio de um ponto de acesso do Object Lambda até o momento em que a resposta começa a ser retornada. Essa métrica depende do tempo de execução da função do AWS Lambda para transformar o objeto antes que a função retorne os bytes para o ponto de acesso do Object Lambda.</p> <p>Unidade: milissegundos</p> <p>Estatísticas válidas: Average (Média), Sum (Soma), Min (Mínimo), Max (Máximo) (o mesmo que p100), Sample Count (Contagem de amostras), qualquer percentil entre p0,0 e p100</p>

Métrica	Descrição
TotalRequestLatency	<p>O tempo por solicitação decorrido do primeiro byte recebido até o último byte enviado para um ponto de acesso do Object Lambda. Essa métrica inclui o tempo necessário para receber o corpo da solicitação e enviar o corpo da resposta, que não está incluído em <code>FirstByteLatency</code> .</p> <p>Unidade: milissegundos</p> <p>Estatísticas válidas: Average (Média), Sum (Soma), Min (Mínimo), Max (Máximo) (o mesmo que p100), Sample Count (Contagem de amostras), qualquer percentil entre p0,0 e p100</p>
HeadRequests	<p>O número de solicitações HTTP HEAD feitas para um bucket do Amazon S3 usando um ponto de acesso do Object Lambda.</p> <p>Unidades: contagem</p> <p>Estatística válida: soma</p>
ListRequests	<p>O número de solicitações HTTP GET que listam o conteúdo de um bucket do Amazon S3. Essa métrica inclui as operações <code>ListObjects</code> e <code>ListObjectsV2</code> .</p> <p>Unidades: contagem</p> <p>Estatística válida: soma</p>
4xxErrors	<p>O número de solicitações com código de status de erro do servidor HTTP 4xx feitas para um bucket do Amazon S3 usando um ponto de acesso do Object Lambda com um valor de 0 ou 1. A estatística Average (Média) mostra a taxa de erros, e a estatística Sum (Soma) mostra a contagem desse tipo de erro, durante cada período.</p> <p>Unidades: contagem</p> <p>Estatísticas válidas: média (relatórios por solicitação), soma (relatórios por período), mín., máx., contagem de amostras</p>

Métrica	Descrição
<code>5xxErrors</code>	<p>O número de solicitações com código de status de erro do servidor HTTP 5xx feitas para um bucket do Amazon S3 usando um ponto de acesso do Object Lambda com um valor de 0 ou 1. A estatística Average (Média) mostra a taxa de erros, e a estatística Sum (Soma) mostra a contagem desse tipo de erro, durante cada período.</p> <p>Unidades: contagem</p> <p>Estatísticas válidas: média (relatórios por solicitação), soma (relatórios por período), mín., máx., contagem de amostras</p>
<code>ProxiedRequests</code>	<p>O número de solicitações HTTP para um ponto de acesso do Object Lambda que retornam a resposta padrão de API do Amazon S3. (Essas solicitações não têm uma função do Lambda configurada.)</p> <p>Unidades: contagem</p> <p>Estatística válida: soma</p>
<code>InvokedLambda</code>	<p>O número de solicitações HTTP para um objeto do S3 em que uma função do Lambda foi invocada.</p> <p>Unidades: contagem</p> <p>Estatística válida: soma</p>
<code>LambdaResponseRequests</code>	<p>O número de solicitações <code>WriteGetObjectResponse</code> feitas pela função do Lambda. Esta métrica aplica-se apenas às solicitações <code>GetObject</code>.</p>
<code>LambdaResponse4xx</code>	<p>O número de erros de cliente HTTP 4xx que ocorrem ao chamar <code>WriteGetObjectResponse</code> de uma função do Lambda. Essa métrica fornece as mesmas informações que <code>4xxErrors</code>, mas somente para chamadas de <code>WriteGetObjectResponse</code>.</p>

Métrica	Descrição
LambdaResponse5xx	O número de erros de servidor HTTP 5xx que ocorrem ao chamar <code>WriteGetObjectResponse</code> de uma função do Lambda. Essa métrica fornece as mesmas informações que <code>5xxErrors</code> , mas somente para chamadas de <code>WriteGetObjectResponse</code> .

Métricas do Amazon S3 on Outposts no CloudWatch

Para obter uma lista de métricas no CloudWatch que são usadas para buckets do S3 no Outposts, consulte [Métricas do CloudWatch](#).

Dimensões do Amazon S3 no CloudWatch

As seguintes dimensões são usadas para filtrar as métricas do Amazon S3.

Dimensão	Descrição
BucketName	Essa dimensão filtra os dados que você solicita apenas para o bucket identificado.
StorageType	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 Archive Access da classe de armazenamento INTELLIGENT_TIERING.<code>IntelligentTieringAIASStorage</code>: o número de bytes usados para objetos no nível Archive Instant Access da classe de armazenamento INTELLIGENT_TIERING.<code>IntelligentTieringDAASStorage</code>: o número de bytes usados para objetos no nível Deep Archive Access da classe de armazenamento INTELLIGENT_TIERING.

Dimensão	Descrição
	<ul style="list-style-type: none"> • <code>IntelligentTieringFAStorage</code> : o número de bytes usados para objetos no nível Frequent Access da classe de armazenamento <code>INTELLIGENT_TIERING</code> . • <code>IntelligentTieringIAStorage</code> : o número de bytes usados para objetos no nível Infrequent Access da classe de armazenamento <code>INTELLIGENT_TIERING</code> . • <code>StandardIAStorage</code> : o número de bytes usados para objetos na classe de armazenamento S3 Standard-Infrequent Access (<code>STANDARD_IA</code>). • <code>StandardIASizeOverhead</code> : o número de bytes usados para objetos com menos de 128 KB na classe de armazenamento <code>STANDARD_IA</code> . • <code>IntAAObjectOverhead</code> : para cada objeto na classe de armazenamento <code>INTELLIGENT_TIERING</code> no nível Archive Access, o S3 Glacier adiciona 32 KB de armazenamento para índice e metadados relacionados. Esses dados adicionais são necessários para identificar e recuperar seu objeto. Você é cobrado nas tarifas do S3 Glacier Flexible Retrieval por esse armazenamento adicional. • <code>IntAAS3ObjectOverhead</code> : para cada objeto na classe de armazenamento <code>INTELLIGENT_TIERING</code> no nível Archive Access, o Amazon S3 usa 8 KB de armazenamento para o nome do objeto e outros metadados. Você é cobrado com as tarifas S3 Standard pelo armazenamento adicional. • <code>IntDAAObjectOverhead</code> : para cada objeto na classe de armazenamento <code>INTELLIGENT_TIERING</code> no nível Deep Archive Access, o S3 Glacier adiciona 32 KB de armazenamento para índice e metadados relacionados. Esses dados adicionais são necessários para identificar e recuperar seu objeto. Você é cobrado pelas taxas do S3 Glacier Deep Archive nesse armazenamento adicional. • <code>IntDAAS3ObjectOverhead</code> : para cada objeto na classe de armazenamento <code>INTELLIGENT_TIERING</code> no nível

Dimensão	Descrição
	<p>Deep Archive Access, o Amazon S3 adiciona 8 KB de armazenamento para índice e metadados relacionados. Esses dados adicionais são necessários para identificar e recuperar seu objeto. Você é cobrado com as tarifas S3 Standard pelo armazenamento adicional.</p> <ul style="list-style-type: none">• <code>OneZoneIAStorage</code> : o número de bytes usados para objetos na classe de armazenamento S3 One Zone-Infrequent Access (<code>ONEZONE_IA</code>).• <code>OneZoneIASizeOverhead</code> : o número de bytes usados para objetos com menos de 128 KB na classe de armazenamento <code>ONEZONE_IA</code>.• <code>ReducedRedundancyStorage</code> : o número de bytes usados para objetos na classe de armazenamento de redundância reduzida (RRS - Reduced Redundancy Storage).• <code>GlacierInstantRetrievalSizeOverhead</code> : o número de bytes usados para objetos com menos de 128 KB na classe de armazenamento Instant Retrieval do S3 Glacier.• <code>GlacierInstantRetrievalStorage</code> : o número de bytes usados para objetos na classe de armazenamento Instant Retrieval do S3 Glacier.• <code>GlacierStorage</code> : o número de bytes usados para objetos na classe de armazenamento S3 Glacier Flexible Retrieval.• <code>GlacierStagingStorage</code> : o número de bytes usados para as partes de objetos fracionados antes que a solicitação <code>CompleteMultipartUpload</code> seja concluída em objetos na classe de armazenamento Flexible Retrieval do S3 Glacier.• <code>GlacierObjectOverhead</code> : para cada objeto arquivado, o S3 Glacier adiciona 32 KB de armazenamento para índice e metadados relacionados. Esses dados adicionais são necessários para identificar e recuperar seu objeto. Você é cobrado nas tarifas do S3 Glacier Flexible Retrieval por esse armazenamento adicional.

Dimensão	Descrição
	<ul style="list-style-type: none"> • <code>GlacierS3ObjectOverhead</code> : 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 metadados. Você é cobrado com as tarifas S3 Standard pelo armazenamento adicional. • <code>DeepArchiveStorage</code> : o número de bytes usados para objetos na classe de armazenamento S3 Glacier Deep Archive. • <code>DeepArchiveObjectOverhead</code> : para cada objeto arquivado, o S3 Glacier adiciona 32 KB de armazenamento para índice e metadados relacionados. Esses dados adicionais são necessários para identificar e recuperar seu objeto. Você é cobrado pelas taxas do S3 Glacier Deep Archive nesse armazenamento adicional. • <code>DeepArchiveS3ObjectOverhead</code> : 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 com as tarifas S3 Standard pelo armazenamento adicional. • <code>DeepArchiveStagingStorage</code> : o número de bytes usados para partes de objetos fracionados antes que a solicitação <code>CompleteMultipartUpload</code> seja concluída em objetos na classe de armazenamento Deep Archive do S3 Glacier. • <code>ExpressOneZone</code> : o número de bytes usados para objetos na classe de armazenamento S3 Express One Zone.
<code>FilterId</code>	<p>Esta dimensão filtra as configurações de métricas que você especifica para métricas de solicitação em um bucket. Ao criar uma configuração de métricas, especifique um ID de filtro (por exemplo, um prefixo, uma etiqueta ou um ponto de acesso). Para obter mais informações, consulte Criar uma configuração de métricas.</p>

Dimensões da funcionalidade Replicação do S3 no CloudWatch

As seguintes dimensões são usadas para filtrar as métricas da funcionalidade Replicação do S3.

Dimensão	Descrição
SourceBucket	O nome do bucket do qual os objetos são replicados.
DestinationBucket	O nome do bucket para o qual os objetos são replicados.
RuleId	Um identificador exclusivo da regra que acionou a atualização da métrica de replicação.

Dimensões do S3 Storage Lens no CloudWatch

Para obter uma lista das dimensões usadas para filtrar métricas do S3 Storage Lens no CloudWatch, consulte [Dimensões](#).

Dimensões de solicitação do S3 Object Lambda no CloudWatch

As dimensões a seguir são usadas para filtrar dados de um ponto de acesso do Object Lambda.

Dimensão	Descrição
AccessPointName	O nome do ponto de acesso do qual as solicitações estão sendo feitas.
DataSourceARN	A origem da qual o ponto de acesso do Object Lambda está recuperando os dados. Se a solicitação invocar uma função do Lambda, isso se referirá ao nome do recurso da Amazon (ARN) do Lambda. Caso contrário, isso se refere ao ARN do ponto de acesso.

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.

Usando a AWS CLI

Por exemplo, se você quiser usar a AWS CLI para obter a média do tamanho de um bucket específico, em bytes, poderá usar o seguinte comando:

```
aws cloudwatch get-metric-statistics --metric-name BucketSizeBytes --namespace AWS/S3
--start-time 2016-10-19T00:00:00Z --end-time 2016-10-20T00:00:00Z --statistics Average
--unit Bytes --region us-west-2 --dimensions Name=BucketName,Value=DOC-EXAMPLE-BUCKET
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 visualizar as métricas usando o console do Amazon CloudWatch

1. Abra o console CloudWatch em <https://console.aws.amazon.com/cloudwatch/>.
2. No painel de navegação à esquerda, escolha Metrics.
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.

Como visualizar 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"
```

Para obter mais informações sobre as permissões necessárias para acessar painéis do CloudWatch, consulte [Permissões de painéis do CloudWatch](#) no Guia do usuário do Amazon CloudWatch.

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](#).

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](#). 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](#).

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](#). 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](#)
- [Criação de uma configuração de métricas que filtre por prefixo, etiqueta de objeto ou ponto de acesso](#)

- [Excluir um filtro de métricas](#)

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](#).

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. As [métricas da funcionalidade Replicação do S3](#) são métricas detalhadas para as regras de replicação na configuração de replicação. Com as métricas de replicação, você pode monitorar o andamento minuto a minuto rastreando bytes pendentes, operações pendentes, operações com falha na replicação e latência de replicação.

Para obter mais informações sobre métricas do CloudWatch para o Amazon S3, consulte [Monitoramento de métricas com o Amazon CloudWatch](#).

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](#).

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 [Definição de preços 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](#)

Usando a 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. As [métricas da funcionalidade Replicação do S3](#) são métricas detalhadas para as regras de replicação na configuração de replicação. Com as métricas de replicação, você pode monitorar o andamento minuto a minuto rastreando bytes pendentes, operações pendentes, operações com falha na replicação e latência de replicação.

Para obter mais informações sobre métricas do CloudWatch para o Amazon S3, consulte [Monitoramento de métricas com o Amazon CloudWatch](#).

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](#).

- 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](#).
- 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](#).

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](#).

Se você especificar um filtro, somente solicitações que operem em objetos únicos podem corresponder ao filtro e serem incluídas nas métricas relatadas. Algumas solicitações, como [DeleteObjects](#) e `ListObjects`, 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. Escolha **Salvar alterações**.

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 ter mais informações, consulte [Criar uma configuração de métricas do CloudWatch para todos os objetos em seu bucket](#).

Usando a 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"
  }
}'
```

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](#)

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](#)

- [Criação de uma configuração de métricas que filtre por prefixo, etiqueta de objeto ou ponto de acesso](#)

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.

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](#). 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](#).

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 do AWS Lambda
- Amazon EventBridge

Para ter mais informações, consulte [Destinos de eventos suportados](#).

Note

As filas FIFO (primeiro entrar, primeiro a sair) do Amazon Simple Queue Service não são compatíveis como destino de notificação de evento do Amazon S3. Para enviar uma notificação de um evento do Amazon S3 para uma fila FIFO do Amazon SQS, você pode usar o Amazon EventBridge. Para ter mais informações, consulte [Habilitar o Amazon EventBridge](#).

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](#)
- [Usar o Amazon SQS, o Amazon SNS e o Lambda](#)
- [Usar o EventBridge](#)

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. Somente um destino pode ser especificado para cada notificação de evento. As notificações de eventos do Amazon S3 enviam uma entrada de evento para cada mensagem de notificação.

Tópicos

- [Destinos de eventos suportados](#)
- [Tipos de evento compatíveis com SQS, SNS e Lambda](#)
- [Tipos de evento compatíveis com o Amazon EventBridge](#)

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

No entanto, somente um tipo de destino pode ser especificado para cada notificação de evento.

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 vento a um destino](#).

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

Note

As filas FIFO (primeiro entrar, primeiro a sair) do Amazon Simple Queue Service não são compatíveis como destino de notificação de evento do Amazon S3. Para enviar uma notificação de um evento do Amazon S3 para uma fila FIFO do Amazon SQS, você pode usar o Amazon EventBridge. Para ter mais informações, consulte [Habilitar o Amazon EventBridge](#).

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>

Tipos de evento	Descrição
<p>s3:ObjectCreated:*</p> <p>s3:ObjectCreated:Put</p> <p>s3:ObjectCreated:Post</p> <p>s3:ObjectCreated:Copy</p> <p>s3:ObjectCreated:CompleteMultipartUpload</p>	<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 <code>s3:ObjectCreated:*</code> para solicitar notificação, qualquer que seja a API usada para criar um objeto.</p> <p><code>s3:ObjectCreated:CompleteMultipartUpload</code> inclui objetos que são criados usando UploadPartCopy para operações de cópia.</p>
<p>s3:ObjectRemoved:*</p> <p>s3:ObjectRemoved>Delete</p> <p>s3:ObjectRemoved>DeleteMarkerCreated</p>	<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 <code>s3:ObjectRemoved>Delete</code>. Se preferir, você pode solicitar uma notificação quando um marcador de exclusão for criado para um objeto com versionamento usando <code>s3:ObjectRemoved>DeleteMarkerCreated</code>. Para obter instruções sobre como excluir objetos com versionamento, consulte Excluir versões de objetos de um bucket com versionamento habilitado. Você também pode usar um caractere curinga <code>s3:ObjectRemoved:*</code> para solicitar uma notificação sempre que um objeto for excluído.</p> <p>Essas notificações de eventos não alertam sobre exclusões automáticas de configurações de ciclo de vida ou operações que falharam.</p>

Tipos de evento	Descrição
<p>s3:ObjectRestore:*</p> <p>s3:ObjectRestore:Post</p> <p>s3:ObjectRestore:Completed</p> <p>s3:ObjectRestore:Delete</p>	<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, S3 Glacier Deep Archive, nível de acesso S3 Intelligent-Tiering Archive e nível de acesso S3 Intelligent-Tiering Deep Archive. Você também pode receber notificações quando a cópia restaurada de um objeto expirar.</p> <p>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.</p>
<p>s3:ReducedRedundancyLostObject</p>	<p>Você recebe esse evento de notificação quando o Amazon S3 detecta que um objeto da classe de armazenamento RRS foi perdido.</p>

Tipos de evento	Descrição
<p>s3:Replication:*</p> <p>s3:Replication:OperationFailedReplication</p> <p>s3:Replication:OperationMissedThreshold</p> <p>s3:Replication:OperationReplicatedAfterThreshold</p> <p>s3:Replication:OperationNotTracked</p>	<p>Com tipos de evento Replication, você pode receber notificações de eventos para configurações de replicação o que tenham métricas de replicação do S3 ou o Controle de Tempo de Replicação do S3 (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 Monitorar o progresso com métricas de replicação e Notificações de eventos do S3.</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>

Tipos de evento	Descrição
<p>s3:LifecycleExpiration:*</p> <p>s3:LifecycleExpiration>Delete</p> <p>s3:LifecycleExpiration>DeleteMarkerCreated</p>	<p>Ao usar os tipos de evento LifecycleExpiration, 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 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.</p>
<p>s3:LifecycleTransition</p>	<p>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.</p>
<p>s3: IntelligentTiering</p>	<p>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.</p>
<p>s3:ObjectTagging:*</p> <p>s3:ObjectTagging:Put</p> <p>s3:ObjectTagging>Delete</p>	<p>Usando os tipos de evento ObjectTagging, você pode habilitar a notificação quando uma etiqueta de objeto é adicionada ou excluída de um objeto.</p> <p>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.</p>

Tipos de evento	Descrição
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 [Usar o EventBridge](#).

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](#).
- 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 \(tópico do SNS ou fila do SQS\)](#).

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](#).

Tópicos

- [Conceder permissões para publicar mensagens de notificação de vento a um destino](#)
- [Habilitar e configurar notificações de eventos usando o console do Amazon S3](#)
- [Configuração de notificações de eventos de forma programática](#)
- [Demonstração: configurar um bucket para notificações \(tópico do SNS ou fila do SQS\)](#)
- [Configurar notificações de eventos usando filtragem de nomes de chave de objeto](#)
- [Estrutura de mensagens de evento](#)

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.

Para solucionar problemas de publicação de mensagens de notificação de eventos em um destino, consulte [Solução de problemas para publicar notificações de eventos do Amazon S3 em um tópico do Amazon Simple Notification Service](#).

Tópicos

- [Conceder permissões para invocar uma função do AWS Lambda](#)
- [Conceder permissões para publicar mensagens em um tópico do SNS ou em uma fila do SQS](#)

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](#).

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 \(tópico do SNS ou fila do SQS\)](#). 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 \(tópico do SNS ou fila do SQS\)](#).

```
{
  "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 \(tópico do SNS ou fila do SQS\)](#).

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`.

Quando o `ArnLike` é usado, as partes partição, serviço, ID da conta, tipo de recurso e ID parcial do recurso do ARN devem ter a correspondência exata com o ARN no contexto da solicitação. Somente a região e o caminho do recurso permitem a correspondência parcial.

Quando `StringLike` é usado em vez de `ArnLike`, a correspondência ignora a estrutura do ARN e permite a correspondência parcial, independentemente da parte que foi marcada como curinga. Para obter mais informações, consulte [Elementos de política JSON do IAM](#) no Guia do usuário do IAM.

```

"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](#).

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](#) e [Conceder permissões para publicar mensagens de notificação de evento a um destino](#).

Note

As filas FIFO (primeiro entrar, primeiro a sair) do Amazon Simple Queue Service não são compatíveis como destino de notificação de evento do Amazon S3. Para enviar uma notificação de um evento do Amazon S3 para uma fila FIFO do Amazon SQS, você pode

usar o Amazon EventBridge. Para ter mais informações, consulte [Habilitar o Amazon EventBridge](#).

Tópicos

- [Habilitar notificações do Amazon SNS, Amazon SQS ou Lambda usando o console do Amazon S3](#)

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](#).

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](#).

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](#).

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>
```

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](#).

Para obter mais informações sobre como configurar notificações de eventos, consulte os seguintes tópicos:

- [Demonstração: configurar um bucket para notificações \(tópico do SNS ou fila do SQS\)](#).
- [Configurar notificações de eventos usando filtragem de nomes de chave de objeto](#)

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.

Note

As filas FIFO (primeiro entrar, primeiro a sair) do Amazon Simple Queue Service não são compatíveis como destino de notificação de evento do Amazon S3. Para enviar uma notificação de um evento do Amazon S3 para uma fila FIFO do Amazon SQS, você pode usar o Amazon EventBridge. Para ter mais informações, consulte [Habilitar o Amazon EventBridge](#).

Tópicos

- [Resumo da demonstração](#)
- [Etapa 1: Criar uma fila do Amazon SQS](#)

- [Etapa 2: Criar um tópico do Amazon SNS](#)
- [Etapa 3: Adicionar a configuração de notificação para o bucket](#)
- [Etapa 4: Testar a configuração](#)

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](#).

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 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](#).

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](#).

```
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](#).

```
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.

Note

Um caractere curinga (“*”) não pode ser usado em filtros como um prefixo ou sufixo. Se seu prefixo ou sufixo contiver um espaço, você deverá substituí-lo pelo caractere “+”. Se você usar qualquer outro caractere especial no valor do prefixo ou sufixo, deverá inseri-los no [formato codificado em URL \(codificado em porcentagem\)](#). Para obter uma lista completa de caracteres especiais que devem ser convertidos em formato codificado em URL quando usados em um prefixo ou sufixo para notificações de eventos, consulte [Caracteres seguros](#).

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](#).

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](#).

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.

Configurações de notificação que usam o `Filter` não podem definir regras de filtragem com prefixos 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](#)
- [Exemplos de configurações de notificação com sobreposição inválida de prefixo e sufixo](#)

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>
    <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](#).

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](#).

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 `sequenceNumber` 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": "8cLeGAmw098X5cv4Zkwcmo8vvZa3eH3eKxsPzbB9wR+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": "AIDAJDPLRKL7UEXAMPLE"
      },
      "requestParameters": {
        "sourceIPAddress": "127.0.0.1"
      },
      "responseElements": {
        "x-amz-request-id": "C3D13FE58DE4C810",
        "x-amz-id-2": "FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/
JRWeUWerMUE5JgHvAN0jpd"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",
        "bucket": {
          "name": "mybucket",
          "ownerIdentity": {
            "principalId": "A3NL1K0ZZKExample"
          },
          "arn": "arn:aws:s3:::mybucket"
        },
        "object": {
          "key": "HappyFace.jpg",
          "size": 1024,
          "eTag": "d41d8cd98f00b204e9800998ecf8427e",
          "versionId": "096fKKXTRTt13on89fv0.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.

Usar o 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	<p>Um objeto foi criado.</p> <p>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.</p>
Objeto excluído (DeleteObject)	<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> <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).</p>
Objeto excluído (validade do ciclo de vida)	

Tipo de evento	Descrição
	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 .
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 .
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 .
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 .
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) .
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 .

Tipo de evento	Descrição
Etiquetas de objeto excluídas	Todas as etiquetas foram removidas de um objeto usando DeleteObjectTagging. Para obter mais informações, consulte Categorizando seu armazenamento usando tags .

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](#).

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 mais informações sobre as ações e os tipos de dados com os quais você pode interagir usando a API do EventBridge, consulte [Amazon EventBridge API Reference](#) na Referência da API do Amazon EventBridge.

Para obter informações sobre preços, consulte [Preços do Amazon EventBridge](#).

Tópicos

- [Permissões do Amazon EventBridge](#)
- [Habilitar o Amazon EventBridge](#)
- [Estrutura de mensagens de eventos do EventBridge](#)
- [Mapeamento e solução de problemas do Amazon EventBridge](#)

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 subsecção Amazon EventBridge. Selecione a opção 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.

Usando a 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 da API REST

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 [Usar o EventBridge](#) 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.
- **resources (recursos):** 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 nesse campo, consulte [Campo de detalhes da mensagem de evento](#).

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-7ededc63b103",
  "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": "1QW9g1Z99LUNbvaaYVpw9xD10LU.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": "617b398000000000"
},
"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 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](#).
- **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](#).
- **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](#).
- **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](#).

- `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](#).
- `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](#).

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:CompleteMulti partUpload	
ObjectRemoved:Delete	Objeto excluído
ObjectRemoved:DeleteMarkerCreated	
LifecycleExpiration:Delete	
LifecycleExpiration:DeleteMarkerCreated	
ObjectRestore:Post	Restauração de objeto iniciada
ObjectRestore:Completed	Restauração de objeto concluída

Tipo de evento do S3	Tipo de detalhes do Amazon EventBridge
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](#)
- [Avaliação de sua atividade de armazenamento e uso com a Lente de Armazenamento do Amazon S3](#)
- [Rastreamento de solicitações do Amazon S3 usando o AWS X-Ray](#)

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](#).

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 ajudar a melhorar suas configurações 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 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 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 configurações de ciclo de vida que fazem 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?](#)
- [Como usar a análise de classe de armazenamento?](#)
- [Como exportar dados de análise de classe de armazenamento?](#)
- [Configurar análise de classe de armazenamento](#)

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](#).
- 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 recuperei?

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 recuperei?

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 Inventário do S3 e análises do S3](#).

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	CumulativeAccessRatio	ObjectAgeForSIATransition	RecommendedObjectAgeForSIATransition
8/17/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
9/2/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/22/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
8/25/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
9/6/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/30/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/28/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/21/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
9/5/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		

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			0	000-014	
9/3/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.02426125	015-029	
8/28/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.03545875	015-029	
8/17/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014	
8/25/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014	
9/6/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.0209529	015-029	
9/4/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.02304819	015-029	
8/22/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014	
8/21/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014	
8/30/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.03073092	015-029	
8/20/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014	

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](#).

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 sua chave gerenciada pelo cliente para criptografia](#).

Para obter mais informações sobre análises, consulte [Análise do Amazon S3 – Análise de classe de armazenamento](#).

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 sua chave gerenciada pelo cliente para criptografia](#).

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. Isso permitirá 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](#).

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 a Lente de Armazenamento do Amazon S3

A Lente de Armazenamento do Amazon S3 é um recurso de análise de armazenamento em nuvem que você pode usar para obter visibilidade em toda a organização sobre o uso e a atividade do armazenamento de objetos. A Lente de Armazenamento do S3 também analisa as métricas para fornecer recomendações contextuais que você pode usar para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados.

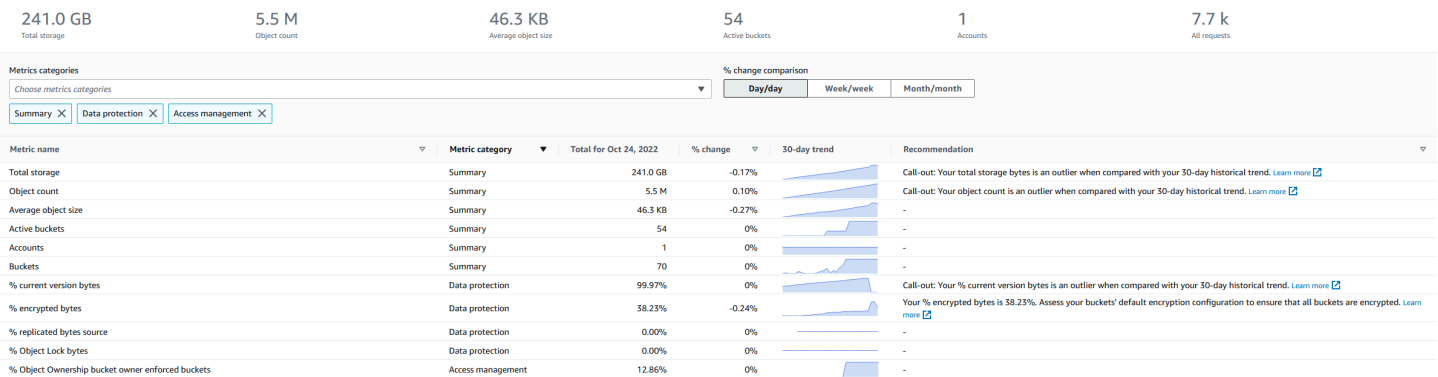
Você pode usar métricas de Lente de Armazenamento do S3 para gerar insights resumidos. Por exemplo, você pode descobrir quanto armazenamento você tem em toda a sua organização ou quais são os buckets e prefixos de crescimento mais rápido. Você também pode usar as métricas da Lente de Armazenamento do S3 para identificar oportunidades de otimização de custos, implementar práticas recomendadas de proteção de dados e gerenciamento de acesso, e melhorar a performance das workloads de aplicações. Por exemplo, você pode identificar buckets que não têm regras de Ciclo de Vida do S3 para anular multipart uploads que estão incompletos há mais de 7 dias. Você também pode identificar buckets que não estão seguindo as práticas recomendadas de proteção de dados, como usar a Replicação do S3 ou o Versionamento do S3.

A Lente de Armazenamento do S3 agrega suas métricas e exibe as informações na seção Account snapshot (Snapshot da conta) na página Buckets do console do Amazon S3. A Lente de Armazenamento do S3 também oferece um painel interativo 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. O painel tem opções de detalhamento para gerar e visualizar insights no nível da organização, conta, Região da AWS, classe de

armazenamento, bucket, prefixo ou grupo de Lente de Armazenamento. Você também pode enviar uma exportação diária de métricas em CSV ou no formato Parquet para um bucket do S3.

Snapshot for Oct 24, 2022

Snapshot is a curated list of frequently used metrics. You can view additional metrics in your dashboard graphs and tables. A metrics glossary is available. [Learn more](#)



Métricas e recursos da Lente de Armazenamento do S3

A Lente de Armazenamento do S3 fornece um painel padrão interativo que é atualizado diariamente. A Lente de Armazenamento do S3 pré-configura esse painel para visualizar um resumo de insights e tendências para toda a sua conta e atualiza esse resumo diariamente no console do S3. As métricas desse painel também são resumidas no snapshot da conta na página Buckets. Para obter mais informações, consulte [Painel padrão](#).

Para criar outros painéis e definir o escopo por Regiões da AWS, buckets do S3 ou contas (para o AWS Organizations), crie uma configuração de painel da Lente de Armazenamento do S3. Você pode criar e gerenciar configurações do painel de Lente de Armazenamento do S3 usando o console do Amazon S3, AWS Command Line Interface (AWS CLI), AWS SDKs ou a API REST do Amazon S3. Ao criar ou editar um painel da Lente de Armazenamento do S3, defina o escopo do painel e a seleção de métricas.

A Lente de Armazenamento do S3 oferece métricas gratuitas, bem como métricas e recomendações avançadas que podem ser adquiridas por um custo adicional. Com métricas e recomendações avançadas, você pode acessar métricas e recursos adicionais para obter informações sobre seu armazenamento. Esses recursos incluem categorias de métricas avançadas, agregação de prefixos, recomendações contextuais e publicação no Amazon CloudWatch. A agregação de prefixos e as recomendações contextuais estão disponíveis somente no console do Amazon S3. Para obter mais informações sobre a definição de preço do S3 Storage Lens, consulte [Definição de preço do Amazon S3](#).

Categorias de métricas

Nos níveis gratuito e avançado, as métricas estão organizadas em categorias que se alinham aos principais casos de uso, como otimização de custos e proteção de dados. As métricas gratuitas incluem métricas de resumo, otimização de custos, proteção de dados, gerenciamento de acesso, performance e eventos. Ao fazer upgrade para métricas e recomendações avançadas, você pode ativar métricas avançadas de otimização de custos e proteção de dados. Você pode usar essas métricas avançadas para reduzir ainda mais os custos de armazenamento do S3 e melhorar sua postura de proteção de dados. Você também pode ativar métricas de atividade e métricas de códigos de status detalhados para melhorar a performance das workloads de aplicações que acessam os buckets do S3. Para obter mais informações sobre as categorias de métricas gratuitas e avançadas, consulte [Seleção de métricas](#).

Você pode avaliar seu armazenamento com base nas práticas recomendadas do S3, como analisar a porcentagem de buckets com criptografia ou que têm o Bloqueio de Objeto do S3 ou o Versionamento do S3 habilitado. Você também pode identificar possíveis oportunidades de redução de custos. Por exemplo, você pode usar métricas de contagem de regras do Ciclo de Vida do S3 para identificar buckets que não têm regras de ciclo de vida de validade ou transição. Você também pode analisar sua atividade de solicitações por bucket para encontrar buckets onde os objetos podem ser transferidos para uma classe de armazenamento de menor custo. Para obter mais informações, consulte [Casos de uso das métricas da Lente de Armazenamento do Amazon S3](#).

Exportação de métricas

Além de visualizar o 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 com ferramentas analíticas. Para obter mais informações, consulte [Exibição de métricas de lente de armazenamento do Amazon S3 usando uma exportação de dados](#).

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 da Lente de Armazenamento do S3. Além disso, as operações de API do CloudWatch permitem que aplicações, inclusive provedores de terceiros, acessem as métricas da Lente de Armazenamento do S3. A opção de publicação no CloudWatch está disponível para painéis com upgrade para métricas e recomendações avançadas da Lente de Armazenamento do S3. Para obter mais informações sobre suporte às métricas do S3 Storage Lens no CloudWatch, consulte [Monitorar métricas do S3 Storage Lens no CloudWatch](#).

Para obter mais informações sobre como usar a Lente de Armazenamento do S3, consulte os tópicos a seguir.

Tópicos

- [Noções básicas sobre o Amazon S3 Storage Lens](#)
- [Usar o Amazon S3 Storage Lens com AWS Organizations](#)
- [Permissões do Amazon S3 Storage Lens](#)
- [Visualizar métricas com a Lente de Armazenamento do Amazon S3](#)
- [Casos de uso das métricas da Lente de Armazenamento do Amazon S3](#)
- [Glossário de métricas de lente de armazenamento do Amazon S3](#)
- [Trabalhar com a Lente de Armazenamento do Amazon S3 usando o console e a API](#)
- [Trabalho com grupos de Lente de Armazenamento do S3](#)

Noções básicas sobre o Amazon S3 Storage Lens

Important

O Amazon S3 agora aplica criptografia do lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. Desde 5 de janeiro de 2023, todos os novos uploads de objetos para o Amazon S3 são automaticamente criptografados sem custo adicional e sem impacto na performance. O status de criptografia automática para a configuração de criptografia padrão do bucket do S3 e para novos uploads de objetos está disponível em logs do AWS CloudTrail, no Inventário do S3, na Lente de Armazenamento do S3, no console do Amazon S3 e como cabeçalho adicional de resposta da API do Amazon S3 na AWS Command Line Interface e em AWS SDKs. Para obter mais informações, consulte [Perguntas frequentes sobre criptografia padrão](#).

A Lente de Armazenamento do Amazon S3 é um recurso de análise de armazenamento em nuvem que você pode usar para obter visibilidade em toda a organização sobre o uso e a atividade do armazenamento de objetos. Use as métricas da Lente de Armazenamento do S3 para gerar insights resumidos, como descobrir quanto armazenamento você tem em toda a organização ou quais são os buckets e prefixos que mais crescem. Você também pode usar as métricas da Lente de Armazenamento do S3 para identificar oportunidades de otimização de custos, implementar práticas

recomendadas de proteção e segurança de dados, e melhorar a performance das workloads de aplicações. Por exemplo, você pode identificar buckets que não têm regras de Ciclo de Vida do S3 para expirar multipart uploads incompletos com mais de 7 dias. Você também pode identificar buckets que não estão seguindo as práticas recomendadas de proteção de dados, como usar a Replicação do S3 ou o Versionamento do S3. A Lente de Armazenamento do S3 também analisa as métricas para fornecer recomendações contextuais que você pode usar para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados.

A Lente de Armazenamento do S3 agrega suas métricas e exibe as informações na seção Account snapshot (Snapshot da conta) na página Buckets do console do Amazon S3. A Lente de Armazenamento do S3 também oferece um painel interativo 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. O painel tem opções de detalhamento para gerar e visualizar insights no nível da organização, conta, Região da AWS, classe de armazenamento, bucket, prefixo ou grupo de Lente de Armazenamento. Você também pode enviar uma exportação diária de métricas em CSV ou no formato Parquet para um bucket do S3. Você pode criar e gerenciar painéis de Lente de Armazenamento do S3 usando o console do Amazon S3, AWS Command Line Interface (AWS CLI), AWS SDKs ou API REST do Amazon S3.

Conceitos e terminologia da Lente de Armazenamento do S3

Esta seção contém a terminologia e os conceitos essenciais para entender e usar a Lente de Armazenamento do Amazon S3.

Tópicos

- [Configuração de painel](#)
- [Painel padrão](#)
- [Painéis](#)
- [Snapshot da conta](#)
- [Exportação de métricas](#)
- [Região inicial](#)
- [Período de retenção](#)
- [Categorias de métricas](#)
- [Recomendações](#)
- [Seleção de métricas](#)

- [S3 Storage Lens e AWS Organizations](#)

Configuração de painel

A Lente de Armazenamento do S3 requer uma configuração de painel que contém as propriedades necessárias para agregar métricas em seu nome para um único painel ou exportação. Ao criar uma configuração, escolha o nome do painel e a região inicial, que não poderão ser alterados depois de criar o painel. Opcionalmente, você pode adicionar etiquetas e configurar uma exportação de métricas em formato CSV ou Parquet.

Na configuração do painel, defina também o escopo do painel e a seleção de métricas. O escopo pode incluir todo o armazenamento da conta da sua organização ou seções que são filtradas por região, bucket e conta. Ao configurar a seleção de métricas, escolha entre métricas gratuitas, ou compre métricas e recomendações avançadas por um custo adicional. Com métricas e recomendações avançadas, você tem acesso a métricas e recursos adicionais. Esses recursos incluem categorias de métricas avançadas, agregação por prefixo, recomendações contextuais e publicação no Amazon CloudWatch. Para obter mais informações sobre a definição de preço do S3 Storage Lens, consulte [Definição de preço do Amazon S3](#).

Painel padrão

O painel padrão do S3 Storage Lens no console é chamado default-account-dashboard. O S3 pré-configura esse painel para visualizar um resumo de insights e tendências para toda a sua conta e atualiza esse resumo diariamente no console do S3. Você não pode modificar o escopo de configuração do painel padrão, mas pode atualizar a seleção de métricas gratuitas para métricas e recomendações avançadas. Você pode configurar a exportação opcional de métricas ou até mesmo desabilitar o painel. No entanto, você não pode excluir o painel padrão.

Note

Se você desabilitar o painel padrão, ele deixará de ser atualizado. Você não receberá mais nenhuma métrica diária nova no painel da Lente de Armazenamento do S3, a exportação de métricas ou o snapshot da conta na página Buckets do S3. Se o painel usar métricas e recomendações avançadas, você não receberá mais cobranças. Você ainda pode ver dados históricos no painel durante o período de validade de 14 dias para consultas de dados. Esse período é de 15 meses com a opção de métricas e recomendações avançadas habilitada. Para acessar dados históricos, reative o painel dentro do período de validade.

Painéis

Você pode criar painéis adicionais da Lente de Armazenamento do S3 e definir o escopo deles por Regiões da AWS, bucket do S3 ou conta (para o AWS Organizations). Ao criar ou editar um painel da Lente de Armazenamento do S3, defina o escopo do painel e a seleção de métricas. A Lente de Armazenamento do S3 oferece métricas gratuitas, bem como métricas e recomendações avançadas que podem ser adquiridas por um custo adicional. Com métricas e recomendações avançadas, você pode acessar métricas e recursos adicionais para obter informações sobre seu armazenamento. Elas incluem categorias de métricas avançadas, agregação por prefixo, recomendações contextuais e publicação no Amazon CloudWatch. Para obter mais informações sobre a definição de preço do S3 Storage Lens, consulte [Definição de preço do Amazon S3](#).

Você também pode desativar ou excluir painéis. 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 durante o período de validade de 14 dias. Esse período é de 15 meses com a opção de métricas e recomendações avançadas habilitada para esse painel. Para acessar dados históricos, reative o painel dentro do período de validade.

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.

Note

- Você pode usar o S3 Storage Lens para criar até 50 painéis por região inicial.
- Os painéis por organização só podem ser limitados a um escopo regional.

Snapshot da conta

O Account snapshot (Snapshot da conta) da Lente de Armazenamento do S3 resume as métricas do painel padrão e exibe o armazenamento total, a contagem de objetos e o tamanho médio dos objetos na página Buckets do console do S3. Esse snapshot de conta oferece acesso rápido a insights sobre o armazenamento sem ter que sair da página Buckets. O snapshot da conta também fornece acesso com um clique ao painel interativo da Lente de Armazenamento do S3.

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. O painel tem opções de detalhamento para gerar insights por organização, conta, bucket, objeto ou prefixo. Você também pode enviar um arquivo exportado em formato CSV ou Parquet das métricas uma vez por dia para um bucket do S3.

Não é possível modificar o escopo do default-account dashboard (painel padrão da conta) porque ele está vinculado ao Account snapshot (Snapshot da conta). No entanto, você pode fazer upgrade da seleção de métricas no painel padrão da conta de métricas gratuitas para métricas e recomendações avançadas pagas. Depois de fazer upgrade, você pode exibir todas as solicitações, bytes carregados e bytes baixados no Account snapshot (Snapshot da conta) da Lente de Armazenamento do S3.

Note

Se você desabilitar o painel padrão, o Account snapshot (Snapshot da conta) não será mais atualizado. Para continuar exibidos as métricas no Account snapshot (Snapshot da conta), você pode reativar o painel padrão 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 em formato CSV ou Parquet e enviadas para um bucket do S3. Você pode usar a exportação de métricas para realizar análises adicionais usando a ferramenta de métricas de sua escolha. 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 da Lente de Armazenamento do S3 pelo console do S3 editando a configuração do painel. Você também pode configurar uma exportação de métricas usando a AWS CLI e AWS SDKs.

Região inicial

A região inicial é a Região da AWS em que todas as métricas da Lente de Armazenamento do S3 para determinada configuração de painel são armazenadas. Você deve escolher uma região inicial ao criar uma configuração de painel da Lente de Armazenamento do S3. Depois de escolher uma região inicial, não será possível alterá-la. Além disso, se você estiver criando um grupo de Lente de Armazenamento, recomendamos que você escolha a mesma região inicial do painel de Lente de Armazenamento.

Note

Você pode escolher uma das seguintes regiões como região inicial:

- US East (N. Virginia) – us-east-1
- US East (Ohio) – us-east-2
- US West (N. California) – us-west-1
- US West (Oregon) – us-west-2
- Asia Pacific (Mumbai) – ap-south-1
- Ásia-Pacífico (Seul) – ap-northeast-2
- Asia Pacific (Singapore) – ap-southeast-1
- Ásia-Pacífico (Sydney) – ap-southeast-2
- Ásia-Pacífico (Tóquio) (ap-northeast-1)
- Canada (Central) – ca-central-1
- China (Pequim): cn-north-1
- China (Ningxia): cn-northwest-1
- Europe (Frankfurt) – eu-central-1
- Europe (Ireland) – eu-west-1
- Europe (London) – eu-west-2
- Europe (Paris) – eu-west-3
- Europe (Stockholm) – eu-north-1
- South America (São Paulo) – sa-east-1

Período de retenção

As métricas da Lente de Armazenamento do S3 são mantidas para que você possa ver tendências históricas e comparar diferenças na atividade e no armazenamento ao longo do tempo. 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.

Todas as métricas do S3 Storage Lens são mantidas durante um período de 15 meses. No entanto, as métricas só estão disponíveis para consultas durante um período específico, que depende da sua

[seleção de métricas](#). Essa duração 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.

Categorias de métricas

Nos níveis gratuito e avançado, as métricas da Lente de Armazenamento do S3 estão organizadas em categorias que se alinham aos principais casos de uso, como otimização de custos e proteção de dados. As métricas gratuitas incluem métricas de resumo, otimização de custos, proteção de dados, gerenciamento de acesso, performance e eventos. Ao atualizar para métricas e recomendações avançadas, você pode ativar métricas adicionais de otimização de custos e proteção de dados que podem ser usadas para reduzir ainda mais os custos de armazenamento do S3 e garantir que seus dados estejam protegidos. Você também pode ativar métricas de atividade e métricas de códigos de status detalhados que pode usar para melhorar a performance dos fluxos de trabalho de aplicações.

A lista a seguir mostra todas as categorias de métricas gratuitas e avançadas. Para obter uma lista completa das métricas individuais incluídas em cada categoria, consulte o [Glossário de métricas](#).

Métricas de resumo

As métricas de resumo fornecem informações gerais sobre seu armazenamento do S3, incluindo o total de bytes armazenados e a contagem de objetos.

Métricas de otimização de custos

As métricas de otimização de custos fornecem informações que você pode usar para gerenciar e otimizar seus custos de armazenamento. Por exemplo, você pode identificar buckets com multipart uploads que estão incompletos há mais de 7 dias.

Com as métricas e recomendações avançadas, você tem acesso a métricas avançadas de otimização de custos. Essas métricas incluem métricas de contagem de regras do Ciclo de Vida do S3 que você pode usar para obter contagens de regras do Ciclo de Vida do S3 para validade e transição por bucket.

Métricas de proteção de dados

As métricas de proteção de dados fornecem informações sobre recursos de proteção de dados, como criptografia e Versionamento do S3. Você pode usar essas métricas para identificar buckets que não estão seguindo as práticas recomendadas de proteção de dados. Por exemplo, você pode identificar buckets que não usam criptografia padrão com chaves do AWS Key Management Service (SSE-KMS) ou que não usam o Versionamento do S3.

Com as métricas e recomendações avançadas, você tem acesso a métricas avançadas de proteção de dados. Essas métricas incluem métricas de contagem de regras de replicação por bucket.

Métricas de gerenciamento de acesso

As métricas de gerenciamento de acesso fornecem informações sobre a Propriedade de Objeto do S3. Você pode usar essas métricas para ver as configurações de Propriedade de Objeto de seus buckets.

Métricas de eventos

As métricas de eventos fornecem informações sobre as Notificações de Eventos do S3. Com as métricas de eventos, você pode ver quais buckets têm as Notificações de Eventos do S3 configuradas.

Métricas de performance

As métricas de performance fornecem informações sobre a Aceleração de Transferência do S3. Com as métricas de performance, você pode ver quais buckets têm a Aceleração de Transferência ativada.

Métricas de atividade (avançadas)

Se você atualizar seu painel para métricas e recomendações avançadas, poderá ativar métricas de atividade. As métricas de atividade fornecem detalhes sobre como seu armazenamento é solicitado (por exemplo, Todas as solicitações, Obter solicitações, Colocar solicitações), Bytes carregados ou baixados e erros.

As métricas de atividade em nível de prefixo podem ser usadas para ajudar a determinar quais prefixos estão sendo usados com pouca frequência, a fim de fazer a [transição para uma classe de armazenamento mais ideal](#) usando o S3 Lifecycle.

Métricas de códigos de status detalhados (avançadas)

Se você atualizar seu painel para métricas e recomendações avançadas, poderá ativar métricas detalhadas de código de status. Métricas detalhadas de código de status fornecem informações para códigos de status HTTP, como 403 Forbidden e 503 Service Unavailable, que você pode usar para solucionar problemas de acesso ou desempenho. Por exemplo, você pode observar a métrica 403 Forbidden error count (Contagem de erros 403 Proibido) para identificar workloads que estão acessando buckets sem as permissões corretas em vigor.

Métricas de código de status detalhadas em nível de prefixo podem ser usadas para obter uma melhor compreensão das ocorrências de código de status HTTP por prefixo. Por exemplo, as métricas de contagem de erros 503 permitem identificar prefixos que recebem solicitações de controle de utilização durante a ingestão de dados.

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 as tendências no armazenamento e atividades 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 de Lente de Armazenamento do S3 para especificar mais detalhes sobre regiões, buckets ou prefixos específicos.

- Chamadas

Chamadas são recomendações que alertam sobre anomalias interessantes no armazenamento e atividade ao longo de um período, as quais podem requerer 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. Nessa pontuação, a métrica do dia atual é subtraída da média dos últimos 30 dias dessa métrica. A métrica do dia atual é então dividida pelo desvio padrão dessa métrica nos últimos 30 dias. A pontuação resultante geralmente é entre -3 e +3. Este número representa o número de desvios padrão entre a métrica do dia atual e a 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 está na faixa de +/- 20% em relação ao dia, semana ou mês anterior.

Como lidar com chamadas em seu armazenamento e atividade — se você receber uma chamada de alteração significativa, isso não será necessariamente um problema. A chamada pode ser o resultado de uma mudança prevista no seu 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 ajudar você 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 incluem métricas relevantes para seu armazenamento, como o número de buckets e objetos em sua conta. As métricas gratuitas também incluem métricas baseadas em casos de uso (por exemplo, métricas de otimização de custos e proteção de dados) que você pode usar para investigar se seu armazenamento está configurado de acordo com as práticas recomendadas do S3. Todas as métricas gratuitas são coletadas diariamente. Os dados ficam disponíveis para consultas por 14 dias. Para obter mais informações sobre quais métricas estão disponíveis com métricas gratuitas, consulte o [Glossário de métricas de lente de armazenamento do Amazon S3](#).

- 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 e recomendações avançadas incluem todas as métricas gratuitas e métricas adicionais, como métricas avançadas de proteção de dados e otimização de custos, métricas de atividade e métricas de códigos de status detalhados. As métricas e recomendações avançadas também fornecem recomendações para ajudar a otimizar o armazenamento. As recomendações são colocadas contextualmente ao lado de métricas relevantes no painel.

As métricas e recomendações avançadas incluem os seguintes recursos:

- **Métricas avançadas:** gere métricas adicionais. Para obter uma lista completa das categorias de métricas avançadas, consulte [Categorias de métricas](#). Para obter uma lista completa das métricas, consulte [Glossário de métricas de lente de armazenamento do Amazon S3](#).
- **Publicação do Amazon CloudWatch** — publica métricas de Lente de Armazenamento do S3 no CloudWatch para criar uma visão unificada da sua integridade operacional nos [painéis](#) do CloudWatch. Também é possível usar operações de API e recursos do CloudWatch, como alarmes e ações acionadas, matemática métrica e detecção de anomalias, para monitorar e realizar ações sobre as métricas da Lente de Armazenamento do S3. Para obter mais informações, consulte [Monitorar métricas do S3 Storage Lens no CloudWatch](#).
- **Agregação de prefixo** — coleta métricas no nível do [prefixo](#). A ativação da agregação de prefixo estende todas as métricas incluídas na configuração do painel no nível do prefixo. As métricas são geradas apenas para prefixos que atendem ao limite configurado. Observe que as métricas aplicáveis no nível do prefixo estão disponíveis com agregação de prefixo, exceto para configurações no nível do bucket e métricas de contagem de regras. As métricas por prefixo não são publicadas no CloudWatch.
- **Agregação do grupo de Lente de Armazenamento** — coleta métricas no nível do grupo de Lente de Armazenamento. Depois de ativar Métricas e recomendações avançadas e Agregação de grupos de Lente de Armazenamento, você poderá especificar quais grupos de Lente de Armazenamento incluir ou excluir do painel de Lente de Armazenamento. Pelo menos um grupo de Lente de Armazenamento deve ser especificado. Os grupos de Lente de Armazenamento especificados também devem residir na região inicial designada na conta do painel. As métricas em nível de grupo de Lente de Armazenamento não são publicadas no CloudWatch.

Todas as métricas avançadas são coletadas diariamente. Os dados ficam disponíveis para consulta por até 15 meses. Para obter mais informações sobre as métricas de armazenamento

agregadas pela Lente de Armazenamento do S3, consulte [Glossário de métricas de lente de armazenamento do Amazon S3](#).

 Note

As recomendações só estão disponíveis quando você usa o painel da Lente de Armazenamento do S3 no console do Amazon S3.

S3 Storage Lens e AWS Organizations

O AWS Organizations é um AWS service (Serviço da AWS) que ajuda você a agregar todas as suas Contas da AWS em uma hierarquia da organização. A Lente de Armazenamento do Amazon S3 funciona com o AWS Organizations para fornecer uma visão única do armazenamento de objetos e da atividade em todo o armazenamento do Amazon S3.

Para obter mais informações, consulte [Usar o Amazon S3 Storage Lens com AWS Organizations](#).

- Acesso confiável

Usando a conta de gerenciamento da sua organização, você deve habilitar o acesso confiável para a Lente de Armazenamento do S3 a fim de agregar métricas de armazenamento e dados de uso para todas as contas membros da sua organização. Depois, você pode criar painéis ou exportações para sua organização usando sua conta de gerenciamento ou dando acesso de 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 a Lente de Armazenamento do S3 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 cancelamento do registro de um administrador delegado também impede automaticamente que todos os painéis da organização criados por esse administrador delegado agreguem novas métricas de armazenamento.

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 um serviço é um tipo exclusivo de função do IAM vinculada diretamente ao S3 Storage Lens. As funções vinculadas ao serviço são predefinidas pela Lente de Armazenamento do S3 e incluem todas as permissões necessárias para coletar métricas diárias de armazenamento e atividades de contas de membros na 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

A Lente de Armazenamento do Amazon S3 é um recurso de análise de armazenamento em nuvem que você pode usar para obter visibilidade em toda a organização sobre o uso e a atividade do armazenamento de objetos. Use as métricas da Lente de Armazenamento do S3 para gerar insights resumidos, como descobrir quanto armazenamento você tem em toda a organização ou quais são os buckets e prefixos que mais crescem. Você também pode usar as métricas da Lente de Armazenamento do S3 para identificar oportunidades de otimização de custos, implementar práticas recomendadas de proteção e segurança de dados, e melhorar a performance das workloads de aplicações. Por exemplo, você pode identificar buckets que não têm regras de Ciclo de Vida do S3 para expirar multipart uploads incompletos com mais de 7 dias. Você também pode identificar buckets que não estão seguindo as práticas recomendadas de proteção de dados, como usar a Replicação do S3 ou o Versionamento do S3. A Lente de Armazenamento do S3 também analisa as métricas para fornecer recomendações contextuais que você pode usar para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados.

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 da Lente de Armazenamento do S3 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](#)
- [Desativação do acesso confiável para o S3 Storage Lens](#)
- [Registro de um administrador delegado para o S3 Storage Lens](#)
- [Cancelamento do registro de um administrador delegado para o S3 Storage Lens](#)

Ativando acesso confiável para o S3 Storage Lens

Ao habilitar o acesso confiável, você permite que a Lente de Armazenamento do Amazon S3 tenha acesso à sua hierarquia do AWS Organizations, associação e estrutura por meio das operações de API do AWS Organizations. A Lente de Armazenamento do S3 se torna um serviço confiável para toda a estrutura de sua organização.

Sempre que uma configuração de painel é criada, a Lente de Armazenamento do S3 cria funções vinculadas a serviços nas contas de gerenciamento ou administrador delegado da sua organização. A função vinculada ao serviço concede permissão à de Lente de Armazenamento do S3 para:

- Descrever organizações
- Listar contas
- Verificar uma lista de acesso AWS service (Serviço da AWS) para as organizações
- Obter administradores delegados para as organizações

A Lente de Armazenamento do S3 pode garantir que tenha acesso para coletar as métricas entre contas de 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 operações de API da organização somente leitura. Esse acesso oferece a visibilidade de administrador delegado aos membros e estruturas de sua organização para que também possam criar painéis da Lente de Armazenamento do S3.

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 de conta pode ver apenas as informações da Lente de Armazenamento do S3 para o escopo de sua conta e não para a organização inteira. Os painéis que exigem acesso confiável não são mais atualizados, mas reterão seus dados históricos de acordo com o período de [disponibilidade dos dados para consulta](#).

Note

- A desativação do acesso confiável para a Lente de Armazenamento do S3 também impede automaticamente que todos os painéis da organização colem e agreguem métricas de armazenamento.
- Suas contas de gerenciamento e administrador delegado ainda poderão ver os dados históricos dos painéis da organização durante o período de disponibilidade dos dados 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 delegado. 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](#).

Você também pode registrar um administrador delegado usando a API REST do AWS Organizations, a 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, a 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](#).

Você também pode cancelar o registro de um administrador delegado usando a API REST do AWS Organizations, a AWS CLI ou SDKs da conta de gerenciamento. Para obter mais informações, consulte [DeregisterDelegatedAdministrator](#) na Referência da API do AWS Organizations.

Note

- O cancelamento do registro de um administrador delegado também impede automaticamente que todos os painéis da organização criados por esse administrador delegado agreguem novas métricas de armazenamento.
- O administrador delegado que tiver o registro cancelado ainda poderá 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. Para conceder essas permissões, você pode usar uma política do IAM baseada em identidade. Você pode vincular essa política a usuários, grupos ou funções do IAM para conceder permissões. Essas permissões podem incluir a capacidade de ativar ou desativar a Lente de Armazenamento do S3 ou acessar qualquer painel ou configuração da Lente de Armazenamento do S3.

O usuário ou a função do IAM deve pertencer à conta que criou ou tem o painel ou a configuração, a menos que as duas condições a seguir sejam verdadeiras:

- Sua conta é membro do AWS Organizations.
- Você recebeu 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 raiz da sua conta para visualizar os painéis da Lente de Armazenamento do Amazon S3. Para acessar os painéis da Lente de Armazenamento do S3, 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 de segurança no IAM](#) no Guia do usuário do IAM.
- 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](#)
- [Definição de permissões de conta para usar a Lente de Armazenamento do S3](#)
- [Configuração de permissões para usar o Amazon S3 Storage Lens com AWS Organizations](#)

Definição de permissões de conta para usar o S3 Storage Lens

Para criar e gerenciar painéis de Lente de Armazenamento do S3 e configurações de painel de Lente de Armazenamento, você deve ter as seguintes permissões, dependendo de quais ações deseja executar:

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.	s3:ListStorageLensConfigurations s3:GetStorageLensConfiguration s3:GetStorageLensConfigurat ionTagging s3:PutStorageLensConfiguration s3:PutStorageLensConfigurat ionTagging
Obter as etiquetas de um painel da Lente de Armazenamento do S3 no console do Amazon S3.	s3:ListStorageLensConfigurations s3:GetStorageLensConfigurat ionTagging
Veja um painel do S3 Storage Lens no console do Amazon S3.	s3:ListStorageLensConfigurations s3:GetStorageLensConfiguration s3:GetStorageLensDashboard
Exclua um painel do S3 Storage Lens no console do Amazon S3.	s3:ListStorageLensConfigurations s3:GetStorageLensConfiguration s3>DeleteStorageLensConfigu ration
Criar ou atualizar uma configuração da Lente de Armazenamento do S3 usando a AWS CLI ou um AWS SDK.	s3:PutStorageLensConfiguration s3:PutStorageLensConfigurat ionTagging
Obter as etiquetas de uma configuração da Lente de Armazenamento do S3 usando a AWS CLI ou um AWS SDK.	s3:GetStorageLensConfigurat ionTagging

Ação	Permissões do IAM
Visualizar uma configuração da Lente de Armazenamento do S3 usando a AWS CLI ou um AWS SDK.	<code>s3:GetStorageLensConfiguration</code>
Excluir uma configuração da Lente de Armazenamento do S3 usando a AWS CLI ou um AWS SDK.	<code>s3:DeleteStorageLensConfiguration</code>

Note

- Você pode usar tags de recurso em uma política do IAM para gerenciar permissões.
- Um usuário ou perfil do IAM com essas permissões pode ver métricas de buckets e prefixos dos quais pode não ter permissão direta para ler ou listar objetos.
- No caso de painéis de Lente de Armazenamento do S3 com métricas em nível de prefixo habilitadas, se um caminho de prefixo selecionado corresponder a uma chave de objeto, o painel poderá exibir a chave de objeto como outro prefixo.
- Para as exportações de métricas, que são 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 as contas de administrador delegado podem usar políticas do IAM para gerenciar as permissões de acesso para configurações e painéis no nível da organização.

Definição de permissões de conta para usar a Lente de Armazenamento do S3

Você pode usar grupos de Lente de Armazenamento do S3 para entender a distribuição do seu armazenamento dentro de buckets com base no prefixo, sufixo, tag do objeto, tamanho do objeto ou idade do objeto. Você pode anexar grupos de Lente de Armazenamento aos seus painéis para ver suas métricas agregadas.

Para trabalhar com grupos de Lente de Armazenamento, você precisa de determinadas permissões. Para obter mais informações, consulte [the section called “Permissões de grupos de Lente de Armazenamento”](#).

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 Organizations Permissões 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.	<code>organizations:EnableAWSServiceAccess</code>
Desative o acesso confiável para a Lente de Armazenamento do S3 em sua organização.	<code>organizations:DisableAWSServiceAccess</code>
Registre um administrador delegado para criar painéis ou configurações do S3 Storage Lens para sua organização.	<code>organizations:RegisterDelegatedAdministrator</code>
Cancele o registro de um administrador delegado para que ele não possa mais criar painéis ou configurações da Lente de Armazenamento do S3 em sua organização.	<code>organizations:DeregisterDelegatedAdministrator</code>
Permissões adicionais para criar configurações para toda a organização da Lente de Armazenamento do S3.	<code>organizations:DescribeOrganization</code> <code>organizations:ListAccounts</code> <code>organizations:ListAWSServiceAccessForOrganization</code> <code>organizations:ListDelegatedAdministrators</code> <code>iam:CreateServiceLinkedRole</code>

Visualizar métricas com a Lente de Armazenamento do Amazon S3

A Lente de Armazenamento do S3 agrega suas métricas e exibe as informações na seção Account snapshot (Snapshot da conta) na página Buckets do console do Amazon S3. A Lente de Armazenamento do S3 também oferece um painel interativo 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. O painel tem opções de detalhamento para gerar e visualizar insights no nível da organização, conta, Região da AWS, classe de armazenamento, bucket, prefixo ou grupo de Lente de Armazenamento. Você também pode enviar uma exportação diária de métricas em CSV ou no formato Parquet para um bucket do S3.

Por padrão, todos os painéis são configurados com métricas gratuitas, que incluem métricas que você pode usar para entender o uso e a atividade em seu armazenamento do S3, otimizar seus custos de armazenamento e implementar as práticas recomendadas de proteção de dados e gerenciamento de acesso. As métricas gratuitas são agregadas até o nível de bucket. Com métricas gratuitas, os dados ficam disponíveis para consultas por até 14 dias.

As métricas e recomendações avançadas incluem os recursos adicionais a seguir, que você pode usar para obter mais informações sobre o uso e a atividade em seu armazenamento e práticas recomendadas para otimizar o armazenamento:

- Recomendações contextuais (disponíveis apenas no painel)
- Métricas avançadas (incluindo métricas de atividade agregadas por bucket)
- Agregação de prefixo
- Agregação de grupos de Armazenamento de Lente
- Agregação de grupos de Armazenamento de Lente
- Publicação no Amazon CloudWatch

Os dados de métricas avançadas 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](#). Para obter mais informações sobre métricas gratuitas e avançadas, consulte [Seleção de métricas](#).

Tópicos

- [Exibição das métricas da lente de armazenamento do S3 nos painéis](#)

- [Exibição de métricas de lente de armazenamento do Amazon S3 usando uma exportação de dados](#)
- [Monitorar métricas do S3 Storage Lens no CloudWatch](#)

Exibição das métricas da lente de armazenamento do S3 nos painéis

No console do Amazon S3, a Lente de Armazenamento do S3 fornece um painel interativo padrão que você pode usar para ver insights e tendências em seus dados. Você também pode usar esse painel para sinalizar valores discrepantes e receber recomendações para otimizar custos de armazenamento e aplicar práticas recomendadas de proteção de dados. Seu painel tem opções de detalhamento para gerar insights em nível de conta, bucket, Região da AWS, prefixo ou grupo de Lente de Armazenamento. Se você tiver habilitado a Lente de Armazenamento do S3 para funcionar com o AWS Organizations, também poderá gerar insights no nível da organização (como dados de todas as contas que fazem parte da hierarquia AWS Organizations). O painel sempre carrega a data mais recente para a qual há métricas disponíveis.

O painel padrão do S3 Storage Lens no console é chamado default-account-dashboard. O Amazon S3 pré-configura esse painel para visualizar insights e tendências resumidos de toda a sua conta e os atualiza diariamente no console do S3. Você não pode modificar o escopo de configuração do painel padrão, mas pode atualizar a seleção de métricas gratuitas para métricas e recomendações avançadas pagas. Com métricas e recomendações avançadas, você tem acesso a métricas e recursos adicionais. Esses recursos incluem categorias de métricas avançadas, agregação por prefixo, recomendações contextuais e publicação no Amazon CloudWatch.

É possível desabilitar o painel padrão, mas não é possível excluí-lo. Se você desabilitar o painel padrão, ele deixará de ser atualizado. Você também não receberá mais novas métricas diárias na Lente de Armazenamento do S3 ou na seção Instantâneo da conta na página Buckets. Você ainda pode ver dados históricos no painel padrão até o fim do período de 14 dias para consultas de dados. Esse período é de 15 meses com a opção de métricas e recomendações avançadas habilitada. Para acessar esses dados, você pode reativar o painel padrão antes do fim do período de validade.

Você pode criar painéis adicionais da Lente de Armazenamento do S3 e definir seu escopo por Regiões da AWS, buckets do S3 ou contas. Você também poderá definir o escopo dos seus painéis por organização se tiver habilitado a Lente de Armazenamento para funcionar com o AWS Organizations. Ao criar ou editar um painel da Lente de Armazenamento do S3, defina o escopo do painel e a seleção de métricas.

Você pode desativar ou excluir qualquer painel adicional criado por você.

- 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 para métricas gratuitas durante o período de validade de 14 dias. Esse período é de 15 meses com a opção de métricas e recomendações avançadas habilitada para esse painel. Para acessar esses dados, você pode reativar o painel antes do fim do período de validade.
- 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.

Tópicos

- [Exibição de um painel do Amazon S3 Storage Lens](#)
- [Noções básicas sobre seu painel do S3 Storage Lens](#)

Exibição de um painel do Amazon S3 Storage Lens

O procedimento a seguir mostra como visualizar um painel da Lente de Armazenamento do S3 no console do S3. Para obter orientações baseadas em casos de uso que mostram como usar o painel para otimizar custos, implementar práticas recomendadas e melhorar o desempenho de aplicativos que acessam seus buckets S3, consulte [Casos de uso das métricas da Lente de Armazenamento do Amazon S3](#).

Note


Você não pode usar as credenciais de usuário raiz da sua conta para visualizar os painéis da Lente de Armazenamento do Amazon S3. Para acessar os painéis da Lente de Armazenamento do S3, você deve conceder as permissões do AWS Identity and Access Management (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](#) e [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

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 Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o painel que deseja visualizar.

Seu painel é aberto na Lente de Armazenamento do S3. A seção Snapshot for date (Snapshot para data) mostra a data mais recente para a qual a Lente de Armazenamento do S3 coletou métricas. Seu painel sempre carrega a data mais recente para a qual há métricas disponíveis.

4. (Opcional) Para alterar a data do painel da Lente de Armazenamento do S3, no seletor de datas que fica no canto superior direito, escolha uma nova data.
5. (Opcional) Para aplicar filtros temporários com o intuito de limitar ainda mais o escopo dos dados do painel, faça o seguinte:
 - a. Expanda a seção Filtros.
 - b. Para filtrar por contas específicas, Regiões da AWS, classes de armazenamento, buckets, prefixos ou grupos de Lente de Armazenamento, escolha as opções pelas quais filtrar.

 Note

Os filtros Prefixos e Grupos de Lente de Armazenamento não podem ser aplicados ao mesmo tempo.

- c. Para atualizar um filtro, escolha Apply (Aplicar).
 - d. Para remover um filtro, clique no X ao lado dele.
6. Em qualquer seção do painel da Lente de Armazenamento do S3, para ver os dados de uma métrica específica, escolha o nome da métrica em Metric (Métrica).
 7. Em qualquer gráfico ou visualização no painel de Lente de Armazenamento do S3, você pode detalhar níveis mais aprofundados de agregação usando as guias Contas, Regiões da AWS, Classes de armazenamento, Buckets, Prefixos ou Grupos de Lente de Armazenamento. Para ver um exemplo, consulte [Descubra buckets inativos do Amazon S3](#).

Noções básicas sobre seu painel do S3 Storage Lens

O painel da Lente de Armazenamento do S3 tem uma guia Overview (Visão geral) primária e até cinco guias adicionais que representam cada nível de agregação:

- Contas

- Regiões da AWS
- Classes de armazenamento
- Buckets
- Prefixos
- Grupos de Lente de Armazenamento

Na guia Overview (Visão geral), os dados do painel são agregados em três seções diferentes: Snapshot for date (Snapshot for data), Trends and distributions (Tendências e distribuições) e Top N overview (Visão geral dos N principais).

Para obter mais informações sobre a Lente de Armazenamento do S3, consulte as seções a seguir.

Snapshot

A seção Snapshot for date (Snapshot para data) mostra o resumo das métricas que a Lente de Armazenamento do S3 agregou para a data selecionada. Esse resumo inclui as seguintes métricas:

- Armazenamento total — A quantidade total de armazenamento usado em bytes.
- Contagem de objetos — O número total de objetos no seu Conta da AWS.
- Tamanho médio do objeto — O tamanho médio do objeto.
- Buckets ativos — O número total de buckets ativos em uso ativo com armazenamento acima de 0 bytes em sua conta.
- Contas — O número de contas cujo armazenamento está no escopo. Esse valor é 1, a menos que você esteja usando o AWS Organizations, e a Lente de Armazenamento do S3 tenha acesso confiável com uma função vinculada a serviço válida. Para obter mais informações, consulte [Usar funções vinculadas a serviços para o Amazon S3 Storage Lens](#).
- Buckets — O número total de buckets em sua conta.

Dados de métricas

Para cada métrica que aparece no snapshot, você pode ver os seguintes dados:

- Nome da métrica — O nome da métrica.
- Categoria de métrica — A categoria em que a métrica está organizada.
- Total por data — A contagem total para a data selecionada.
- % de alteração — A alteração percentual desde a data do último instantâneo.

- Tendência de 30 dias — Uma linha de tendência que mostra as alterações da métrica durante um período de 30 dias.
- Recomendação — Uma recomendação contextual baseada nos dados fornecidos no instantâneo. As recomendações estão disponíveis com as métricas e recomendações avançadas. Para obter mais informações, consulte [Recomendações](#).

Categorias de métricas

Opcionalmente, você pode atualizar a seção Snapshot for date (Snapshot para data) do painel para exibir métricas para outras categorias. Se você quiser ver dados de snapshot para métricas adicionais, poderá escolher entre as seguintes Metrics categories (Categorias de métricas):

- Otimização de custo
- Proteção de dados
- Atividade (disponível com métricas avançadas)
- Gerenciamento de acesso
- Desempenho
- Eventos

A seção Snapshot for date (Snapshot para data) exibe somente uma seleção de métricas para cada categoria. Para ver todas as métricas de uma categoria específica, escolha a métrica nas seções Trends and distributions (Tendências e distribuições) ou Top N overview (Visão geral dos N principais). Para obter mais informações sobre as categorias de métricas, consulte [Categorias de métricas](#). Para obter uma lista completa de métricas da Lente de Armazenamento do S3, consulte [Glossário de métricas de lente de armazenamento do Amazon S3](#).

Tendências e distribuições

A segunda seção da guia Overview (Visão geral) é Trends and distributions (Tendências e distribuições). Na seção Trends and distributions (Tendências e distribuições), você pode escolher duas métricas para comparar em um intervalo de datas definido por você. A seção Trends and distributions (Tendências e distribuições) mostra a relação entre duas métricas ao longo do tempo. Essa seção exibe gráficos que podem ser usados para ver a distribuição de Storage classe (Classe de armazenamento) e Region (Região) entre as duas tendências que você está monitorando. Opcionalmente, você pode detalhar um ponto de dados em um dos gráficos para realizar uma análise mais profunda.

Para acessar uma demonstração que usa a seção Trends and distributions (Tendências e distribuições), consulte [Identificar buckets que não usam a criptografia do lado do servidor com AWS KMS como criptografia padrão \(SSE-KMS\)](#).

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). Esta seção mostra as métricas selecionadas no número principal de contas, Regiões da AWS, buckets, prefixos ou grupos de Lente de Armazenamento. Se você tiver ativado a Lente de Armazenamento do S3 para funcionar com o AWS Organizations, também poderá ver as métricas selecionadas em toda a sua organização.

Para acessar uma demonstração que usa a seção Top N overview (Visão geral dos N principais), consulte [Identifique os seus maiores buckets do S3](#).

Opções de detalhamento e análise

Para fornecer uma experiência de análise fluida, o painel de Lente de Armazenamento do S3 fornece um menu de ação, que aparece quando você escolhe qualquer valor do gráfico. Para usar esse menu, escolha qualquer valor de gráfico para ver os valores das métricas associadas e escolha entre duas opções na caixa exibida:

- A ação Drill down (Busca detalhada) 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 Dimensão selecionada e aplica esse valor da guia como um filtro. Essas guias incluem Contas, Regiões da AWS, Classes de armazenamento, Buckets, Prefixos (para painéis com Métricas avançadas e Agregação de prefixo habilitadas), e Grupos de Lente de Armazenamento (para painéis com Métricas avançadas e Agregação de grupos de Lente de Armazenamento habilitadas). Com Analisar por, você pode ver os dados no contexto da nova dimensão para uma análise mais aprofundada.

As ações Busca detalhada e Analisar por poderão ser desativadas se o resultado produzir resultados ilógicos ou não tiver valor. As ações Busca detalhada e Analisar por aplicam filtros sobre quaisquer filtros existentes em todas as guias do painel. Você também pode remover os filtros conforme necessário.

Guias

As guias no âmbito de dimensão fornecem uma exibição detalhada de todos os valores dentro de determinada dimensão. Por exemplo, a guia Regiões da AWS mostra métricas para todas as

Regiões da AWS, e a guia Buckets mostra métricas para todos os buckets. Cada guia de dimensão contém um layout idêntico composto por quatro seções:

- Um gráfico de tendências que exibe os N itens principais da dimensão nos últimos 30 dias para a métrica selecionada. Por padrão, este gráfico exibe os 10 principais itens, mas você pode diminuí-lo para pelo menos 3 itens ou aumentá-lo para até 50 itens.
- Um gráfico de histograma que mostra um gráfico de barras verticais para a data e métrica selecionadas. Talvez seja necessário rolar horizontalmente se tiver um número grande de itens a serem exibidos neste gráfico.
- Um gráfico de análise de bolhas que representa todos os itens dentro da dimensão. Esse gráfico representa a primeira métrica no eixo x e a segunda métrica no eixo y. A terceira métrica é representada pelo tamanho da bolha.
- Uma exibição de grade métrica que contém cada item da dimensão listado em linhas. As colunas representam cada métrica disponível, organizadas em guias de categoria de métricas para facilitar a navegação.

Exibição de métricas de lente de armazenamento do Amazon S3 usando uma exportação de dados

As métricas da Lente de Armazenamento do Amazon S3 são geradas diariamente em arquivos de exportação de métricas em formato CSV ou Apache Parquet 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](#)
- [O que é um manifesto de exportação do S3 Storage Lens?](#)
- [Compreendendo o esquema de exportação do Amazon S3 Storage Lens](#)

Usar uma AWS KMS key para criptografar suas exportações de métricas

Para conceder permissão à Lente de Armazenamento do Amazon S3 para criptografar as exportações de métricas usando uma chave gerenciada pelo cliente, é necessário usar uma política de chaves. Para atualizar a política de chaves a fim de poder usar uma chave do KMS para criptografar as exportações de métricas da Lente de Armazenamento do S3, siga estas etapas.

Como conceder permissões à Lente de Armazenamento do S3 para 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 esquerdo, 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. Para usar essa política, substitua os *user input placeholders* por suas próprias informações.

```
{
  "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. Escolha Salvar alterações.

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 operação de API PUT de política de chaves do AWS KMS ([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, stor
  "reportFiles": [
    {
      "key": "DestinationPrefix/StorageLens/123456789012/my-dashboard-
configuration-id/V_1/reports/dt=2020-11-03/a38f6bc4-2e3d-4355-ac8a-e2fdcf3de158.csv",
      "size": 1603959,
      "md5Checksum": "2177e775870def72b8d84febe1ad3574"
    }
  ]
}
```

Veja a seguir um exemplo de manifesto em um arquivo `manifest.json` para uma exportação 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 configuration_id da configuração da Lente de Armazenamento do S3.
ReportDate	String	report_date	A data em que as métricas foram monitoradas.
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.
StorageClass	String	storage_class	A classe de armazenamento do bucket em questão.
RecordType	ENUM	record_type	O tipo de artefato que está sendo relatado

Nome do atributo	Tipo de dados	Nome da coluna	Descrição
			(CONTA, BUCKET ou PREFIXO).
RecordValue	String	record_value	O valor do artefato RecordType e . <div data-bbox="1187 512 1510 831" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E1F5FE;"> <p> Note</p> <p>O record_value é codificado em URL.</p> </div>
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.

Note

Você pode identificar métricas para grupos da Lente de Armazenamento procurando os valores STORAGE_LENS_GROUP_BUCKET ou STORAGE_LENS_GROUP_ACCOUNT na coluna record_type. A coluna record_value exibirá o nome do recurso da Amazon

(ARN) para o grupo da Lente de Armazenamento, por exemplo, `arn:aws:s3:us-east-1:123456789012:storage-lens-group/slg-1`.

Table with 10 columns: version_number, configuration_id, report_date, aws_account_number, aws_region, storage_class, record_type, record_value, bucket_name, metric_name, metric_value. It lists various metrics for storage lens groups across different configurations and dates.

Veja a seguir um exemplo de exportação de métricas da Lente de Armazenamento do S3 com dados de grupos da Lente de Armazenamento.

Table with 10 columns: version_number, configuration_id, report_date, aws_account_number, aws_region, storage_class, record_type, record_value, bucket_name, metric_name, metric_value. This table provides a more detailed view of storage lens metrics, including specific bucket names and lens group details.

Monitorar métricas do S3 Storage Lens no CloudWatch

É possível publicar métricas da Lente de Armazenamento do S3 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 da Lente de Armazenamento do S3. Além disso, as operações de API do CloudWatch permitem que aplicações, inclusive provedores de terceiros, acessem as métricas da Lente de Armazenamento do S3. 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 painéis novos 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 da Lente de Armazenamento do S3, consulte [Definição de preço do Amazon S3](#). Não são aplicadas cobranças adicionais para publicação de métricas no CloudWatch, mas outras cobranças do CloudWatch, como painéis, alarmes e chamadas de API, são aplicáveis. Para obter mais informações, consulte [Preço 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 por 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](#): 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, com equipes, com partes interessadas e com pessoas externas às suas organizações.

- [Alarmes e ações acionadas](#): 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 a métrica Incomplete Multipart Upload Bytes (Bytes de multipart uploads incompletos) exceder 1 GB por três dias consecutivos.
- [Detecção de anomalias](#): 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 na métrica Object Lock Enabled Bytes (Bytes de Bloqueio de Objeto ativado) para detectar a remoção não autorizada das configurações de Bloqueio de Objeto.
- [Matemática métrica](#): 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 métrica para obter o tamanho médio de objetos 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](#)
- [Habilitar publicação no CloudWatch para o S3 Storage Lens](#)
- [Trabalhar com métricas do S3 Storage Lens no CloudWatch](#)

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. Com as métricas avançadas habilitadas, é possível usar os [painéis do CloudWatch](#) para monitorar as métricas da Lente de Armazenamento do S3 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 por 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](#)
- [Dimensões](#)

Métricas

As métricas da Lente de Armazenamento do S3 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 por 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](#).

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

Quando a Lente de Armazenamento do S3 envia dados para o CloudWatch, dimensões são anexadas 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 da Lente de Armazenamento do S3, 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 de bucket e conta. Para obter mais informações, consulte [Filtrar métricas usando dimensões](#).

Para ver quais dimensões estão disponíveis para sua configuração do S3 Storage Lens, consulte a tabela a seguir.

Dimensão	Descrição	Bucket	Conta	Organização	Organização e bucket	Organização e conta
<code>configuration_id</code>	O nome do painel para a configuração da Lente de Armazenamento do S3 relatada nas métricas
<code>metrics_version</code>	A versão das métricas da Lente de Armazenamento do S3. A versão das métricas tem um valor fixo de 1.0.
<code>organization_id</code>	O ID do AWS Organizations das métricas
<code>aws_account_number</code>	A Conta da AWS associada às métricas.
<code>aws_region</code>	A Região da AWS das métricas
<code>bucket_name</code>	O nome do bucket do S3 relatado nas métricas
<code>storage_class</code>	A classe de armazenamento do bucket relatado nas métricas
<code>record_type</code>	O detalhamento das métricas: ORGANIZATION, ACCOUNT, BUCKET

Habilitar publicação no CloudWatch para o S3 Storage Lens

É possível publicar métricas da Lente de Armazenamento do S3 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 da Lente de Armazenamento do S3. Além disso, as operações de API do CloudWatch permitem que aplicações, inclusive provedores de terceiros, acessem as métricas da Lente de Armazenamento do S3. 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 por 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 painéis novos 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 com upgrade para métricas e recomendações avançadas da Lente de Armazenamento do S3. Para saber mais sobre o preço das métricas e recomendações avançadas da Lente de Armazenamento do S3, consulte [Definição de preço do Amazon S3](#). Não são aplicadas cobranças adicionais para publicação de métricas no CloudWatch, mas outras cobranças do CloudWatch, como painéis, alarmes e chamadas de 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. No momento, as métricas da Lente de Armazenamento do S3 não podem ser consumidas por fluxos do CloudWatch.

Uso do console do S3

Ao atualizar um painel da Lente de Armazenamento do S3, não é possível alterar o nome do painel nem a região inicial. Também não é possível alterar o escopo do painel padrão, que está definido como o armazenamento de toda a sua conta.

Como atualizar um painel da Lente de Armazenamento do S3 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 esquerdo, escolha S3 Storage Lens (Lente de Armazenamento do S3) e Dashboards (Painéis).
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 por 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 Save changes (Salvar alterações).

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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).
3. Escolha Create dashboard (Criar painel).
4. Em General (Geral), defina as seguintes 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 à sua conta conectada).

 Note

Ao criar uma configuração no nível da organização que inclui todas as contas, não será possível incluir ou excluir buckets, apenas regiões.

- b. Escolha as regiões e os buckets que deseja que a Lente de Armazenamento do S3 inclua na configuração do painel fazendo o seguinte:

- 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 Advanced metrics and recommendations features (Recursos de métricas e recomendações avançadas), selecione as opções que deseja habilitar:
 - Advanced metrics (Métricas avançadas)
 - 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

Note

Para obter mais informações sobre os recursos de métricas e recomendações avançadas, consulte [Seleção de métricas](#).

9. Se você ativou Advanced metrics (Métricas avançadas), selecione as Advanced metrics categories (Categorias de métricas avançadas) que deseja exibir no painel da Lente de Armazenamento do S3:

- Métricas de atividade
- Detailed status code metrics (Métricas detalhadas do código de status)
- Advanced cost optimization metrics (Métricas avançadas de otimização de custos)
- Advanced data protection metrics (Métricas avançadas de proteção de dados)

Para obter mais informações sobre categorias de métricas, consulte [Categorias de métricas](#). Para obter uma lista completa de métricas, consulte [Glossário de métricas de lente de armazenamento do Amazon S3](#).

10. (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](#).

11. Escolha Create dashboard (Criar painel).

Usar a 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 da Lente de Armazenamento do S3. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control put-storage-lens-configuration --account-id=555555555555 --config-id=your-configuration-id --region=us-east-1 --storage-lens-configuration=file:///./config.json

config.json
{
```

```

    "Id": "SampleS3StorageLensConfiguration", //Use this property to identify your S3
Storage Lens configuration.
    "AwsOrg": { //Use this property when enabling S3 Storage Lens for AWS Organizations.
      "Arn": "arn:aws:organizations::123456789012:organization/o-abcdefgh"
    },
    "AccountLevel": {
      "ActivityMetrics": {
        "IsEnabled":true
      },
      "AdvancedCostOptimizationMetrics": {
        "IsEnabled":true
      },
      "AdvancedDataProtectionMetrics": {
        "IsEnabled":true
      },
      "DetailedStatusCodesMetrics": {
        "IsEnabled":true
      },
    },
    "BucketLevel": {
      "ActivityMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
      },
      "ActivityMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
      },
      "AdvancedCostOptimizationMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
      },
      "DetailedStatusCodesMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
      },
    },
    "PrefixLevel":{
      "StorageMetrics":{
        "IsEnabled":true, //Mark this as false if you want only free metrics.
        "SelectionCriteria":{
          "MaxDepth":5,
          "MinStorageBytesPercentage":1.25,
          "Delimiter":"/
        }
      }
    }
  },
  "Exclude": { //Replace with "Include" if you prefer to include Regions.

```

```

"Regions": [
  "eu-west-1"
],
"Buckets": [ //This attribute is not supported for AWS Organizations-level
configurations.
  "arn:aws:s3:::source_bucket1"
]
},
"IsEnabled": true, //Whether the configuration is enabled
"DataExport": { //Details about the metrics export
  "S3BucketDestination": {
    "OutputSchemaVersion": "V_1",
    "Format": "CSV", //You can add "Parquet" if you prefer.
    "AccountId": "111122223333",
    "Arn": "arn:aws:s3:::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": {
      "SSE3": {}
    }
  },
  "CloudWatchMetrics": {
    "IsEnabled": true //Mark this as false if you want to export only free metrics.
  }
}
}
}

```

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 Storage Lens
        configuration.
        String awsOrgARN = "arn:aws:organizations::123456789012:organization/o-
        abcdefgh";
        Format exportFormat = Format.CSV;

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                .withDelimiter("/")
                .withMaxDepth(5)
                .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
            PrefixLevelStorageMetrics()
                .withIsEnabled(true)
                .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                .withAdvancedCostOptimizationMetrics(new
            AdvancedCostOptimizationMetrics().withIsEnabled(true))
```

```
        .withAdvancedDataProtectionMetrics(new
AdvancedDataProtectionMetrics().withIsEnabled(true))
        .withDetailedStatusCodesMetrics(new
DetailedStatusCodesMetrics().withIsEnabled(true))
        .withPrefixLevel(new
PrefixLevel().withStorageMetrics(prefixStorageMetrics));
    AccountLevel accountLevel = new AccountLevel()
        .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
        .withAdvancedCostOptimizationMetrics(new
AdvancedCostOptimizationMetrics().withIsEnabled(true))
        .withAdvancedDataProtectionMetrics(new
AdvancedDataProtectionMetrics().withIsEnabled(true))
        .withDetailedStatusCodesMetrics(new
DetailedStatusCodesMetrics().withIsEnabled(true))
        .withBucketLevel(bucketLevel);

    Include include = new Include()
        .withBuckets(Arrays.asList("arn:aws:s3:::bucketName"))
        .withRegions(Arrays.asList("us-west-2"));

    StorageLensDataExportEncryption exportEncryption = new
StorageLensDataExportEncryption()
        .withSSES3(new SSES3());
    S3BucketDestination s3BucketDestination = new S3BucketDestination()
        .withAccountId(exportAccountId)
        .withArn(exportBucketArn)
        .withEncryption(exportEncryption)
        .withFormat(exportFormat)
        .withOutputSchemaVersion(OutputSchemaVersion.V_1)
        .withPrefix("Prefix");
    CloudWatchMetrics cloudWatchMetrics = new CloudWatchMetrics()
        .withIsEnabled(true);
    StorageLensDataExport dataExport = new StorageLensDataExport()
        .withCloudWatchMetrics(cloudWatchMetrics)
        .withS3BucketDestination(s3BucketDestination);

    StorageLensAwsOrg awsOrg = new StorageLensAwsOrg()
        .withArn(awsOrgARN);

    StorageLensConfiguration configuration = new StorageLensConfiguration()
        .withId(configurationId)
        .withAccountLevel(accountLevel)
        .withInclude(include)
        .withDataExport(dataExport)
```

```
        .withAwsOrg(awsOrg)
        .withIsEnabled(true);

    List<StorageLensTag> tags = Arrays.asList(
        new StorageLensTag().withKey("key-1").withValue("value-1"),
        new StorageLensTag().withKey("key-2").withValue("value-2")
    );

    AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
        .withCredentials(new ProfileCredentialsProvider())
        .withRegion(US_WEST_2)
        .build();

    s3ControlClient.putStorageLensConfiguration(new
    PutStorageLensConfigurationRequest()
        .withAccountId(sourceAccountId)
        .withConfigId(configurationId)
        .withStorageLensConfiguration(configuration)
        .withTags(tags)
    );
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

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](#)
- [Trabalhar com métricas do S3 Storage Lens no CloudWatch](#)

Trabalhar com métricas do S3 Storage Lens no CloudWatch

É possível publicar métricas da Lente de Armazenamento do S3 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 da Lente de Armazenamento do S3. Além disso, as operações de API do CloudWatch permitem que aplicações, inclusive provedores de terceiros, acessem as métricas da Lente de Armazenamento do S3. 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 painéis novos 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 com upgrade para métricas e recomendações avançadas da Lente de Armazenamento do S3. Para saber mais sobre o preço das métricas e recomendações avançadas da Lente de Armazenamento do S3, consulte [Definição de preço do Amazon S3](#). Não são aplicadas cobranças adicionais para publicação de métricas no CloudWatch, mas outras cobranças do CloudWatch, como painéis, alarmes e chamadas de API, são aplicáveis. Para obter mais informações, consulte [Preço 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 por 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.

No momento, as métricas da Lente de Armazenamento do S3 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](#)
- [Definir alarmes, acionar ações e usar detecção de anomalias](#)
- [Filtrar métricas usando dimensões](#)
- [Calcular novas métricas com matemática métrica](#)
- [Usar expressões de pesquisa em gráficos](#)

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 a métrica Incomplete Multipart Upload Bytes (Bytes de multipart uploads incompletos) exceder 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 na métrica Object Lock Enabled Bytes (Bytes de Bloqueio de Objeto ativado) para detectar a remoção não autorizada das configurações de Bloqueio de Objeto.

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 bucket 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 filtragem por ID de configuração, 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 o tamanho médio de objetos dividindo `StorageBytes` por `ObjectCount`, ou o número de bytes 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 gráficos

Com as métricas da Lente de Armazenamento do S3, é possível criar uma expressão de pesquisa. Por exemplo, é possível criar uma expressão de pesquisa para todas as métricas chamadas `IncompleteMultipartUploadStorageBytes` e adicionar `SUM` à expressão. Com essa expressão de pesquisa, é possível visualizar o total de bytes de multipart uploads 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 chamadas `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.

Casos de uso das métricas da Lente de Armazenamento do Amazon S3

Você pode usar o painel da Lente de Armazenamento do Amazon S3 para visualizar insights e tendências, sinalizar discrepâncias e receber recomendações. As métricas da Lente de Armazenamento do S3 são organizadas em categorias que estão alinhadas aos principais casos de uso. Você pode usar essas métricas para fazer o seguinte:

- Identificar oportunidades para otimização de custos
- Aplicar práticas recomendadas para proteção de dados
- Aplicar práticas recomendadas para gerenciamento de acesso
- Melhorar a performance das workloads de aplicações

Por exemplo, com as métricas de otimização de custos, você pode identificar oportunidades para reduzir os custos de armazenamento do Amazon S3. Você pode identificar buckets com multipart uploads existentes há mais de 7 dias ou buckets que estão acumulando versões desatualizadas.

Da mesma forma, você pode usar métricas de proteção de dados para identificar buckets que não estão seguindo as práticas recomendadas para proteção de dados em sua organização. Por

exemplo, você pode identificar buckets que não usam chaves do AWS Key Management Service (SSE-KMS) para criptografia padrão ou que não têm o Versionamento do S3 ativado.

Com as métricas de gerenciamento de acesso da Lente de Armazenamento do S3, você pode identificar as configurações de bucket para a Propriedade de Objeto do S3 a fim de migrar as permissões da lista de controle de acesso (ACL) para políticas de bucket e desativar as ACLs.

Se a opção [S3 Storage Lens advanced metrics](#) (Métricas avançadas da Lente de Armazenamento do S3) estiver ativada, você poderá usar as métricas detalhadas de código de status para obter contagens de solicitações bem-sucedidas ou malsucedidas, que podem ser usadas para solucionar problemas de acesso ou performance.

Com métricas avançadas, você também pode acessar métricas adicionais de otimização de custos e proteção de dados que podem ser usadas com o intuito de identificar oportunidades para reduzir ainda mais os custos gerais de armazenamento do S3 e se alinhar melhor às práticas recomendadas para proteção de dados. Por exemplo, as métricas avançadas de otimização de custos incluem contagens de regras de ciclo de vida que podem ser usadas para identificar buckets que não têm regras de ciclo de vida para expirar multipart uploads incompletos com mais de 7 dias. As métricas avançadas de proteção de dados incluem contagens de regras de replicação.

Para obter mais informações sobre categorias de métricas, consulte [Categorias de métricas](#). Para obter uma lista completa de métricas da Lente de Armazenamento do S3, consulte [Glossário de métricas de lente de armazenamento do Amazon S3](#).

Tópicos

- [Como usar o Amazon S3 Storage Lens para otimizar seus custos de armazenamento](#)
- [Usar a Lente de Armazenamento do S3 para proteger seus dados](#)
- [Como usar a Lente de Armazenamento do S3 para auditar configurações da Propriedade de Objeto](#)
- [Usar a Lente de Armazenamento do S3 para melhorar a performance](#)

Como usar o Amazon S3 Storage Lens para otimizar seus custos de armazenamento

Você pode usar as métricas de otimização de custos da Lente de Armazenamento do S3 para reduzir seu custo geral de armazenamento do S3. As métricas de otimização de custos podem ajudar você a confirmar que configurou o Amazon S3 de forma econômica e de acordo com as práticas recomendadas. Por exemplo, você pode identificar as seguintes oportunidades de otimização de custos:

- Buckets com multipart uploads incompletos com mais de 7 dias
- Buckets que estão acumulando várias versões desatualizadas
- Buckets que não têm regras de ciclo de vida para anular multipart uploads incompletos
- Buckets que não têm regras de ciclo de vida para expirar objetos de versões desatualizadas
- Buckets que não têm regras de ciclo de vida para migrar objetos para uma classe de armazenamento diferente

Depois, você pode usar esses dados para adicionar outras regras de ciclo de vida aos seus buckets.

Os exemplos a seguir mostram como você pode usar as métricas de otimização de custos no painel da Lente de Armazenamento do S3 para otimizar seus custos de armazenamento.

Tópicos

- [Identifique os seus maiores buckets do S3](#)
- [Descubra buckets inativos do Amazon S3](#)
- [Localize uploads fracionados incompletos](#)
- [Reduza o número de versões não atuais retidas](#)
- [Identificar buckets que não têm regras de ciclo de vida e revisar as contagens de regras de ciclo de vida](#)

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 classes de armazenamento. Com a Lente de Armazenamento do S3, você tem 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.

Etapa 1: 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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).

3. Na lista Dashboards (Painéis), escolha o painel que deseja visualizar.

Quando o painel é aberto, você pode ver a data mais recente em que a Lente de Armazenamento do S3 coletou métricas. Seu painel sempre carrega a data mais recente para a qual há métricas disponíveis.

4. Para ver a classificação dos seus maiores buckets de acordo com a métrica Total storage (Armazenamento total) para um intervalo de datas selecionado, role para baixo até a seção Top N overview for date (Visão geral dos N principais para data).

Você pode alternar a ordem de classificação para mostrar os buckets menores. Você também pode ajustar a seleção de Metric (Métrica) para classificar seus buckets de acordo com qualquer uma das métricas disponíveis. A seção Top N overview for date (Visão geral dos N principais para data) também mostra a variação percentual em relação ao dia ou semana anterior e uma linha em spark para visualizar a tendência. Essa tendência é de 14 dias para métricas gratuitas e 30 dias para métricas e recomendações avançadas.

 Note

Com métricas e recomendações avançadas da Lente de Armazenamento do S3, as métricas ficam disponíveis para consultas por 15 meses. Para obter mais informações, consulte [Seleção de métricas](#).

5. Para obter informações mais detalhadas sobre seus buckets, role até o topo da página e escolha a guia Bucket.

Na guia Bucket, você pode ver detalhes como a taxa de crescimento recente, o tamanho médio do objeto, os maiores prefixos e o número de objetos.

Etapa 2: Navegar até os buckets e investigar

Depois de identificar seus maiores buckets do S3, você pode navegar até cada bucket no console do S3 para visualizar os objetos no bucket, entender a workload associada ao bucket e identificar os proprietários internos. Você pode entrar em contato com os proprietários do bucket para descobrir se o crescimento é esperado ou se precisa de monitoramento e controle adicionais.

Descubra buckets inativos do Amazon S3

Se você tiver [métricas avançadas do S3 Storage Lens](#) habilitadas, poderá usar [métricas de atividade](#) para entender o quão 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.

Pré-requisito

Para ver as métricas de atividade no painel da Lente de Armazenamento do S3, você deve ativar a opção **Advanced metrics and recommendations** (Métricas e recomendações avançadas) da Lente de Armazenamento do S3 e selecionar **Activity metrics** (Métricas de atividade). Para obter mais informações, consulte [Criação e atualização dos painéis do Amazon S3 Storage Lens](#).

Etapa 1: Identificar os buckets ativos

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 **Storage Lens** (Lente de Armazenamento) e **Dashboards** (Painéis).
3. Na lista **Dashboards** (Painéis), escolha o painel que deseja visualizar.
4. Escolha a guia **Bucket**, depois role para baixo até a seção **Bubble analysis by buckets for date** (Análise de bolhas por buckets para data).

Na seção **Bubble analysis by buckets for date** (Análise de bolhas por buckets para data), você pode plotar seus buckets em várias dimensões usando três métricas quaisquer para representar o X-axis (Eixo X), Y-axis (Eixo Y) e Size (Tamanho) da bolha.

5. Para encontrar buckets que ficaram inativos, para X-axis (Eixo X), Y-axis (Eixo Y) e Size (Tamanho), escolha as métricas **Total storage** (Armazenamento total), **% retrieval rate** (% da taxa de recuperação) e **Average object size** (Tamanho médio dos objetos).

6. Na seção **Bubble analysis by buckets for date** (Análise de bolhas por buckets para data), encontre os buckets com taxas de recuperação iguais a zero (ou próximas de zero) e um tamanho de armazenamento relativo maior, depois escolha a bolha que representa o bucket.

Uma caixa aparecerá com opções para obter insights mais granulares. Faça um dos seguintes procedimentos:

- a. Para atualizar a guia **Bucket** para exibir métricas somente para o bucket selecionado, escolha **Drill Down** (Detalhar) e **Apply** (Aplicar).
- b. Para agregar os dados de bucket por conta, Região da AWS, classe de armazenamento ou bucket, escolha **Analyze by** (Analisar por), depois selecione uma opção em **Dimension** (Dimensão). Por exemplo, para agregar por classe de armazenamento, escolha **Storage class** (Classe de armazenamento) em **Dimension** (Dimensão).

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.

A guia **Bucket** do seu painel é atualizada para exibir os dados da agregação ou filtro selecionado. Se você agregou por classe de armazenamento ou outra dimensão, essa nova guia é aberta em seu painel (por exemplo, a guia **Storage class** (Classe de armazenamento)).

Etapa 2: Investigar os buckets inativos

Aqui, você pode identificar os proprietários de buckets inativos em sua conta ou organização e descobrir se esse armazenamento ainda é necessário. Depois, é possível otimizar os custos definindo as [configurações de validade de ciclo de vida](#) para esses buckets ou arquivando os dados em uma das [classes de armazenamento do Amazon S3 Glacier](#).

Para impedir que o problema de buckets inativos prossiga, você pode [fazer a transição dos seus dados de forma automática usando configurações do ciclo de vida do S3](#) em seus buckets, ou habilitar o [arquivamento automático com o S3 Intelligent-Tiering](#).

Você também pode usar a etapa 1 para identificar os buckets ativos. Depois, você pode garantir que esses buckets usem a [classe de armazenamento do S3](#) correta para garantir que atendam às solicitações com mais eficiência em termos de performance e custo.

Localize uploads fracionados incompletos

Você pode usar multipart uploads para fazer upload de objetos muito grandes (até 5 TB) como um conjunto de partes para obter uma throughput aprimorada e uma recuperação mais rápida diante de problemas de rede. Quando o processo de multipart upload não termina, as partes incompletas permanecem no bucket (em um estado inutilizável). Essas partes incompletas incorrem em custos de armazenamento até que o processo de upload seja concluído, ou até que as partes incompletas sejam removidas. Para obter mais informações, consulte [Carregar e copiar objetos usando multipart upload](#).

Com a Lente de Armazenamento do S3, você pode identificar o número de bytes de multipart uploads incompletos em sua conta ou em toda a sua organização, incluindo multipart uploads incompletos há mais de 7 dias. Para obter uma lista completa de métricas de multipart uploads incompletos, consulte [Glossário de métricas de lente de armazenamento do Amazon S3](#).

Uma prática recomendada é configurar regras de ciclo de vida para expirar multipart uploads incompletos há mais de determinada quantidade de dias. Ao criar uma regra de ciclo de vida para expirar multipart uploads incompletos, recomendamos 7 dias como um bom ponto de partida.

Etapa 1: Analise as tendências gerais de multipart uploads 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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o painel que deseja visualizar.
4. Na seção Snapshot for date (Snapshot para data), em Metrics categories (Categorias de métricas), escolha Cost optimization (Otimização de custos).

A seção Snapshot for date (Snapshot para data) é atualizada para exibir métricas de Cost optimization (Otimização de custos), que incluem Incomplete multipart upload bytes greater than 7 days old (Bytes de multipart uploads incompletos há mais de 7 dias).

Em qualquer gráfico no painel da Lente de Armazenamento do S3, você pode ver métricas para multipart uploads incompletos. Você pode usar essas métricas para avaliar ainda mais o impacto dos bytes de multipart uploads incompletos no seu armazenamento, incluindo a contribuição deles para as tendências gerais de crescimento. Você também pode detalhar níveis mais profundos de agregação usando as guias Account (Conta), Região da AWS, Bucket ou

Storage class (Classe de armazenamento) para uma análise mais profunda de seus dados. Para ver um exemplo, consulte [Descubra buckets inativos do Amazon S3](#).

Etapa 2: Identificar os buckets que têm o maior número de bytes de multipart uploads incompletos, mas não têm regras de ciclo de vida para anular multipart uploads incompletos

Pré-requisito

Para ver a métrica Abort incomplete multipart upload lifecycle rule count (Contagem de regras de ciclo de vida para anulação de multipart uploads incompletos) no painel da Lente de Armazenamento do S3, ative Advanced metrics and recommendations (Métricas e recomendações avançadas) da Lente de Armazenamento do S3, depois selecione Advanced cost optimization metrics (Métricas avançadas de otimização de custos). Para obter mais informações, consulte [Criação e atualização dos painéis do Amazon 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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o painel que deseja visualizar.
4. Para identificar buckets específicos com multipart uploads incompletos acumulados há mais de 7 dias, acesse a seção Top N overview for date (Visão geral dos N principais para data).

Por padrão, a seção Top N overview for date (Visão geral dos N principais para data) exibe métricas para os 3 principais buckets. Você pode aumentar ou diminuir o número de buckets no campo Top N (N principais). A seção Top N overview for date (Visão geral dos N principais para data) também mostra a variação percentual em relação ao dia ou semana anterior e uma linha em spark para visualizar a tendência. (Essa tendência é de 14 dias para métricas gratuitas e 30 dias para métricas e recomendações avançadas.)

Note

Com métricas e recomendações avançadas da Lente de Armazenamento do S3, as métricas ficam disponíveis para consultas por 15 meses. Para obter mais informações, consulte [Seleção de métricas](#).

5. Em Metric (Métrica), escolha Incomplete multipart upload bytes greater than 7 days old (Bytes de multipart uploads incompletos há mais de 7 dias) na categoria Cost optimization (Otimização de custos).

Em Top number buckets (Número buckets principais), você pode ver os buckets com a maior quantidade de bytes armazenados de multipart uploads incompletos há mais de 7 dias.

6. Para ver métricas mais detalhadas por bucket para multipart uploads incompletos, vá até o topo da página e escolha a guia Bucket.
7. Role para baixo até a seção Buckets. Em Metrics categories (Categorias de métricas), selecione Cost optimization (Otimização de custos). Depois, desmarque Summary (Resumo).

A lista Buckets é atualizada para exibir todas as métricas de Cost optimization (Otimização de custos) disponíveis para os buckets mostrados.

8. Para filtrar a lista Buckets a fim de exibir somente métricas específicas de otimização de custos, escolha o ícone de preferências



).

9. Desmarque todas as métricas de otimização de custos exceto Incomplete multipart upload bytes greater than 7 days old (Bytes de multipart uploads incompletos há mais de 7 dias) e Abort incomplete multipart upload lifecycle rule count (Contagem de regras de ciclo de vida para anulação de multipart uploads incompletos).
10. (Opcional) Em Page size (Tamanho da página), escolha o número de buckets a serem exibidos na lista.
11. Selecione a opção Confirm (Confirmar).

A lista Buckets é atualizada para exibir métricas por bucket para multipart uploads incompletos e contagens de regras de ciclo de vida. Você pode usar esses dados para identificar os buckets com mais bytes de multipart uploads incompletos há mais de 7 dias e que não têm regras de ciclo de vida para anular multipart uploads incompletos. Depois, você pode navegar até esses buckets no console do S3 e adicionar regras de ciclo de vida para excluir os multipart uploads incompletos que ficaram esquecidos.

Etapa 3: Adicionar uma regra de ciclo de vida para excluir multipart uploads incompletos depois de 7 dias

Para gerenciar uploads incompletos de várias partes automaticamente, use o console do S3 para criar uma configuração de ciclo de vida para expirar bytes de uploads incompletos de várias partes

em um bucket após determinado número de dias. Para obter mais informações, consulte [Configurar uma política de ciclo de vida de bucket para anular uploads incompletos de várias partes](#).


Reduza o número de versões não atuais retidas

Quando habilitado, o Versionamento do S3 retém várias cópias distintas do mesmo objeto que podem ser usadas para recuperar dados rapidamente se um objeto for acidentalmente excluído ou substituído. Se você habilitou o Versionamento do S3 sem configurar regras de ciclo de vida para fazer a transição ou expirar versões desatualizadas, um grande número de versões anteriores desatualizadas pode se acumular, o que pode ter implicações no custo do armazenamento. Para obter mais informações, consulte [Usando o versionamento em buckets do S3](#).

Etapa 1: Identificar os buckets com mais versões de objetos desatualizadas

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 Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o painel que deseja visualizar.
4. Na seção Snapshot for date (Snapshot para data), em Metrics categories (Categorias de métricas), escolha Cost optimization (Otimização de custos).

A seção Snapshot for date (Snapshot para data) é atualizada para exibir as métricas de Cost optimization (Otimização de custos), que incluem a métrica % noncurrent version bytes (% de bytes de versões desatualizadas). A métrica % noncurrent version bytes (% de bytes de versões desatualizadas) representa a proporção do total de bytes de armazenamento que é atribuída a versões desatualizadas, dentro do escopo do painel e para a data selecionada.


 Note

Se a sua % noncurrent version bytes (% de bytes de versões desatualizadas) for superior a 10% do seu armazenamento por conta, talvez você esteja armazenando muitas versões de objetos.

5. Para identificar buckets específicos que estão acumulando um grande número de versões desatualizadas:
 - a. Role para baixo até a seção Top N overview for date (Visão geral dos N principais para data). Em Top N (N principais), insira o número de buckets para os quais deseja ver dados.

- b. Em Metric (Métrica), escolha % noncurrent version bytes (% de bytes de versões desatualizadas).

Em Top number buckets (Número buckets principais), você pode ver os buckets (para o número especificado) com a maior % noncurrent version bytes (% de bytes de versões desatualizadas). A seção Top N overview for date (Visão geral dos N principais para data) também mostra a variação percentual em relação ao dia ou semana anterior e uma linha em spark para visualizar a tendência. Essa tendência é de 14 dias para métricas gratuitas e 30 dias para métricas e recomendações avançadas.

 Note

Com métricas e recomendações avançadas da Lente de Armazenamento do S3, as métricas ficam disponíveis para consultas por 15 meses. Para obter mais informações, consulte [Seleção de métricas](#).

- c. Para ver métricas mais detalhadas por bucket para versões de objetos desatualizadas, vá até o topo da página e escolha a guia Bucket.

Em qualquer gráfico ou visualização no painel da Lente de Armazenamento do S3, você pode detalhar níveis mais profundos de agregação usando as guias Account (Conta), Região da AWS, Storage class (Classe de armazenamento) ou Bucket. Para ver um exemplo, consulte [Descubra buckets inativos do Amazon S3](#).

- d. Na seção Buckets, para Metric categories (Categorias de métricas), selecione Cost optimization (Otimização de custos). Depois, desmarque Summary (Resumo).

Agora é possível ver a métrica % noncurrent version bytes (% de bytes de versões desatualizadas) ao lado de outras métricas relacionadas às versões desatualizadas.

Etapa 2: Identificar buckets que não têm regras de ciclo de vida de transição e expiração para gerenciar versões desatualizadas

Pré-requisito

Para ver as métricas Noncurrent version transition lifecycle rule count (Contagem de regras de ciclo de vida para transição de versões desatualizadas) e Noncurrent version expiration lifecycle rule count (Contagem de regras de ciclo de vida para validade de versões desatualizadas) no painel da Lente de Armazenamento do S3, ative a opção Advanced metrics and recommendations (Métricas

e recomendações avançadas) da Lente de Armazenamento do S3 e selecione Advanced cost optimization metrics (Métricas avançadas de otimização de custos). Para obter mais informações, consulte [Criação e atualização dos painéis do Amazon 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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o painel que deseja visualizar.
4. No painel da Lente de Armazenamento, escolha a guia Bucket.
5. Role para baixo até a seção Buckets. Em Metrics categories (Categorias de métricas), selecione Cost optimization (Otimização de custos). Depois, desmarque Summary (Resumo).

A lista Buckets é atualizada para exibir todas as métricas de Cost optimization (Otimização de custos) disponíveis para os buckets mostrados.

6. Para filtrar a lista Buckets a fim de exibir somente métricas específicas de otimização de custos, escolha o ícone de preferências



7. Desmarque todas as métricas de otimização de custos, exceto:
 - % noncurrent version bytes (% de bytes de versões desatualizadas)
 - Noncurrent version transition lifecycle rule count (Contagem de regras de ciclo de vida para transição de versões desatualizadas)
 - Noncurrent version expiration lifecycle rule count (Contagem de regras de ciclo de vida para validade de versões desatualizadas)
8. (Opcional) Em Page size (Tamanho da página), escolha o número de buckets a serem exibidos na lista.
9. Selecione a opção Confirm (Confirmar).

A lista Buckets é atualizada para exibir as métricas de bytes de versões desatualizadas e contagens de regras de ciclo de vida de versões desatualizadas. Você pode usar esses dados para identificar buckets que têm uma alta porcentagem de bytes de versões desatualizadas, mas que não têm regras de ciclo de vida de transição e validade. Depois, você pode navegar até esses buckets no console do S3 e adicionar regras de ciclo de vida a esses buckets.

Etapa 3: Adicionar regras de ciclo de vida para fazer a transição ou expirar versões de objetos desatualizadas

Depois de determinar quais buckets exigem investigação adicional, você pode navegar até os buckets no console do S3 e adicionar uma regra de ciclo de vida para expirar versões desatualizadas depois de um número especificado de dias. Alternativamente, para reduzir custos mantendo as versões desatualizadas, você pode configurar uma regra de ciclo de vida para fazer a transição das versões desatualizadas para uma das classes de armazenamento do Amazon S3 Glacier. Para obter mais informações, consulte [Exemplo 6: Especificar uma regra de ciclo de vida para um bucket com versionamento habilitado](#).

Identificar buckets que não têm regras de ciclo de vida e revisar as contagens de regras de ciclo de vida

A Lente de Armazenamento do S3 fornece métricas de contagem de regras do Ciclo de Vida do S3 que você pode usar para identificar buckets que não têm regras de ciclo de vida. Para encontrar buckets que não têm regras de ciclo de vida, você pode usar a métrica Total buckets without lifecycle rules (Total de buckets sem regras de ciclo de vida). Um bucket sem o Ciclo de Vida do S3 configurado pode ter armazenamento que não é mais necessário ou pode ser migrado para uma classe de armazenamento de menor custo. Você também pode usar métricas de contagem de regras de ciclo de vida para identificar buckets que não têm tipos específicos de regras de ciclo de vida, como regras de validade ou transição.

Pré-requisito


Para ver métricas de contagem de regras de ciclo de vida e a métrica Total buckets without lifecycle rules (Total de buckets sem regras de ciclo de vida) no painel da Lente de Armazenamento do S3, ative Advanced metrics and recommendations (Métricas e recomendações avançadas) da Lente de Armazenamento do S3, depois selecione Advanced cost optimization metrics (Métricas avançadas de otimização de custos). Para obter mais informações, consulte [Criação e atualização dos painéis do Amazon S3 Storage Lens](#).

Etapa 1: Identificar buckets sem regras 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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o painel que deseja visualizar.

4. Para identificar buckets específicos sem regras de ciclo de vida, role para baixo até a seção Top N overview for dates (Visão geral dos N principais para data).

Por padrão, a seção Top N overview for date (Visão geral dos N principais para data) exibe métricas para os 3 principais buckets. No campo Top N (N principais), você pode aumentar o número de buckets. A seção Top N overview for date (Visão geral dos N principais para data) também mostra a variação percentual em relação ao dia ou semana anterior e uma linha em spark para visualizar a tendência. Essa tendência é de 14 dias para métricas gratuitas e 30 dias para métricas e recomendações avançadas.

 Note

Com métricas e recomendações avançadas da Lente de Armazenamento do S3, as métricas ficam disponíveis para consultas por 15 meses. Para obter mais informações, consulte [Seleção de métricas](#).

5. Em Metric (Métrica), escolha Total buckets without lifecycle rules (Total de buckets sem regras de ciclo de vida) na categoria Cost optimization (Otimização de custos).
6. Analise os dados a seguir para Total buckets without lifecycle rules (Total de buckets sem regras de ciclo de vida):
 - Top number accounts (Número contas principais): veja as contas que têm mais buckets sem regras de ciclo de vida.
 - Top number Regions (Número regiões principais): veja um detalhamento dos buckets sem regras de ciclo de vida por região.
 - Top number buckets (Número buckets principais): veja os buckets que não têm regras de ciclo de vida.

Em qualquer gráfico ou visualização no painel da Lente de Armazenamento do S3, você pode detalhar níveis mais profundos de agregação usando as guias Account (Conta), Região da AWS, Storage class (Classe de armazenamento) ou Bucket. Para ver um exemplo, consulte [Descubra buckets inativos do Amazon S3](#).

Depois de identificar quais buckets não têm regras de ciclo de vida, você também pode revisar contagens específicas de regras de ciclo de vida para os buckets.

Etapa 2: Revisar a contagem de regras do ciclo de vida dos 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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o painel que deseja visualizar.
4. No painel da Lente de Armazenamento do S3, escolha a guia Bucket.
5. Role para baixo até a seção Buckets. Em Metrics categories (Categorias de métricas), selecione Cost optimization (Otimização de custos). Depois, desmarque Summary (Resumo).

A lista Buckets é atualizada para exibir todas as métricas de Cost optimization (Otimização de custos) disponíveis para os buckets mostrados.

6. Para filtrar a lista Buckets a fim de exibir somente métricas específicas de otimização de custos, escolha o ícone de preferências



7. Desmarque todas as métricas de otimização de custos, exceto:
 - Transition lifecycle rule count (Contagem de regras de ciclo de vida para transição)
 - Expiration lifecycle rule count (Contagem de regras de ciclo de vida para validade)
 - Noncurrent version transition lifecycle rule count (Contagem de regras de ciclo de vida para transição de versões desatualizadas)
 - Noncurrent version expiration lifecycle rule count (Contagem de regras de ciclo de vida para validade de versões desatualizadas)
 - Abort incomplete multipart upload lifecycle rule count (Contagem de regras de ciclo de vida para anulação de multipart uploads incompletos)
 - Total lifecycle rule count (Contagem total de regras de ciclo de vida)
8. (Opcional) Em Page size (Tamanho da página), escolha o número de buckets a serem exibidos na lista.
9. Selecione a opção Confirm (Confirmar).

A lista Buckets é atualizada para exibir as métricas de contagem de regras de ciclo de vida para os buckets. Você pode usar esses dados para identificar buckets sem regras de ciclo de vida ou que não tenham tipos específicos de regras de ciclo de vida; por exemplo, regras de validade ou

transição. Depois, você pode navegar até esses buckets no console do S3 e adicionar regras de ciclo de vida a esses buckets.

Etapa 3: Adicionar regras de ciclo de vida

Depois de identificar buckets sem regras de ciclo de vida, você pode adicionar regras de ciclo de vida. Para obter mais informações, consulte [Definir uma configuração de ciclo de vida em um bucket](#) e [Exemplos de configuração do S3 Lifecycle](#).

Usar a Lente de Armazenamento do S3 para proteger seus dados

Você pode usar as métricas de proteção de dados da Lente de Armazenamento do Amazon S3 para identificar buckets nos quais as práticas recomendadas de proteção de dados não foram aplicadas. Você pode usar essas métricas para agir e aplicar configurações padrão que se alinham às práticas recomendadas para proteger seus dados em todos os buckets de sua conta ou organização. Por exemplo, você pode usar métricas de proteção de dados para identificar buckets que não usam chaves do AWS Key Management Service (AWS KMS) para criptografia padrão ou solicitações que usam o AWS Signature Version 2 (SigV2).

Os casos de uso a seguir fornecem estratégias para usar o painel da Lente de Armazenamento do S3 para identificar valores discrepantes e aplicar as práticas recomendadas de proteção de dados em seus bucket da Lente de Armazenamento do S3.

Tópicos

- [Identificar buckets que não usam a criptografia do lado do servidor com AWS KMS como criptografia padrão \(SSE-KMS\)](#)
- [Identificar buckets com o Versionamento do S3 ativado](#)
- [Identificar solicitações que usam o AWS Signature Version 2 \(SigV2\)](#)
- [Contar o número total de regras de replicação para cada bucket](#)
- [Identificar a porcentagem de bytes do Bloqueio de Objeto](#)

Identificar buckets que não usam a criptografia do lado do servidor com AWS KMS como criptografia padrão (SSE-KMS)

Com a criptografia padrão do Amazon S3, você pode definir o comportamento de criptografia padrão para um bucket do S3. Para obter mais informações, consulte [the section called “Definir criptografia de bucket padrão”](#).

Você pode usar as métricas SSE-KMS enabled bucket count (Contagem de buckets habilitados para SSE-KMS) e % SSE-KMS enabled buckets (% de buckets habilitados para SSE-KMS) para identificar buckets que usam criptografia do lado do servidor com chaves do AWS KMS (SSE-KMS) como criptografia padrão. A Lente de Armazenamento do S3 também fornece métricas para bytes não criptografados, objetos não criptografados, bytes criptografados e objetos criptografados. Para obter uma lista completa de métricas, consulte [Glossário de métricas de lente de armazenamento do Amazon S3](#).

Você pode analisar as métricas de criptografia do SSE-KMS no contexto das métricas gerais de criptografia para identificar buckets que não usam SSE-KMS. Se você quiser usar SSE-KMS para todos os buckets em sua conta ou organização, poderá atualizar as configurações de criptografia padrão desses buckets para usar SSE-KMS. Além de SSE-KMS, você pode usar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) ou chaves fornecidas pelo cliente (SSE-C). Para obter mais informações, consulte [Proteger dados com criptografia](#).

Etapa 1: Identificar quais buckets estão usando SSE-KMS como criptografia 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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o nome do painel que deseja visualizar.
4. Na seção Trends and distributions (Tendências e distribuições), escolha % SSE-KMS enabled bucket count (% de contagem de buckets habilitados para SSE-KMS) como métrica primária e % encrypted bytes (% de bytes criptografados) como métrica secundária.

O gráfico Trend for date (Tendências para data) é atualizado para exibir tendências de SSE-KMS e bytes criptografados.

5. Para ver insights mais granulares por bucket para SSE-KMS:
 - a. Escolha um ponto no gráfico. Uma caixa aparecerá com opções para obter insights mais granulares.
 - b. Escolha a dimensão Buckets. Em seguida, escolha Aplicar.
6. No gráfico Distribution by buckets for date (Distribuição por buckets para data), escolha a métrica SSE-KMS enabled bucket count (Contagem de buckets habilitados para SSE-KMS).
7. Agora você pode ver quais buckets têm SSE-KMS habilitada e quais não.

Etapa 2: Atualizar as configurações de criptografia padrão do bucket

Agora que você determinou quais buckets usam SSE-KMS no contexto de % encrypted bytes (% de bytes criptografados), você pode identificar buckets que não usam SSE-KMS. Depois, opcionalmente, você pode navegar até esses buckets no console do S3 e atualizar suas configurações de criptografia padrão para usar SSE-KMS ou SSE-S3. Para obter mais informações, consulte [Configurar a criptografia padrão](#).

Identificar buckets com o Versionamento do S3 ativado

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. Você pode usar a métrica Versioning-enabled bucket count (Contagem de buckets habilitados para Versionamento) para ver quais buckets usam o Versionamento do S3. Depois, você pode usar o console do S3 para ativar o Versionamento do S3 para outros buckets.

Etapa 1: Identificar buckets com o Versionamento do S3 ativado

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 (Lente de Armazenamento) e Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o nome do painel que deseja visualizar.
4. Na seção Trends and distributions (Tendências e distribuições), escolha Versioning-enabled bucket count (Contagem de buckets habilitados para Versionamento) como métrica primária e Buckets como métrica secundária.

O gráfico Trend for date (Tendência para data) é atualizado para exibir tendências dos buckets com Versionamento do S3 ativado. Logo abaixo da linha de tendências, você pode ver as subseções Storage class distribution (Distribuição por classe de armazenamento) e Region distribution (Distribuição por região).

5. Para ver informações mais granulares de qualquer um dos buckets exibidos no gráfico Trend for date (Tendência para data) com o intuito de realizar uma análise mais profunda, faça o seguinte:
 - a. Escolha um ponto no gráfico. Uma caixa aparecerá com opções para obter insights mais granulares.

- b. Escolha uma dimensão para aplicar aos dados para uma análise mais profunda: Account (Conta), Região da AWS, Storage class (Classe de armazenamento) ou Bucket. Em seguida, escolha Aplicar.
6. Na seção Bubble analysis by buckets for date (Análise de bolhas por buckets para data), escolha as métricas Versioning-enabled bucket count (Contagem de buckets habilitados para Versionamento), Buckets e Active buckets (Buckets ativos).

A seção Bubble analysis by buckets for date (Análise de bolhas por buckets para data) é atualizada para exibir dados das métricas que você selecionou. Você pode usar esses dados para ver quais buckets estão com o Versionamento do S3 ativado no contexto da sua contagem total de buckets. Na seção Bubble analysis by buckets for date (Análise de bolhas por buckets para data), você pode plotar seus buckets em várias dimensões usando três métricas quaisquer para representar o X-axis (Eixo X), Y-axis (Eixo Y) e Size (Tamanho) da bolha.

Etapa 2: Habilitar o Versionamento do S3

Depois de identificar os buckets com Versionamento do S3 ativado, você pode identificar os buckets que nunca tiveram o Versionamento do S3 ativado ou que estão com o versionamento suspenso. Depois, opcionalmente, você pode ativar o versionamento para esses buckets no console S3. Para obter mais informações, consulte [Habilitar o versionamento em buckets](#).

Identificar solicitações que usam o AWS Signature Version 2 (SigV2)

Você pode usar a métrica All unsupported signature requests (Todas as solicitações de assinatura incompatíveis) para identificar solicitações que usam o AWS Signature Version 2 (SigV2). Esses dados podem ajudar você a identificar aplicações específicas que usam o SigV2. Depois, você pode migrar essas aplicações para o AWS Signature Version 4 (SigV4).

O SigV4 é o método de assinatura recomendado para todas as novas aplicações do S3. O SigV4 oferece segurança aprimorada e é compatível com todas as Regiões da AWS. Para obter mais informações, consulte [Atualização do Amazon S3 – Período de defasagem do SigV2 estendido e modificado](#).

Pré-requisito


Para ver All unsupported signature requests (Todas as solicitações de assinatura incompatíveis) no painel da Lente de Armazenamento do S3, você deve ativar a opção Advanced metrics and recommendations (Métricas e recomendações avançadas) da Lente de Armazenamento do S3 e

selecionar Advanced data protection metrics (Métricas avançadas de proteção de dados). Para obter mais informações, consulte [Criação e atualização dos painéis do Amazon S3 Storage Lens](#).

Etapa 1: Examinar as tendências de assinatura do SigV2 por Conta da AWS, região e 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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o nome do painel que deseja visualizar.
4. Para identificar buckets, contas e regiões específicos com solicitações que usam SigV2:
 - a. Em Top N overview for date (Visão geral dos N principais para data), em Top N (N principais), insira o número de buckets para os quais deseja ver dados.
 - b. Em Metric (Métrica), escolha All unsupported signature requests (Todas as solicitações de assinatura incompatíveis) na categoria Data protection (Proteção de dados).

A seção Top N overview for date (Visão geral dos N principais para data) é atualizada para exibir dados de solicitações SigV2 por conta, Região da AWS e bucket. A seção Top N overview for date (Visão geral dos N principais para data) também mostra a variação percentual em relação ao dia ou semana anterior e uma linha em spark para visualizar a tendência. Essa tendência é de 14 dias para métricas gratuitas e 30 dias para métricas e recomendações avançadas.

 Note

Com métricas e recomendações avançadas da Lente de Armazenamento do S3, as métricas ficam disponíveis para consultas por 15 meses. Para obter mais informações, consulte [Seleção de métricas](#).

Etapa 2: Identificar buckets que são acessados por aplicações por meio de solicitações SigV2

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 Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).

3. Na lista Dashboards (Painéis), escolha o nome do painel que deseja visualizar.
4. No painel da Lente de Armazenamento, escolha a guia Bucket.
5. Role para baixo até a seção Buckets. Em Metrics categories (Categorias de métricas), escolha Data protection (Proteção de dados). Depois, desmarque Summary (Resumo).

A lista Buckets é atualizada para exibir todas as métricas de Data protection (Proteção de dados) disponíveis para os buckets mostrados.

6. Para filtrar a lista Buckets a fim de exibir somente métricas específicas de proteção de dados, escolha o ícone de preferências



7. Desmarque todas as métricas de proteção de dados, exceto:
 - All unsupported signature requests (Todas as solicitações de assinatura incompatíveis)
 - % all unsupported signature requests (% de todas as solicitações de assinatura incompatíveis)
8. (Opcional) Em Page size (Tamanho da página), escolha o número de buckets a serem exibidos na lista.
9. Selecione a opção Confirm (Confirmar).

A lista Buckets é atualizada para exibir métricas por bucket para solicitações SigV2. Você pode usar esses dados para identificar buckets específicos que têm solicitações SigV2. Depois, você pode usar essas informações para migrar aplicações para SigV4. Para obter mais informações, consulte [Autenticação de solicitações \(AWS Signature Version 4\)](#) na Referência de APIs do Amazon Simple Storage Service.

Contar o número total de regras de replicação para cada bucket


A Replicação do S3 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. Para obter mais informações, consulte [Replicação de objetos](#).

Você pode usar as métricas de contagem de regras de replicação da Lente de Armazenamento do S3 para obter informações detalhadas por bucket sobre os buckets que estão configurados para replicação. Essas informações incluem regras de replicação dentro e entre buckets e regiões.

Pré-requisito

Para ver as métricas de contagem de regras de replicação no painel da Lente de Armazenamento do S3, você deve ativar a opção *Advanced metrics and recommendations* (Métricas e recomendações avançadas) da Lente de Armazenamento do S3 e selecionar *Advanced data protection metrics* (Métricas avançadas de proteção de dados). Para obter mais informações, consulte [Criação e atualização dos painéis do Amazon S3 Storage Lens](#).

Etapa 1: Contar o número total de regras de replicação para cada 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 esquerdo, escolha *Storage Lens* (Lente de Armazenamento) e *Dashboards* (Painéis).
3. Na lista *Dashboards* (Painéis), escolha o nome do painel que deseja visualizar.
4. No painel da Lente de Armazenamento, escolha a guia *Bucket*.
5. Role para baixo até a seção *Buckets*. Em *Metrics categories* (Categorias de métricas), escolha *Data protection* (Proteção de dados). Depois, desmarque *Summary* (Resumo).
6. Para filtrar a lista *Buckets* para exibir somente métricas de contagem de regras de replicação, escolha o ícone de preferências ).
7. Desmarque todas as métricas de proteção de dados, exceto as métricas de contagem de regras de replicação:
 - *Same-Region Replication rule count* (Contagem de regras de replicação na mesma região)
 - *Cross-Region Replication rule count* (Contagem de regras de replicação entre regiões)
 - *Same-account replication rule count* (Contagem de regras de replicação na mesma conta)
 - *Cross-account replication rule count* (Contagem de regras de replicação entre contas)
 - *Total replication rule count* (Contagem total de regras de replicação)
8. (Opcional) Em *Page size* (Tamanho da página), escolha o número de buckets a serem exibidos na lista.
9. Selecione a opção *Confirm* (Confirmar).

Etapa 2: Adicionar regras de replicação

Depois de ter a contagem de regras de replicação por bucket, você pode, opcionalmente, criar regras de replicação adicionais. Para obter mais informações, consulte [Demonstrações: exemplos para configurar a replicação](#).

Identificar a porcentagem de bytes do Bloqueio de Objeto

Com o Bloqueio de Objeto do S3, é possível armazenar objetos usando um modelo gravar uma vez, ler muitas (WORM). Use o Bloqueio de Objeto para ajudar a evitar que um objeto seja excluído ou substituído por um período fixo ou indefinidamente. Só é possível ativar o Bloqueio de Objeto ao criar um bucket e ao ativar o Versionamento do S3. No entanto, você pode editar o período de retenção para versões individuais de objetos ou aplicar retenções legais para buckets com o Bloqueio de Objeto ativado. Para obter mais informações, consulte [Usar o bloqueio de objetos do S3](#).

Você pode usar as métricas do Bloqueio de Objeto na Lente de Armazenamento do S3 para ver a métrica % Object Lock bytes (% de bytes do Bloqueio de Objeto) para sua conta ou organização. Você pode usar essas informações para identificar buckets em sua conta ou organização que não estejam seguindo as práticas recomendadas de proteção de dados.

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 Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o nome do painel que deseja visualizar.
4. Na seção Snapshot, em Metrics categories (Categorias de métricas), escolha Data protection (Proteção de dados).


A seção Snapshot é atualizada para exibir as métricas de proteção de dados, incluindo a métrica % Object Lock bytes (% de bytes do Bloqueio de Objeto). Você pode ver a porcentagem geral de bytes do Bloqueio de Objeto para sua conta ou organização.

5. Para ver a métrica % Object Lock bytes (% de bytes do Bloqueio de Objeto) por bucket, role para baixo até a seção Top N overview (Visão geral dos N principais).

Para obter dados por objeto para o Bloqueio de Objeto, você também pode usar as métricas Object Lock object count (Contagem de objetos do Bloqueio de Objeto) e % Object Lock objects (% de objetos do Bloqueio de Objeto).

6. Em Metric (Métrica), escolha % Object Lock bytes (% de bytes do Bloqueio de Objeto) na categoria Data protection (Proteção de dados).

Por padrão, a seção Top N overview for date (Visão geral dos N principais para data) exibe métricas para os 3 principais buckets. No campo Top N (N principais), você pode aumentar o número de buckets. A seção Top N overview for date (Visão geral dos N principais para data) também mostra a variação percentual em relação ao dia ou semana anterior e uma linha em spark para visualizar a tendência. Essa tendência é de 14 dias para métricas gratuitas e 30 dias para métricas e recomendações avançadas.

 Note

Com métricas e recomendações avançadas da Lente de Armazenamento do S3, as métricas ficam disponíveis para consultas por 15 meses. Para obter mais informações, consulte [Seleção de métricas](#).

7. Revise os seguintes dados para % Object Lock bytes (% de bytes do Bloqueio de Objeto):
 - Top number accounts (Número contas principais): veja quais contas têm o maior e o menor valor para % Object Lock bytes (% de bytes do Bloqueio de Objeto).
 - Top number Regions (Número regiões principais): veja um detalhamento de % Object Lock bytes (% de bytes do Bloqueio de Objeto) por região.
 - Top number buckets (Número buckets principais): veja quais buckets têm o maior e o menor valor para % Object Lock bytes (% de bytes do Bloqueio de Objeto).

Como usar a Lente de Armazenamento do S3 para auditar configurações da Propriedade de Objeto

A Propriedade de Objeto do S3 é uma configuração por bucket do S3 que você pode usar para desabilitar as listas de controle de acesso (ACLs) e controlar a propriedade dos objetos em seu bucket. Se você definir a Propriedade de Objeto como Imposto pelo proprietário do bucket, poderá desabilitar as [listas de controle de acesso \(ACLs\)](#) e assumir a propriedade de todos os objetos de seu bucket. Essa abordagem simplifica o gerenciamento de acesso aos 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.

A maioria dos casos de uso modernos no Amazon S3 não exigem mais o uso de ACLs. Portanto, recomendamos desabilitar as ACLs, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Ao definir a Propriedade de Objeto como Imposto pelo proprietário do bucket, você pode desabilitar as ACLs e confiar em políticas para controle de acesso. Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

Com as métricas de gerenciamento de acesso da Lente de Armazenamento do S3, você pode identificar buckets que não estão com as ACLs desativadas. Depois de identificar esses buckets, você pode migrar as permissões de ACL para políticas e desativar as ACLs para esses buckets.

Tópicos

- [Etapa 1: Identificar tendências gerais para as configurações da Propriedade de Objeto](#)
- [Etapa 2: Identificar tendências por bucket para as configurações da Propriedade de Objeto](#)
- [Etapa 3: Atualizar sua configuração da Propriedade de Objeto para Imposto pelo proprietário do bucket a fim de desativar as ACLs](#)

Etapa 1: Identificar tendências gerais para as configurações da Propriedade de Objeto

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 (Lente de Armazenamento) e Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o nome do painel que deseja visualizar.
4. Na seção Snapshot for date (Snapshot para data), em Metrics categories (Categorias de métricas), escolha Access management (Gerenciamento de acesso).

A seção Snapshot for date (Snapshot para data) é atualizada para exibir a métrica % Object Ownership bucket owner enforced (% de Imposto pelo proprietário do bucket para Propriedade de Objeto). Você pode ver a porcentagem de buckets em sua conta ou organização que usam a configuração Imposto pelo proprietário do bucket na Propriedade de Objeto para desabilitar ACLs.

Etapa 2: Identificar tendências por bucket para as configurações da Propriedade de Objeto

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 Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o nome do painel que deseja visualizar.
4. Para ver métricas mais detalhadas por bucket, escolha a guia Bucket.
5. Na seção Distribution by buckets for date (Distribuição por buckets para data), escolha a métrica % Object Ownership bucket owner enforced (% de Imposto pelo proprietário do bucket para Propriedade de Objeto).

O gráfico é atualizado para mostrar uma divisão por bucket para % Object Ownership bucket owner enforced (% de Imposto pelo proprietário do bucket para Propriedade de Objeto).

Você pode ver quais buckets usam a configuração Imposto pelo proprietário do bucket para a Propriedade de Objeto a fim de desabilitar ACLs.

6. Para ver as configurações Imposto pelo proprietário do bucket no contexto, role para baixo até a seção Buckets. Em Metrics categories (Categorias de métricas), selecione Access management (Gerenciamento de acesso). Depois, desmarque Summary (Resumo).

A lista Buckets exibe dados para as três configurações da Propriedade de Objeto: Imposto pelo proprietário do bucket, Proprietário do bucket preferido e Autor do objeto.

7. Para filtrar a lista Buckets a fim de exibir métricas somente para uma configuração específica da Propriedade de Objeto, escolha o ícone de preferências



).

8. Desmarque as métricas que não deseja visualizar.
9. (Opcional) Em Page size (Tamanho da página), escolha o número de buckets a serem exibidos na lista.
10. Selecione a opção Confirm (Confirmar).

Etapa 3: Atualizar sua configuração da Propriedade de Objeto para Imposto pelo proprietário do bucket a fim de desativar as ACLs

Depois de identificar os buckets que usam as configurações Autor do objeto e Proprietário do bucket preferido para Propriedade de Objeto, você pode migrar suas permissões de ACL para

políticas de bucket. Quando terminar de migrar suas permissões de ACL, você poderá atualizar suas configurações da Propriedade de Objeto para Imposto pelo proprietário do bucket a fim de desativar as ACLs. Para obter mais informações, consulte [Pré-requisitos para desabilitar ACLs](#).

Usar a Lente de Armazenamento do S3 para melhorar a performance

Se a opção [S3 Storage Lens advanced metrics](#) (Métricas avançadas da Lente de Armazenamento do S3) estiver ativada, você poderá usar as métricas detalhadas de código de status para obter contagens de solicitações bem-sucedidas ou malsucedidas. Você pode usar essas informações para solucionar problemas de acesso ou performance. As métricas detalhadas de código de status mostram contagens para códigos de status HTTP, como 403 Proibido e 503 Serviço Indisponível. Você pode examinar as tendências gerais para obter métricas detalhadas de código de status em buckets do S3, contas e organizações. Depois, você pode detalhar as métricas por bucket para identificar workloads que estão acessando esses buckets e causando erros.

Por exemplo, você pode observar a métrica 403 Forbidden error count (Contagem de erros 403 Proibido) para identificar workloads que estão acessando buckets sem as permissões corretas em vigor. Depois de identificar essas workloads, você pode analisar em mais detalhes fora da Lente de Armazenamento do S3 para solucionar os erros 403 Proibido.

Este exemplo mostra como fazer uma análise de tendência para o erro 403 Proibido usando as métricas 403 Forbidden error count (Contagem de erros 403 Proibido) e % 403 Forbidden errors (% de erros 403 Proibido). Você pode usar essas métricas para identificar workloads que estão acessando buckets sem as permissões corretas em vigor. Você pode fazer uma análise de tendência semelhante para qualquer uma das outras Detailed status code metrics (Métricas de códigos de status detalhados). Para obter mais informações, consulte [Glossário de métricas de lente de armazenamento do Amazon S3](#).

Pré-requisito

Para ver Detailed status code metrics (Métricas de códigos de status detalhados) no painel da Lente de Armazenamento do S3, você deve ativar a opção Advanced metrics and recommendations (Métricas e recomendações avançadas) da Lente de Armazenamento do S3 e selecionar Detailed status code metrics (Métricas de códigos de status detalhados). Para obter mais informações, consulte [Criação e atualização dos painéis do Amazon S3 Storage Lens](#).

Tópicos

- [Etapa 1: Fazer uma análise de tendências para um código de status HTTP individual](#)
- [Etapa 2: Analisar a contagem de erros por bucket](#)

- [Etapa 3: Solucionar problemas](#)

Etapa 1: Fazer uma análise de tendências para um código de status HTTP individual

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 Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o nome do painel que deseja visualizar.
4. Na seção Trends and distributions (Tendências e distribuições), em Primary metric (Métrica primária), escolha 403 Forbidden error count (Contagem de erros 403 Proibido) na categoria Detailed status codes (Códigos de status detalhados). Em Secondary metric (Métrica secundária), escolha % 403 Forbidden errors (% de erros 403 Proibido).
5. Role para baixo até a seção Top N overview for date (Visão geral dos N principais para data). Em Metrics (Métrica), escolha 403 Forbidden error count (Contagem de erros 403 Proibido) ou % 403 Forbidden errors (% de erros 403 Proibido) na categoria Detailed status codes (Códigos de status detalhados).

A seção Top N overview for date (Visão geral dos N principais para data) para exibir as principais contagens de erros 403 Proibido por conta, Região da AWS e bucket.

Etapa 2: Analisar a contagem de erros por 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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e Dashboards (Painéis).
3. Na lista Dashboards (Painéis), escolha o nome do painel que deseja visualizar.
4. No painel da Lente de Armazenamento, escolha a guia Bucket.
5. Role para baixo até a seção Buckets. Em Metrics categories (Categorias de métricas), selecione as métricas Detailed status code (Códigos de status detalhados). Depois, desmarque Summary (Resumo).

A lista Buckets é atualizada para exibir todas as métricas disponíveis para códigos de status detalhados. Você pode usar essas informações para ver quais buckets têm uma grande

proporção de determinados códigos de status HTTP e quais códigos de status são comuns entre os buckets.

6. Para filtrar a lista Buckets a fim de exibir somente métricas específicas de códigos de status detalhados, escolha o ícone de preferências



7. Desmarque todas as métricas de códigos de status detalhados que não quiser ver na lista Buckets.
8. (Opcional) Em Page size (Tamanho da página), escolha o número de buckets a serem exibidos na lista.
9. Selecione a opção Confirm (Confirmar).

A lista Buckets exibe métricas de contagem de erros para o número de buckets especificado. Você pode usar essas informações para identificar buckets específicos que estão apresentando muitos erros e solucionar erros por bucket.

Etapa 3: Solucionar problemas

Depois de identificar os buckets que estão com alta proporção de códigos de status HTTP específicos, você pode solucionar esses erros. Para obter mais informações, consulte as informações a seguir.

- [Por que estou recebendo o erro 403 Proibido quando tento fazer upload de arquivos no Amazon S3?](#)
- [Por que estou recebendo o erro 403 Proibido quando tento modificar uma política de bucket no Amazon S3?](#)
- [Como soluciono os erros 403 Proibido do meu bucket do Amazon S3 onde todos os recursos são da mesma Conta da AWS?](#)
- [Como soluciono um erro HTTP 500 ou 503 do Amazon S3?](#)

Glossário de métricas de lente de armazenamento do Amazon S3

O glossário de métricas da Lente de Armazenamento do Amazon S3 oferece uma lista completa de métricas gratuitas e avançadas para a Lente de Armazenamento do S3.

A Lente de Armazenamento do S3 oferece métricas gratuitas para todos os painéis e configurações, com a opção de upgrade para métricas avançadas.

- Métricas gratuitas contêm métricas relevantes para o uso do armazenamento, como o número de buckets e objetos na sua conta. As métricas gratuitas também incluem métricas baseadas em casos de uso, como otimização de custos e métricas de proteção de dados. Todas as métricas gratuitas são coletadas diariamente, e os dados ficam disponíveis para consultas por até 14 dias.
- Métricas e recomendações avançadas incluem todas as métricas gratuitas e métricas adicionais, como métricas avançadas de proteção de dados e otimização de custos. As métricas avançadas também incluem categorias de métricas adicionais, como métricas de atividade e métricas detalhadas de código de status. Os dados de métricas avançadas 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](#). Para obter mais informações sobre os recursos de métricas e recomendações avançadas, consulte [Seleção de métricas](#).

Note

Para grupos de Lente de Armazenamento, apenas métricas de armazenamento de nível gratuito estão disponíveis. As métricas de nível avançado não estão disponíveis no nível do grupo de Lente de Armazenamento.

Nomes de métricas

A coluna Nome da métrica na tabela a seguir fornece o nome de cada métrica da Lente de Armazenamento do S3 no console S3. A coluna CloudWatch e exportação fornece o nome de cada métrica no Amazon CloudWatch e o arquivo de exportação de métricas que você pode configurar no painel da Lente de Armazenamento do S3.

Fórmulas de métricas derivadas

As métricas derivadas não estão disponíveis para a exportação de métricas e a opção de publicação do CloudWatch. No entanto, você pode usar as fórmulas de métricas mostradas na coluna Fórmula de métricas derivadas para computá-las.

Interpretar os símbolos de prefixo da Lente de Armazenamento do Amazon S3 para os múltiplos de unidades métricas (K, M, G etc.)

Os múltiplos de unidades de métricas da Lente de Armazenamento do S3 são escritos com símbolos de prefixo. Esses símbolos de prefixo correspondem aos 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 prefixos do SI](#).

Note

- A unidade de medida para bytes de armazenamento do S3 está em gigabytes binários (GB), em que 1 GB é 2^{30} bytes, 1 TB é 2^{40} bytes e 1 PB é 2^{50} bytes. Essa unidade de medida também é conhecida como gibibyte (GiB), conforme definido pela Comissão Eletrotécnica Internacional (IEC).
- Quando um objeto atinge o fim de seu ciclo de vida, com base em sua configuração de ciclo de vida, o Amazon S3 o coloca em uma fila para remoção e o remove assincronamente. Assim, pode haver um atraso entre a data de expiração e a data em que o Amazon S3 remove um objeto. A Lente de Armazenamento do S3 não inclui métricas para objetos que expiraram, mas não foram removidos. Para obter mais informações sobre as ações de expiração no ciclo de vida do S3, consulte [Expirando objetos](#).

Glossário de métricas da Lente de Armazenamento do S3

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Cate 2	De	Fórm da métr deriv
Armazenamento total	StorageBytes	O armazenamento total, incluindo uploads incompletos de várias partes, metadados de objetos e marcadores de exclusão	Grati	Resi	N:	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Cate 2	De	Fórm da métr deriv
Contagem de objetos	ObjectCount	A contagem total de objetos	Gratu	Resu	N	-
Tamanho médio do objeto	-	O tamanho médio do objeto	Gratu	Resu	Y	$\text{sum(StorageBytes)/sum(ObjectCount)}$
Buckets ativos	-	O número total de buckets em uso ativo com armazenamento > 0 bytes	Gratu	Resu	Y	-
Buckets	-	O número total de buckets	Gratu	Resu	Y	-
Contas	-	O número de contas cujo armazenamento está no escopo	Gratu	Resu	Y	-
Bytes da versão atual	CurrentVersionStorageBytes	O número de bytes da versão atual de um objeto	Gratu	Otim o de custo	N	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
% de bytes da versão atual	-	A porcentagem de bytes no escopo de versões atuais de objetos	Gratuito	Otimizado de custo	Yes	$\text{sum}(\text{CurrentVersionStorageBytes}) / \text{sum}(\text{StorageBytes})$
Contagem de objetos de versões atuais	CurrentVersionObjectCount	A contagem dos objetos de versão atual	Gratuito	Otimizado de custo	No	-
% de objetos em versão atual	-	A porcentagem de objetos no escopo que são de uma versão atual	Gratuito	Otimizado de custo	Yes	$\text{sum}(\text{CurrentVersionObjectCount}) / \text{sum}(\text{ObjectCount})$
Bytes de versões desatualizadas	NonCurrentVersionStorageBytes	O número de bytes de versões desatualizadas	Gratuito	Otimizado de custo	No	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
% de bytes de versões desatualizadas	-	A porcentagem de bytes no escopo de versões desatualizadas	Gratuito	Otimizado de custo	Yes	$\frac{\text{sum(NonCurrentVersionStorageBytes)}}{\text{sum(StorageBytes)}}$
Contagem de objetos de versões desatualizadas	NonCurrentVersionObjectCount	A contagem de versões de objetos desatualizadas	Gratuito	Otimizado de custo	No	-
% de objetos de versões desatualizadas	-	A porcentagem de objetos no escopo que são uma versão não atual	Gratuito	Otimizado de custo	Yes	$\frac{\text{sum(NonCurrentVersionObjectCount)}}{\text{sum(ObjectCount)}}$
Bytes de marcadores de exclusão	DeleteMarkerStorageBytes	O número de bytes no escopo que são marcadores de exclusão	Gratuito	Otimizado de custo	No	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
% de bytes de marcadores de exclusão	-	A porcentagem de bytes no escopo que são marcadores de exclusão	Gratuito	Otimizado de custo	Y	$\frac{\text{sum(DeleteMarkerStorageBytes)}}{\text{sum(StorageBytes)}}$
Contagem de objetos de marcadores de exclusão	DeleteMarkerObjectCount	O número total de objetos com um marcador de exclusão	Gratuito	Otimizado de custo	N	-
% de objetos de marcadores de exclusão	-	A porcentagem de objetos no escopo com um marcador de exclusão	Gratuito	Otimizado de custo	Y	$\frac{\text{sum(DeleteMarkerObjectCount)}}{\text{sum(ObjectCount)}}$
Bytes de multipart uploads incompletos	IncompleteMultipartUploadStorageBytes	O total de bytes no escopo para multipart uploads incompletos	Gratuito	Otimizado de custo	N	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
% de bytes de multipart uploads incompletos	-	A porcentagem de bytes no escopo que são o resultado de multipart uploads incompletos	Gratuito	Otimizado de custo	Y	$\text{sum}(\text{IncompleteMultipartUploadStorageBytes}) / \text{sum}(\text{StorageBytes})$
Contagem de objetos de multipart uploads incompletos	IncompleteMultipartUploadObjectCount	O número de objetos no escopo com multipart uploads incompletos	Gratuito	Otimizado de custo	N	-
% de objetos de multipart uploads incompletos	-	A porcentagem de objetos no escopo com multipart uploads incompletos	Gratuito	Otimizado de custo	Y	$\text{sum}(\text{IncompleteMultipartUploadObjectCount}) / \text{sum}(\text{ObjectCount})$
Bytes de armazenamento de multipart uploads incompletos há mais de 7 dias	IncompleteMPUStorageBytesOlderThan7Days	O total de bytes no escopo para multipart uploads incompletos há mais de 7 dias	Gratuito	Otimizado de custo	N	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
% de bytes de armazenamento de multipart uploads incompletos há mais de 7 dias	-	A porcentagem de bytes para multipart uploads incompletos há mais de 7 dias	Gratuito	Otimizado de custo	Y	$\text{sum}(\text{IncompleteMPUStorageBytesOlderThan7Days}) / \text{sum}(\text{StorageBytes})$
Contagem de objetos de multipart uploads incompletos há mais de 7 dias	IncompleteMPUObjectCountOlderThan7Days	O número de objetos que são o resultado de multipart uploads incompletos há mais de 7 dias	Gratuito	Otimizado de custo	N	-
% da contagem de objetos de multipart uploads incompletos há mais de 7 dias	-	A porcentagem de objetos que são o resultado de multipart uploads incompletos há mais de 7 dias	Gratuito	Otimizado de custo	Y	$\text{sum}(\text{IncompleteMPUObjectCountOlderThan7Days}) / \text{sum}(\text{ObjectCount})$
Contagem de regras de ciclo de vida para transição	TransitionLifecycleRuleCount	A contagem de regras de ciclo de vida para transição de objetos para outra classe de armazenamento	Avançado (Avançado)	Otimizado de custo	N	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
Média de regras de ciclo de vida para transição por bucket	-	O número médio de regras de ciclo de vida para transição de objetos para outra classe de armazenamento	Avançado (Avançado)	Otimização de custos	Y	$\text{sum(TransitionLifecycleRuleCount) / sum(DistinctNumberOfBuckets)}$
Contagem de regras de ciclo de vida para validade	ExpirationLifecycleRuleCount	A contagem de regras de ciclo de vida para expirar objetos	Avançado (Avançado)	Otimização de custos	N	-
Média de regras de ciclo de vida para validade por bucket	-	O número médio de regras de ciclo de vida para expirar objetos	Avançado (Avançado)	Otimização de custos	Y	$\text{sum(ExpirationLifecycleRuleCount) / sum(DistinctNumberOfBuckets)}$
Contagem de regras de ciclo de vida para transição de versões desatualizadas	NoncurrentVersionTransitionLifecycleRuleCount	A contagem de regras de ciclo de vida para transição de versões objetos desatualizadas para outra classe de armazenamento	Avançado (Avançado)	Otimização de custos	N	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
Média de regras de ciclo de vida para transição de versões desatualizadas por bucket	-	O número médio de regras de ciclo de vida para transição de versões objetos desatualizadas para outra classe de armazenamento	Avançado (Avançado)	Otimização de custos	Y	$\text{sum}(\text{NoncurrentVersionTransitionLifecycleRuleCount}) / \text{sum}(\text{DistinctNumberOfBuckets})$
Contagem de regras de ciclo de vida para validade de versões desatualizadas	NoncurrentVersionExpirationLifecycleRuleCount	A contagem de regras de ciclo de vida para expirar versões de objetos desatualizadas	Avançado (Avançado)	Otimização de custos	N	-
Média de regras de ciclo de vida para validade de versões desatualizadas por bucket	-	O número médio de regras de ciclo de vida para expirar versões de objetos desatualizadas	Avançado (Avançado)	Otimização de custos	Y	$\text{sum}(\text{NoncurrentVersionExpirationLifecycleRuleCount}) / \text{sum}(\text{DistinctNumberOfBuckets})$

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
Contagem de regras de ciclo de vida para anulação de multipart uploads incompletos	AbortIncompleteMPULifecycleRuleCount	A contagem de regras de ciclo de vida para excluir multipart uploads incompletos	Avançado	Otimização de custos	Não	-
Média de regras de ciclo de vida para anular multipart uploads incompletos por bucket	-	O número médio de regras de ciclo de vida para excluir multipart uploads incompletos	Avançado	Otimização de custos	Yes	$\text{sum}(\text{AbortIncompleteMPULifecycleRuleCount}) / \text{sum}(\text{DistinctNumberOfBuckets})$
Contagem de regras de ciclo de vida para marcadores de exclusão de objetos expirados	ExpiredObjectDeleteMarkerLifecycleRuleCount	A contagem de regras de ciclo de vida para remover os marcadores de exclusão de objetos expirados	Avançado	Otimização de custos	Não	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
Média de regras de ciclo de vida de marcadores de exclusão de objetos expirados por bucket	-	O número médio de regras de ciclo de vida para remover os marcadores de exclusão de objetos expirados	Avançado	Otimização de custos	Y	$\text{sum}(\text{ExpiredObjectDeleteMarkerLifecycleRuleCount}) / \text{sum}(\text{DistinctNumberOfBuckets})$
Contagem total de regras de ciclo de vida	TotalLifecycleRuleCount	A contagem total de regras de ciclo de vida	Avançado	Otimização de custos	N	-
Contagem média de regras de ciclo de vida por bucket	-	O número médio de regras de ciclo de vida	Avançado	Otimização de custos	Y	$\text{sum}(\text{TotalLifecycleRuleCount}) / \text{sum}(\text{DistinctNumberOfBuckets})$
Bytes criptografados	EncryptedStorageBytes	O número total de bytes criptografados	Gratuito	Proteção de dados	N	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Cate 2	De	Fórm da métr deriv
% de bytes criptografados	-	A porcentagem do total de bytes criptografados	Gratu	Prote de dado	Y	$\text{sum(EncryptedObjectCount)} / \text{sum(StorageBytes)}$
Contagem de objetos criptografados	Encrypted ObjectCount	A contagem total de objetos criptografados	Gratu	Prote de dado	N	-
% de objetos criptografados	-	A porcentagem de objetos criptografados	Gratu	Prote de dado	Y	$\text{sum(EncryptedStorageBytes)} / \text{sum(ObjectCount)}$
Bytes não criptografados	UnencryptedStorage Bytes	O número de bytes não criptografados	Gratu	Prote de dado	Y	$\text{sum(StorageBytes)} - \text{sum(EncryptedStorageBytes)}$
% de bytes não criptografados	-	A porcentagem de bytes não criptografados	Gratu	Prote de dado	Y	$\text{sum(UnencryptedStorageBytes)} / \text{sum(StorageBytes)}$

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
Contagem de objetos não criptografados	UnencryptedObjectCount	A contagem total de objetos não criptografados	Gratuito	Proteção de dados	Y	$\text{sum}(\text{ObjectCount}) - \text{sum}(\text{EncryptedObjectCount})$
% de objetos não criptografados	-	A porcentagem de objetos não criptografados	Gratuito	Proteção de dados	Y	$\frac{\text{sum}(\text{UnencryptedStorageBytes})}{\text{sum}(\text{ObjectCount})}$
Bytes de armazenamento replicados da origem	ReplicatedStorageBytesSource	O número total de bytes replicados do bucket de origem	Gratuito	Proteção de dados	N	-
% de bytes replicados da origem	-	A porcentagem do total de bytes replicados do bucket de origem	Gratuito	Proteção de dados	Y	$\frac{\text{sum}(\text{ReplicatedStorageBytesSource})}{\text{sum}(\text{StorageBytes})}$
Contagem de objetos replicados da origem	ReplicatedObjectCountSource	A contagem de objetos replicados do bucket de origem	Gratuito	Proteção de dados	N	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
% de objetos replicados da origem	-	A porcentagem do total de objetos replicados do bucket de origem	Gratuito	Proteção de dados	Y	$\text{sum}(\text{ReplicatedStorageObjectCount}) / \text{sum}(\text{ObjectCount})$
Bytes de armazenamento de replicação do destino	ReplicatedStorageBytes	O número total de bytes replicados para o bucket de destino	Gratuito	Proteção de dados	Y	-
% de bytes replicados para o destino	-	A porcentagem do total de bytes replicados para o bucket de destino	Gratuito	Proteção de dados	Y	$\text{sum}(\text{ReplicatedStorageBytes}) / \text{sum}(\text{StorageBytes})$
Contagem de objetos replicados para o destino	ReplicatedObjectCount	A contagem de objetos replicados para o bucket de destino	Gratuito	Proteção de dados	Y	-
% de objetos replicados para o destino	-	A porcentagem do total de objetos replicados para o bucket de destino	Gratuito	Proteção de dados	Y	$\text{sum}(\text{ReplicatedObjectCount}) / \text{sum}(\text{ObjectCount})$

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
Bytes do Bloqueio de Objeto	ObjectLockEnabledStorageBytes	A contagem total de bytes de armazenamento com Bloqueio de Objeto ativado	Gratuito	Proteção de dados	Y	$\text{sum}(\text{UnencryptedStorageBytes}) / \text{sum}(\text{ObjectLockEnabledStorageCount}) - \text{sum}(\text{ObjectLockEnabledStorageBytes})$
% de bytes do Bloqueio de Objeto	-	A porcentagem de bytes de armazenamento com Bloqueio de Objeto ativado	Gratuito	Proteção de dados	Y	$\text{sum}(\text{ObjectLockEnabledStorageBytes}) / \text{sum}(\text{StorageBytes})$
Contagem de objetos com Bloqueio de Objeto	ObjectLockEnabledObjectCount	A contagem total de objetos com Bloqueio de Objeto	Gratuito	Proteção de dados	Y	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
% de objetos com Bloqueio de Objeto	-	A porcentagem do total de objetos com Bloqueio de Objeto ativado	Gratuito	Proteção de dados	Y	$\text{sum}(\text{ObjectLockEnabledObjectCount}) / \text{sum}(\text{ObjectCount})$
Contagem de buckets com Versionamento ativado	VersioningEnabledBucketCount	A contagem de buckets com o Versionamento do S3 ativado	Gratuito	Proteção de dados	N	-
% de bucket com Versionamento ativado	-	A porcentagem de buckets com o Versionamento do S3 ativado	Gratuito	Proteção de dados	Y	$\text{sum}(\text{VersioningEnabledBucketCount}) / \text{sum}(\text{DistinctNumberOfBuckets})$
Contagem de buckets com exclusão de MFA ativada	MFADeleteEnabledBucketCount	A contagem de buckets com exclusão de MFA (autenticação multifator) ativada	Gratuito	Proteção de dados	N	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
% de buckets com exclusão de MFA ativada	-	A porcentagem de buckets com exclusão de MFA (autenticação multifator) ativada	Gratuito	Proteção de dados	Y	$\frac{\text{sum(MFADeleteEnabledBucketCount)}}{\text{sum(DistinctNumberOfBuckets)}}$
Contagem de buckets com SSE-KMS ativada	SSEKMSEnabledBucketCount	A contagem de buckets que usam criptografia do lado do servidor com chaves do AWS Key Management Service (SSE-KMS) para criptografia padrão	Gratuito	Proteção de dados	N	-
% de buckets com SSE-KMS ativada	-	A porcentagem de buckets que usam SSE-KMS para criptografia padrão	Gratuito	Proteção de dados	Y	$\frac{\text{sum(SSEKMSEnabledBucketCount)}}{\text{sum(DistinctNumberOfBuckets)}}$

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
Todas as solicitações de assinatura incompatíveis	AllUnsupportedSignatureRequests	O número total de solicitações que usam versões de assinatura da AWS incompatíveis	Avançado (Avançado)	Proteção de dados	Não	-
% de todas as solicitações de assinatura incompatíveis	-	A porcentagem de solicitações que usam versões de assinatura da AWS incompatíveis	Avançado (Avançado)	Proteção de dados	Sim	$\frac{\text{sum}(\text{AllUnsupportedSignatureRequests})}{\text{sum}(\text{AllRequests})}$
Todas as solicitações TLS incompatíveis	AllUnsupportedTLRequests	O número de solicitações que usam versões de Transport Layer Security (TLS) incompatíveis	Avançado (Avançado)	Proteção de dados	Não	-
% de todas as solicitações TLS incompatíveis	-	A porcentagem de solicitações que usam versões de TLS incompatíveis	Avançado (Avançado)	Proteção de dados	Sim	$\frac{\text{sum}(\text{AllUnsupportedTLRequests})}{\text{sum}(\text{AllRequests})}$

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
Todas as solicitações de SSE-KMS	AllSSEKMSRequests	O número total de solicitações que especificam SSE-KMS	Avançado (Avançado)	Proteção de dados	Não	-
% de todas as solicitações de SSE-KMS	-	A porcentagem de solicitações que especificam SSE-KMS	Avançado (Avançado)	Proteção de dados	Y	$\frac{\text{sum}(\text{AllSSEKMSRequests})}{\text{sum}(\text{AllRequests})}$
Contagem de regras de replicação na mesma região	SameRegionReplicationRuleCount	A contagem de regras para replicação na mesma região (SRR)	Avançado (Avançado)	Proteção de dados	Não	-
Média de regras para replicação na mesma região por bucket	-	O número médio de regras de replicação para SRR	Avançado (Avançado)	Proteção de dados	Y	$\frac{\text{sum}(\text{SameRegionReplicationRuleCount})}{\text{sum}(\text{DistinctNumberOfBuckets})}$
Contagem de regras de replicação entre regiões	CrossRegionReplicationRuleCount	A contagem de regras para replicação entre regiões diferentes (CRR)	Avançado (Avançado)	Proteção de dados	Não	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
Média de regras para replicação entre regiões diferentes por bucket	-	O número médio de regras de replicação para CRR	Avançado (Avançado)	Proteção de dados	Y	$\frac{\text{sum}(\text{CrossRegionReplicationRuleCount})}{\text{sum}(\text{DistinctNumberOfBuckets})}$
Contagem de regras de replicação na mesma conta	SameAccountReplicationRuleCount	A contagem de regras para replicação na mesma conta	Avançado (Avançado)	Proteção de dados	N	-
Média de regras para replicação da mesma conta por bucket	-	O número médio de regras para replicação na mesma conta	Avançado (Avançado)	Proteção de dados	Y	$\frac{\text{sum}(\text{SameAccountReplicationRuleCount})}{\text{sum}(\text{DistinctNumberOfBuckets})}$
Contagem de regras de replicação entre contas	CrossAccountReplicationRuleCount	A contagem de regras para replicação entre contas diferentes	Avançado (Avançado)	Proteção de dados	N	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
Média de regras para replicação entre contas diferentes por bucket	-	O número médio de regras para replicação entre contas diferentes	Avançado (Avançado)	Proteção de dados	Y	$\text{sum}(\text{CrossAccountReplicationRuleCount}) / \text{sum}(\text{DistinctNumberOfBuckets})$
Contagem de regras de replicação de destino inválido	InvalidDestinationReplicationRuleCount	A contagem de regras de replicação com um destino de replicação que não é válido	Avançado (Avançado)	Proteção de dados	N	-
Média de regras de replicação de destino inválido por bucket	-	O número médio de regras de replicação com um destino de replicação que não é válido	Avançado (Avançado)	Proteção de dados	Y	$\text{sum}(\text{InvalidReplicationRuleCount}) / \text{sum}(\text{DistinctNumberOfBuckets})$
Contagem total de regras de replicação	-	A contagem total de regras de replicação	Avançado (Avançado)	Proteção de dados	Y	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Disponível	Fórmula da métrica derivada
Média de contagem de regras de replicação por bucket	-	A média de contagem total de regras de replicação	Avançado (Avançado)	Proteção de dados	Y	$\text{sum}(\text{all replication rule count metrics}) / \text{sum}(\text{DistinctNumber Of Buckets})$
Contagem de buckets com Imposto pelo proprietário do bucket em Propriedade de Objeto	ObjectOwnershipBucketOwnerEnforcedBucketCount	A contagem total de buckets com listas de controle de acesso (ACLs) desabilitadas pela configuração Imposto pelo proprietário do bucket para Propriedade de Objeto	Gratuito	Gestão de acesso	N	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Cate 2	De	Fórm da métr deriv
% de buckets com Imposto pelo proprietário do bucket para Propriedade de Objeto	-	A porcentagem de buckets com ACLs desabilitadas pela configuração Imposto pelo proprietário do bucket para Propriedade de Objeto	Gratu	Gere	Y	sum(ObjectOwnershipBucketOwnerEnforcedBucketCount)/sum(DistinctNumberOfBuckets)
Contagem de buckets com Proprietário do bucket preferido para Propriedade de Objeto	ObjectOwnershipBucketOwnerPreferredBucketCount	A contagem total de buckets que usam a configuração Proprietário do bucket preferido para Propriedade de Objeto	Gratu	Gere	N	-
% de buckets com Proprietário do bucket preferido para Propriedade de Objeto	-	A porcentagem de buckets que usam a configuração Proprietário do bucket preferido para Propriedade de Objeto	Gratu	Gere	Y	sum(ObjectOwnershipBucketOwnerPreferredBucketCount)/sum(DistinctNumberOfBuckets)

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
Contagem de buckets com Autor do objeto para Propriedade de Objeto	ObjectOwnershipObjectWriterBucketCount	A contagem total de buckets que usam a configuração Autor do objeto para Propriedade de Objeto	Gratuito	Gerenciamento de acesso	Não	-
% de buckets com Autor do objeto para Propriedade de Objeto	-	A porcentagem de buckets que usam a configuração Autor do objeto para Propriedade de Objeto	Gratuito	Gerenciamento de acesso	Yes	$\text{sum}(\text{ObjectOwnershipObjectWriterBucketCount}) / \text{sum}(\text{DistinctNumberOfBuckets})$
Contagem de buckets com Aceleração de Transferência ativada	TransferAccelerationEnabledBucketCount	A contagem total de buckets com Aceleração de Transferência ativada	Gratuito	Desempenho	Não	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Disponibilidade	Fórmula da métrica derivada
% de buckets com Aceleração de Transferência ativada	-	A porcentagem de buckets com Aceleração de Transferência ativada	Gratuito	Desempenho	Y	$\text{sum}(\text{TransferAccelerationEnabledBucketCount}) / \text{sum}(\text{DistinctNumberOfBuckets})$
Contagem de buckets com Notificações de Eventos ativadas	EventNotificationEnabledBucketCount	A contagem total de buckets com Notificações de Eventos ativadas	Gratuito	Eventos	N	
% de buckets com Notificações de Eventos ativadas	-	A porcentagem de buckets com Notificações de Eventos ativadas	Gratuito	Eventos	Y	$\text{sum}(\text{EventNotificationEnabledBucketCount}) / \text{sum}(\text{DistinctNumberOfBuckets})$
Todas as solicitações	AllRequests	O número total de solicitações feitas	Avançado (Avançado)	Atividade	N	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Cate 2	De	Fórm da métr deriv
Solicitações GET	GetRequests	O número total de solicitações GET feitas	Avan (Avan)	Ativi	NI	-
Solicitações PUT	PutRequests	O número total de solicitações PUT feitas	Avan (Avan)	Ativi	NI	-
Solicitações HEAD	HeadRequests	O número total de solicitações HEAD feitas	Avan (Avan)	Ativi	NI	-
Solicitações DELETE	DeleteRequests	O número total de solicitações DELETE feitas	Avan (Avan)	Ativi	NI	-
Solicitações LIST	ListRequests	O número total de solicitações LIST feitas	Avan (Avan)	Ativi	NI	-
Solicitações POST	PostRequests	O número total de solicitações POST feitas	Avan (Avan)	Ativi	NI	-
Solicitações SELECT	SelectRequests	O número total de solicitações do S3 Select	Avan (Avan)	Ativi	NI	-
Bytes verificados do Select	SelectScannedBytes	O número de bytes verificados do S3 Select	Avan (Avan)	Ativi	NI	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
Bytes devolvidos do Select	SelectReturnedBytes	O número de bytes devolvidos do S3 Select	Avançado (Avançado)	Atividade	Não	-
Bytes baixados	BytesDownloaded	O número de bytes baixados	Avançado (Avançado)	Atividade	Não	-
% de taxa de recuperação	-	A porcentagem de bytes baixados	Avançado (Avançado)	Atividade	Y	$\frac{\text{sum}(\text{BytesDownloaded})}{\text{sum}(\text{StorageBytes})}$
Bytes carregados	BytesUploaded	O número de bytes enviados por upload	Avançado (Avançado)	Atividade	Não	-
% de relação de consumo	-	A porcentagem de bytes carregados	Avançado (Avançado)	Atividade	Y	$\frac{\text{sum}(\text{BytesUploaded})}{\text{sum}(\text{StorageBytes})}$
Erros 4xx	4xxErrors	O número total de códigos de status HTTP 4xx	Avançado (Avançado)	Atividade	Não	-
Erros 5xx	5xxErrors	O número total de códigos de status HTTP 5xx	Avançado (Avançado)	Atividade	Não	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
Total de erros	-	A soma de todos os erros 4xx e 5xx	Avançado (Avançado)	Atividade	Y	$\text{sum}(4xx\text{Errors}) + \text{sum}(5xx\text{Errors})$
% de taxa de erros	-	O número total de erros 4xx e 5xx como uma porcentagem do total de solicitações	Avançado (Avançado)	Atividade	Y	$\frac{\text{sum}(\text{Total Errors})}{\text{sum}(\text{Total Requests})}$
Contagem de status 200 OK	200OKStatusCount	A contagem total de códigos de status 200 OK	Avançado (Avançado)	Código de status detalhado	N	-
% de status 200 OK	-	O número total de códigos de status 200 OK como uma porcentagem do total de solicitações	Avançado (Avançado)	Código de status detalhado	Y	$\frac{\text{sum}(200\text{OK StatusCount})}{\text{sum}(\text{AllRequests})}$
Contagem de status 206 Conteúdo Parcial	206PartialContentStatusCount	A contagem total de códigos de status 206 Conteúdo Parcial	Avançado (Avançado)	Código de status detalhado	N	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
% de status 206 Conteúdo Parcial	-	O número total de códigos de status 206 Conteúdo Parcial como uma porcentagem do total de solicitações	Avançado (Avançado)	Código de status detalhado	Y	$\text{sum}(206\text{PartialContentStatusCount}) / \text{sum}(\text{AllRequests})$
Contagem de erros 400 Solicitação Inválida	400BadRequestErrorCount	A contagem total de códigos de status 400 Solicitação Inválida	Avançado (Avançado)	Código de status detalhado	N	-
% de erros 400 Solicitação Inválida	-	O número total de códigos de status 400 Solicitação Inválida como uma porcentagem do total de solicitações	Avançado (Avançado)	Código de status detalhado	Y	$\text{sum}(400BadRequestErrorCount) / \text{sum}(\text{AllRequests})$
Contagem de erros 403 Proibido	403ForbiddenErrorCount	A contagem total de códigos de status 403 Proibido	Avançado (Avançado)	Código de status detalhado	N	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
% de erros 403 Proibido	-	O número total de códigos de status 403 Proibido como uma porcentagem do total de solicitações	Avançado (Avançado)	Código de status detalhado	Y	$\frac{\text{sum}(403ForbiddenErrorCount)}{\text{sum}(AllRequests)}$
Contagem de erros 404 Não Encontrado	404NotFoundErrorCount	A contagem total de códigos de status 404 Não Encontrado	Avançado (Avançado)	Código de status detalhado	N	-
% de erros 404 Não Encontrado	-	O número total de códigos de status 404 Não Encontrado como uma porcentagem do total de solicitações	Avançado (Avançado)	Código de status detalhado	Y	$\frac{\text{sum}(404NotFoundErrorCount)}{\text{sum}(AllRequests)}$
Contagem de erros 500 Erro Interno do Servidor	500InternalServerErrorCount	A contagem total de códigos de status 500 Erro Interno do Servidor	Avançado (Avançado)	Código de status detalhado	N	-

Nome da métrica	CloudWatch e exportação	Descrição	Nível	Categoria ²	Dimensão	Fórmula da métrica derivada
% de erros 500 Erro Interno do Servidor	-	O número total de códigos de status 500 Erro Interno do Servidor como uma porcentagem do total de solicitações	Avançado (Avançado)	Código de status detalhado	Y	$\text{sum}(500\text{InternalServerErrorCount}) / \text{sum}(\text{AllRequests})$
Contagem de erros 503 Serviço Indisponível	503ServiceUnavailableErrorCount	A contagem total de códigos de status 503 Serviço Indisponível	Avançado (Avançado)	Código de status detalhado	N	-
% de erros 503 Serviço Indisponível	-	O número total de códigos de status 503 Serviço Indisponível como uma porcentagem do total de solicitações	Avançado (Avançado)	Código de status detalhado	Y	$\text{sum}(503ServiceUnavailableErrorCount) / \text{sum}(\text{AllRequests})$

¹ Todas as métricas de armazenamento gratuito estão disponíveis no nível do grupo de Lente de Armazenamento. As métricas de nível avançado não estão disponíveis no nível do grupo de Lente de Armazenamento.

² As métricas de contagem de regras e de configurações de bucket não estão disponíveis no nível do prefixo.

Trabalhar com a Lente de Armazenamento do Amazon S3 usando o console e a API

A Lente de Armazenamento do Amazon S3 é um recurso de análise de armazenamento em nuvem que você pode usar para obter visibilidade em toda a organização sobre o uso e a atividade do armazenamento de objetos. Use as métricas da Lente de Armazenamento do S3 para gerar insights resumidos, como descobrir quanto armazenamento você tem em toda a organização ou quais são os buckets e prefixos que mais crescem. Você também pode usar as métricas da Lente de Armazenamento do S3 para identificar oportunidades de otimização de custos, implementar práticas recomendadas de proteção e segurança de dados, e melhorar a performance das workloads de aplicações. Por exemplo, você pode identificar buckets que não têm regras de Ciclo de Vida do S3 para expirar multipart uploads incompletos com mais de 7 dias. Você também pode identificar buckets que não estão seguindo as práticas recomendadas de proteção de dados, como usar a Replicação do S3 ou o Versionamento do S3. A Lente de Armazenamento do S3 também analisa as métricas para fornecer recomendações contextuais que você pode usar para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados.

A Lente de Armazenamento do S3 agrega suas métricas e exibe as informações na seção Account snapshot (Snapshot da conta) na página Buckets do console do Amazon S3. A Lente de Armazenamento do S3 também oferece um painel interativo 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. O painel tem opções de detalhamento para gerar e visualizar insights no nível da organização, conta, Região da AWS, classe de armazenamento, bucket, prefixo ou grupo de Lente de Armazenamento. Você também pode enviar uma exportação diária de métricas em CSV ou no formato Parquet para um bucket do S3.

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](#)
- [Exemplos da Lente de Armazenamento do Amazon S3 usando a AWS CLI](#)
- [Exemplos do Amazon S3 Storage Lens usando o SDK for Java](#)

Usar o Amazon S3 Storage Lens no console

A Lente de Armazenamento do Amazon S3 é um recurso de análise de armazenamento em nuvem que você pode usar para obter visibilidade em toda a organização sobre o uso e a atividade do armazenamento de objetos. Use as métricas da Lente de Armazenamento do S3 para gerar insights resumidos, como descobrir quanto armazenamento você tem em toda a organização ou quais são os buckets e prefixos que mais crescem. Você também pode usar as métricas da Lente de Armazenamento do S3 para identificar oportunidades de otimização de custos, implementar práticas recomendadas de proteção e segurança de dados, e melhorar a performance das workloads de aplicações. Por exemplo, você pode identificar buckets que não têm regras de Ciclo de Vida do S3 para expirar multipart uploads incompletos com mais de 7 dias. Você também pode identificar buckets que não estão seguindo as práticas recomendadas de proteção de dados, como usar a Replicação do S3 ou o Versionamento do S3. A Lente de Armazenamento do S3 também analisa as métricas para fornecer recomendações contextuais que você pode usar para otimizar os custos de armazenamento e aplicar as práticas recomendadas de proteção de dados.

A Lente de Armazenamento do S3 agrega suas métricas e exibe as informações na seção Account snapshot (Snapshot da conta) na página Buckets do console do Amazon S3. A Lente de Armazenamento do S3 também oferece um painel interativo 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. O painel tem opções de detalhamento para gerar e visualizar insights no nível da organização, conta, Região da AWS, classe de armazenamento, bucket, prefixo ou grupo de Lente de Armazenamento. Você também pode enviar uma exportação diária de métricas em CSV ou no formato Parquet para um bucket do S3.

Note

As atualizações na configuração do painel podem levar até 48 horas para serem exibidas ou visualizadas com precisão.

Tópicos

- [Criação e atualização dos painéis do Amazon S3 Storage Lens](#)
- [Desabilitar ou excluir painéis do Amazon S3 Storage Lens](#)
- [Trabalhar com AWS Organizations para criar painéis no nível da organização](#)

Criação e atualização dos painéis do Amazon S3 Storage Lens

A Lente de Armazenamento do S3 agrega suas métricas e exibe as informações na seção Account snapshot (Snapshot da conta) na página Buckets do console do Amazon S3. A Lente de Armazenamento do S3 também oferece um painel interativo 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. O painel tem opções de detalhamento para gerar e visualizar insights no nível da organização, conta, Região da AWS, classe de armazenamento, bucket, prefixo ou grupo de Lente de Armazenamento. Você também pode enviar uma exportação diária de métricas em CSV ou no formato Parquet para um bucket do S3.

O painel padrão do Amazon S3 Storage Lens é o painel de conta padrão. Esse painel é pré-configurado pelo Amazon S3 para ajudar você a visualizar tendências e insights resumidos para as métricas gratuitas e avançadas agregadas de toda a sua conta no console. Você não pode modificar o escopo de configuração padrão do painel, mas pode atualizar a seleção de métricas das métricas gratuitas para as métricas e recomendações avançadas pagas, configurar a exportação opcional de métricas ou até mesmo desativar o painel padrão. O painel padrão não pode ser excluído.

Você também pode criar painéis personalizados adicionais da Lente de Armazenamento do S3 que podem ser colocados no escopo de sua organização no AWS Organizations ou em regiões ou buckets específicos em uma conta.

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.

Etapa 1: Definir o escopo do painel

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 navegação na parte superior da página, escolha o nome da região da AWS exibida no momento. Em seguida, escolha a região para a qual você deseja mudar.
3. No painel de navegação esquerdo, em Lente de Armazenamento do S3, selecione Painéis.
4. Escolha Create dashboard (Criar painel).
5. Na página Dashboard (Painel), na seção General (Geral), faça o seguinte:
 - a. Visualize a Região de origem para seu painel. A região de origem é a Região da AWS em que a configuração e as métricas referentes a esse painel do Lente de Armazenamento estão armazenadas.

- b. 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.

- 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.

6. 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.

Etapa 2: Configurar a seleção de métricas

1. Na seção Metrics selection (Seleção de métricas), escolha o tipo de métricas que você deseja agregar para este painel.
 - Para incluir métricas gratuitas agregadas por bucket e disponíveis para consultas por 14 dias, escolha Free metrics (Métricas gratuitas).
 - Para ativar métricas avançadas e outras opções avançadas, escolha Advanced metrics and recommendations (Métricas e recomendações avançadas). Essas opções incluem agregação avançada de prefixos, publicação do Amazon CloudWatch e recomendações contextuais. Os dados ficam disponíveis para consultas por 15 meses. As métricas e recomendações avançadas têm um custo adicional. Para obter mais informações, consulte [Definição de preço do Amazon S3](#).

Para obter mais informações sobre métricas avançadas e métricas gratuitas, consulte [Seleção de métricas](#).

2. Em Advanced metrics and recommendations features (Recursos de métricas e recomendações avançadas), selecione as opções que deseja habilitar:
 - Advanced metrics (Métricas avançadas)
 - Publicação no CloudWatch
 - Agregação de prefixo

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.

3. Se você ativou Advanced metrics (Métricas avançadas), selecione as Advanced metrics categories (Categorias de métricas avançadas) que deseja exibir no painel da Lente de Armazenamento do S3:
 - Métricas de atividade
 - Detailed status code metrics (Métricas detalhadas do código de status)
 - Advanced cost optimization metrics (Métricas avançadas de otimização de custos)

- Advanced data protection metrics (Métricas avançadas de proteção de dados)

Para obter mais informações sobre categorias de métricas, consulte [Categorias de métricas](#). Para obter uma lista completa de métricas, consulte [Glossário de métricas de lente de armazenamento do Amazon S3](#).

4. Caso tenha optado por habilitar a agregação de prefixos, configure o seguinte:

- a. Escolha o tamanho mínimo de limite de prefixo para este painel.

Por exemplo, um limite de prefixo de 5% indica que serão agregados os prefixos que compõem 5% ou mais do tamanho total 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 valor é 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.

(Opcional) Etapa 3: 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 acontece 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 a exportação de métricas foi ativada, escolha o formato de saída da exportação diária: 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 bucket do S3 de destino (formato: `s3://bucket-name/prefix`).

O bucket deve estar na região inicial do seu painel da Lente de Armazenamento do S3. O console do S3 mostra a Destination bucket permission (Permissão do bucket de destino) que será adicionada pelo Amazon S3 à política do bucket de destino. O Amazon S3 atualiza a política no bucket de destino para permitir que o S3 coloque dados nesse bucket.

5. (Opcional) Para habilitar a criptografia do lado do servidor para a exportação de métricas, escolha Specify an encryption key (Especificar uma chave de criptografia). Depois, escolha Tipo de criptografia: Chaves gerenciadas do Amazon S3 (SSE-S3) ou Chave do AWS Key Management Service (SSE-KMS).

Você pode escolher entre uma [chave gerenciada pelo Amazon S3](#) (SSE-S3) e uma [chave do AWS Key Management Service \(AWS KMS\)](#) (SSE-KMS).

6. (Opcional) Para especificar uma chave do AWS KMS, você deve escolher uma chave do KMS ou inserir o nome do recurso da Amazon (ARN) de uma chave.

Se escolher uma chave gerenciada pelo cliente, você deverá conceder permissão de criptografia à Lente de Armazenamento do S3 na política de chaves do AWS KMS. Para ter mais informações, consulte [Usar uma AWS KMS key para criptografar suas exportações de métricas](#).

7. Escolha Create dashboard (Criar painel).

Para obter mais visibilidade do seu armazenamento, você pode criar um ou mais grupos da Lente de Armazenamento do S3 e anexá-los ao seu painel. Um grupo da Lente de Armazenamento do S3 é um filtro personalizado definido para objetos com base em prefixos, sufixos, tags de objeto, tamanho do objeto, idade do objeto ou uma combinação desses filtros.

Você pode usar grupos da Lente de Armazenamento do S3 para obter visibilidade granular de grandes buckets compartilhados, como data lakes, para tomar decisões de negócios mais bem informadas. Por exemplo, você pode simplificar a alocação de armazenamento e otimizar os relatórios de custos dividindo o uso do armazenamento em grupos específicos de objetos para projetos individuais e centros de custo dentro de um bucket ou em vários buckets.

Para usar grupos da Lente de Armazenamento do S3, você deve atualizar o painel para usar métricas e recomendações avançadas. Para obter mais informações sobre grupos de Lente

de Armazenamento do S3, consulte [the section called “Trabalho com grupos de Lente de Armazenamento do S3”](#).

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.

Etapa 1: Atualizar o escopo do painel

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 Storage Lens, Dashboards (Lente de Armazenamento, Painéis).
3. Escolha o painel que você deseja editar e escolha Edit (Editar).

A página Edit dashboard (Editar painel) é aberta.

Note

Você não pode alterar o seguinte:

- O nome do painel
- A região inicial
- O escopo do painel padrão, que tem escopo para o armazenamento de toda a sua conta


4. (Opcional) Na página de configuração do painel, na seção General (Geral), atualize e adicione etiquetas 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. 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.

- b. 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.

Etapa 2: Atualizar a seleção de métricas

1. Na seção Metrics selection (Seleção de métricas), escolha o tipo de métricas que você deseja agregar para este painel.
 - Para incluir métricas gratuitas agregadas por bucket e disponíveis para consultas por 14 dias, escolha Free metrics (Métricas gratuitas).
 - Para ativar métricas avançadas e outras opções avançadas, escolha Advanced metrics and recommendations (Métricas e recomendações avançadas). Essas opções incluem agregação avançada de prefixos, publicação do Amazon CloudWatch e recomendações contextuais. Os dados ficam disponíveis para consultas por 15 meses. As métricas e recomendações avançadas têm um custo adicional. Para obter mais informações, consulte [Definição de preço do Amazon S3](#).

Para obter mais informações sobre métricas avançadas e métricas gratuitas, consulte [Seleção de métricas](#).
2. Em Advanced metrics and recommendations features (Recursos de métricas e recomendações avançadas), selecione as opções que deseja habilitar:
 - Advanced metrics (Métricas avançadas)
 - Publicação no CloudWatch

- Agregação de prefixo

 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.

3. Se você ativou Advanced metrics (Métricas avançadas), selecione as Advanced metrics categories (Categorias de métricas avançadas) que deseja exibir no painel da Lente de Armazenamento do S3:

- Métricas de atividade
- Detailed status code metrics (Métricas detalhadas do código de status)
- Advanced cost optimization metrics (Métricas avançadas de otimização de custos)
- Advanced data protection metrics (Métricas avançadas de proteção de dados)

Para obter mais informações sobre categorias de métricas, consulte [Categorias de métricas](#). Para obter uma lista completa de métricas, consulte [Glossário de métricas de lente de armazenamento do Amazon S3](#).

4. Caso tenha optado por habilitar a agregação de prefixos, configure o seguinte:

- a. Escolha o tamanho mínimo de limite de prefixo para este painel.

Por exemplo, um limite de prefixo de 5% indica que serão agregados os prefixos que compõem 5% ou mais do tamanho total 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.

(Opcional) Etapa 3: 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. Para desativar a exportação de métricas, escolha Disable (Desativar).

A exportação de métricas acontece 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 ativada, escolha o formato de saída da exportação de métricas diária: 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 bucket do S3 de destino (formato: `s3://bucket-name/prefix`).


O bucket deve estar na região inicial do seu painel da Lente de Armazenamento do S3. O console do S3 mostra a Destination bucket permission (Permissão do bucket de destino) que será adicionada pelo Amazon S3 à política do bucket de destino. O Amazon S3 atualiza a política no bucket de destino para permitir que o S3 coloque dados nesse bucket.

5. (Opcional) Para habilitar a criptografia do lado do servidor para a exportação de métricas, escolha Specify an encryption key (Especificar uma chave de criptografia). Depois, escolha Tipo de criptografia: Chaves gerenciadas do Amazon S3 (SSE-S3) ou Chave do AWS Key Management Service (SSE-KMS).

Você pode escolher entre uma [chave gerenciada pelo Amazon S3](#) (SSE-S3) e uma [chave do AWS Key Management Service \(AWS KMS\)](#) (SSE-KMS).

6. (Opcional) Para especificar uma chave do AWS KMS, você deve escolher uma chave do KMS ou inserir o nome do recurso da Amazon (ARN) de uma chave. Em Chave do AWS KMS, especifique sua chave do KMS de uma das seguintes maneiras:
 - Para escolher entre uma lista de chaves do KMS disponíveis, selecione Escolher entre suas AWS KMS keys e escolha sua chave do KMS na lista de chaves disponíveis.

As chaves Chave gerenciada pela AWS (aws/s3) e as chaves gerenciadas pelo cliente são exibidas nessa lista. Para ter mais informações sobre chaves gerenciadas pelo cliente, consulte [Chaves de clientes e chaves da AWS](#) no Guia do desenvolvedor do AWS Key Management Service.

 Note

A Chave gerenciada pela AWS (aws/S3) não é compatível com a criptografia SSE-KMS com a Lente de Armazenamento do S3.

- Para inserir o ARN da chave do KMS, escolha Inserir ARN da AWS KMS key e insira o ARN da chave do KMS no campo exibido.
- Para criar uma chave gerenciada pelo cliente no console do AWS KMS, selecione Criar uma chave do KMS.

Se escolher uma chave gerenciada pelo cliente, você deverá conceder permissão de criptografia à Lente de Armazenamento do S3 na política de chaves do AWS KMS. Para ter mais informações, consulte [Usar uma AWS KMS key para criptografar suas exportações de métricas](#).

Para obter mais informações sobre como criar uma AWS KMS key, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

7. Escolha Salvar alterações.

Para obter mais visibilidade do seu armazenamento, você pode criar um ou mais grupos da Lente de Armazenamento do S3 e anexá-los ao seu painel. Um grupo da Lente de Armazenamento do S3 é um filtro personalizado definido para objetos com base em prefixos, sufixos, tags de objeto, tamanho do objeto, idade do objeto ou uma combinação desses filtros.

Você pode usar grupos da Lente de Armazenamento do S3 para obter visibilidade granular de grandes buckets compartilhados, como data lakes, para tomar decisões de negócios mais bem informadas. Por exemplo, você pode simplificar a alocação de armazenamento e otimizar os relatórios de custos dividindo o uso do armazenamento em grupos específicos de objetos para projetos individuais e centros de custo dentro de um bucket ou em vários buckets.

Para usar grupos da Lente de Armazenamento do S3, você deve atualizar o painel para usar métricas e recomendações avançadas. Para obter mais informações sobre grupos de Lente

de Armazenamento do S3, consulte [the section called “Trabalho com grupos de Lente de Armazenamento do S3”](#).

Desabilitar ou excluir painéis do Amazon S3 Storage Lens

A Lente de Armazenamento do S3 agrega suas métricas e exibe as informações na seção Account snapshot (Snapshot da conta) na página Buckets do console do Amazon S3. A Lente de Armazenamento do S3 também oferece um painel interativo 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. O painel tem opções de detalhamento para gerar e visualizar insights no nível da organização, conta, Região da AWS, classe de armazenamento, bucket, prefixo ou grupo de Lente de Armazenamento. Você também pode enviar uma exportação diária de métricas em CSV ou no formato Parquet para um bucket do S3.

O painel padrão do Amazon S3 Storage Lens é o painel de conta padrão. Esse painel é pré-configurado pelo Amazon S3 para ajudar você a visualizar tendências e insights resumidos para as métricas gratuitas e avançadas agregadas de toda a sua conta no console. Você não pode modificar o escopo de configuração padrão do painel, mas pode atualizar a seleção de métricas das métricas gratuitas para as métricas e recomendações avançadas pagas, configurar a exportação opcional de métricas ou até mesmo desativar o painel padrão. 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.

Os dados das seleções de métricas ficam disponíveis para consultas por 14 dias, e os dados de métricas e recomendações avançadas ficam disponíveis para consultas por 15 meses.

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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e 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 inserindo o nome do painel no campo de texto e escolha Confirmar.

Exclusão de um painel do Amazon S3 Storage Lens

Note

Você não pode excluir o painel padrão. No entanto, você poderá desabilitá-lo. Antes de excluir um painel que você criou, 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 ter mais informações, consulte [Desativação de um painel do Amazon S3 Storage Lens](#).
- A exclusão do painel excluirá todas as configurações associadas a ele.
- A exclusão de um painel torna 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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e 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 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

A Lente de Armazenamento do S3 agrega suas métricas e exibe as informações na seção Account snapshot (Snapshot da conta) na página Buckets do console do Amazon S3. A Lente de Armazenamento do S3 também oferece um painel interativo 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. O painel tem opções de detalhamento para gerar e visualizar insights no nível da organização, conta, Região da AWS, classe de armazenamento, bucket, prefixo ou grupo de Lente de Armazenamento. Você também pode enviar uma exportação diária de métricas em CSV ou no formato Parquet para um bucket do S3.

O painel padrão do Amazon S3 Storage Lens é o painel de conta padrão. Esse painel é pré-configurado pelo Amazon S3 para ajudar você a visualizar tendências e insights resumidos para as métricas gratuitas e avançadas agregadas de toda a sua conta no console. Você não pode modificar o escopo de configuração padrão do painel, mas pode atualizar a seleção de métricas das métricas gratuitas para as métricas e recomendações avançadas pagas, configurar a exportação opcional de métricas ou até mesmo desativar o painel padrão. 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.

Um painel do S3 Storage Lens fornece um rico recurso de informações sobre seu escopo de armazenamento. Um painel visualiza mais de 30 métricas que representam tendências e informações, incluindo resumo do 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, use o AWS Organizations e habilite o acesso confiável da Lente de Armazenamento do S3 utilizando 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.

Ativação do acesso confiável para o S3 Storage Lens em sua organização

A ativação do acesso confiável permite que a Lente de Armazenamento do Amazon S3 acesse sua hierarquia, associação e estrutura do AWS Organizations por meio de operações de API do AWS Organizations. A Lente de Armazenamento do S3 se torna um serviço confiável para toda a estrutura de sua organização. Ela pode criar perfis vinculados ao serviço 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 a Lente de Armazenamento do S3 colete métricas de uso de armazenamento entre contas e atividades para painéis dentro de contas em suas organizações.

Para ter mais informações, consulte [Usar funções vinculadas a serviços para o Amazon S3 Storage Lens](#).

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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e 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 a Lente de Armazenamento do S3 a funcionar apenas no 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 do proprietário da conta às métricas do painel da Lente de Armazenamento do S3 para funcionar apenas no 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 agregar 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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e 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 operações de API somente leitura do AWS Organizations. 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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e 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 o registro de uma conta como administrador delegado é cancelado, a conta perde a autorização para acessar todas as operações de API 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 esquerdo, escolha Storage Lens (Lente de Armazenamento) e 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 da Lente de Armazenamento do Amazon S3 usando a AWS CLI

A Lente de Armazenamento do S3 agrega suas métricas e exibe as informações na seção Account snapshot (Snapshot da conta) na página Buckets do console do Amazon S3. A Lente de Armazenamento do S3 também oferece um painel interativo 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. O painel tem opções de detalhamento para gerar e visualizar insights no nível da organização, conta, Região da AWS, classe de armazenamento, bucket, prefixo ou grupo de Lente de Armazenamento. Você também pode enviar uma exportação diária de métricas em CSV ou no formato Parquet para um bucket do S3. 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 a Lente de Armazenamento do S3 com a AWS Command Line Interface.

Tópicos

- [Arquivos auxiliares para usar o Amazon S3 Storage Lens](#)
- [Uso de configurações do Amazon S3 Storage Lens com a AWS CLI](#)
- [Usar a Lente de Armazenamento do Amazon S3 com exemplos do AWS Organizations usando a AWS CLI](#)

Arquivos auxiliares para usar o Amazon S3 Storage Lens

Use os seguintes arquivos JSON e suas principais entradas para seus exemplos.

Exemplo de configuração da Lente de Armazenamento do S3 em JSON

Example **config.json**

O arquivo `config.json` contém os detalhes de uma configuração de métricas e recomendações avançadas para organizações da Lente de Armazenamento do S3. Para usar o exemplo a seguir, substitua *user input placeholders* por suas próprias informações.

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 your S3
Storage Lens configuration.
  "AwsOrg": { //Use this property when enabling S3 Storage Lens for AWS Organizations.
    "Arn": "arn:aws:organizations::123456789012:organization/o-abcdefgh"
  },
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "DetailedStatusCodesMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
        "IsEnabled":true
      },
      "AdvancedDataProtectionMetrics": {
        "IsEnabled":true
      },
      "AdvancedCostOptimizationMetrics": {
        "IsEnabled":true
      },
      "DetailedStatusCodesMetrics": {
        "IsEnabled":true
      },
      "PrefixLevel":{
        "StorageMetrics":{
          "IsEnabled":true,
          "SelectionCriteria":{
```

```

        "MaxDepth":5,
        "MinStorageBytesPercentage":1.25,
        "Delimiter":"/"
    }
}
},
"Exclude": { //Replace with "Include" if you prefer to include Regions.
    "Regions": [
        "eu-west-1"
    ],
    "Buckets": [ //This attribute is not supported for AWS Organizations-level
configurations.
        "arn:aws:s3:::source_bucket1"
    ]
},
"IsEnabled": true, //Whether the configuration is enabled
"DataExport": { //Details about the metrics export
    "S3BucketDestination": {
        "OutputSchemaVersion": "V_1",
        "Format": "CSV", //You can add "Parquet" if you prefer.
        "AccountId": "111122223333",
        "Arn": "arn:aws:s3:::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": {
            "SSES3": {}
        }
    },
    "CloudWatchMetrics": {
        "IsEnabled": true
    }
}
}
}
}

```

Exemplo de configuração da Lente de Armazenamento do S3 com grupos de Lente de Armazenamento em JSON

Example `config.json`

O arquivo `config.json` contém os detalhes para aplicar à configuração de Lente de Armazenamento ao usar grupos de Lente de Armazenamento. Para usar o exemplo, substitua *user input placeholders* pelas suas próprias informações.

Para anexar todos os grupos de Lente de Armazenamento ao seu painel, atualize a configuração de Lente de Armazenamento com a seguinte sintaxe:

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled": true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled": true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled": true
    },
    "BucketLevel": {
      "ActivityMetrics": {
        "IsEnabled": true
      },
      "StorageLensGroupLevel": {},
      "IsEnabled": true
    }
  }
}
```

Para incluir apenas dois grupos de Lente de Armazenamento na configuração do painel da Lente de Armazenamento (*slg-1* and *slg-2*), use a seguinte sintaxe:

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled": true
    },
    "AdvancedCostOptimizationMetrics": {
```

```

    "IsEnabled": true
  },
  "AdvancedDataProtectionMetrics": {
    "IsEnabled": true
  },
  "BucketLevel": {
    "ActivityMetrics": {
      "IsEnabled": true
    },
  },
  "StorageLensGroupLevel": {
    "SelectionCriteria": {
      "Include": [
        "arn:aws:s3:us-east-1:111122223333:storage-lens-group/slg-1",
        "arn:aws:s3:us-east-1:444455556666:storage-lens-group/slg-2"
      ]
    },
  },
  "IsEnabled": true
}

```

Para impedir que apenas determinados grupos de Lente de Armazenamento sejam anexados à configuração do seu painel, use a seguinte sintaxe:

```

{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled": true
    },
  },
  "AdvancedCostOptimizationMetrics": {
    "IsEnabled": true
  },
  "AdvancedDataProtectionMetrics": {
    "IsEnabled": true
  },
  "BucketLevel": {
    "ActivityMetrics": {
      "IsEnabled": true
    },
  },
  "StorageLensGroupLevel": {
    "SelectionCriteria": {
      "Exclude": [
        "arn:aws:s3:us-east-1:111122223333:storage-lens-group/slg-1",
        "arn:aws:s3:us-east-1:444455556666:storage-lens-group/slg-2"
      ]
    },
  },
}

```

```
    ],
    "IsEnabled": true
}
```

Configuração de tags de exemplo da Lente de Armazenamento do S3 em JSON

Example **tags.json**

O arquivo `tags.json` contém as etiquetas que você deseja aplicar à configuração da Lente de Armazenamento do S3. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
[
  {
    "Key": "key1",
    "Value": "value1"
  },
  {
    "Key": "key2",
    "Value": "value2"
  }
]
```

Exemplo de configuração da Lente de Armazenamento do S3 Permissões IAM

Example **permissions.json**: nome de painel específico

Este exemplo de política mostra um arquivo `permissions.json` do IAM da Lente de Armazenamento do S3 com um nome de painel especificado. Substitua *value1*, *us-east-1*, *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`: sem um nome de painel específico

Este exemplo de política mostra um arquivo `permissions.json` do IAM da Lente de Armazenamento do S3 sem um nome de painel especificado. Substitua `value1`, `us-east-1` 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/*"
    }
  ]
}

```

Uso de configurações do Amazon S3 Storage Lens com a AWS CLI

Você pode usar a AWS CLI para listar, criar, excluir, obter, marcar e atualizar suas configurações da Lente de Armazenamento do S3. Os exemplos a seguir usam os arquivos JSON auxiliares para

entradas de chave. Para usar esses exemplos, substitua os *user input placeholders* por suas próprias informações.

Criar uma configuração da Lente de Armazenamento do S3

Example Criar uma configuração da Lente de Armazenamento do S3

```
aws s3control put-storage-lens-configuration --account-id=111122223333 --  
config-id=example-dashboard-configuration-id --region=us-east-1 --storage-lens-  
configuration=file:///./config.json --tags=file:///./tags.json
```

Criar uma configuração da Lente de Armazenamento do S3 sem etiquetas

Example Criar uma configuração da Lente de Armazenamento do S3 sem etiquetas

```
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 da Lente de Armazenamento do S3 sem um próximo token

Example Listar as configurações da Lente de Armazenamento do S3 sem um 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=abcdefghijkl234
```

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
```

Adicionar etiquetas em uma configuração da Lente de Armazenamento do S3

Example Adicionar etiquetas em uma configuração da Lente de Armazenamento do S3

```
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 a Lente de Armazenamento do Amazon S3 com exemplos do 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](#).

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

Para usar este exemplo, substitua **111122223333** pelo ID da Conta da AWS apropriada.

```
aws organizations register-delegated-administrator --service-principal storage-  
lens.s3.amazonaws.com --account-id 111122223333
```

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

Para usar este exemplo, substitua **111122223333** pelo ID da Conta da AWS apropriada.

```
aws organizations deregister-delegated-administrator --service-principal storage-  
lens.s3.amazonaws.com --account-id 111122223333
```

Exemplos do Amazon S3 Storage Lens usando o SDK for Java

A Lente de Armazenamento do S3 agrega suas métricas e exibe as informações na seção Account snapshot (Snapshot da conta) na página Buckets do console do Amazon S3. A Lente de Armazenamento do S3 também oferece um painel interativo 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. O painel tem opções de detalhamento para gerar e visualizar insights no nível da organização, conta, Região da AWS, classe de armazenamento, bucket, prefixo ou grupo de Lente de Armazenamento. Você também pode enviar uma exportação diária de métricas em CSV ou no formato Parquet para um bucket do S3. 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 a Lente de Armazenamento do S3 com a AWS SDK for Java.

Tópicos

- [Usando configurações do Amazon S3 Storage Lens usando o SDK for Java](#)

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](#)
- [Excluir uma configuração do S3 Storage Lens](#)
- [Obter uma configuração do S3 Storage Lens](#)
- [Listar configurações do S3 Storage Lens](#)
- [Adicionar etiquetas em uma configuração da Lente de Armazenamento do S3](#)
- [Obter tags para uma configuração do S3 Storage Lens](#)
- [Excluir tags para uma configuração do S3 Storage Lens](#)
- [Atualizar a configuração padrão da Lente de Armazenamento do S3 com métricas e recomendações avançadas](#)
- [Anexar um grupo de Lente de Armazenamento a um painel da Lente de Armazenamento do S3](#)
- [Usar a Lente de Armazenamento do Amazon S3 com exemplos do AWS Organizations usando SDK for Java](#)

Criar e atualizar uma configuração do S3 Storage Lens

Exemplo 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.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 Storage Lens
        configuration.
        String awsOrgARN = "arn:aws:organizations::123456789012:organization/o-
        abcdefgh";
        Format exportFormat = Format.CSV;

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                .withDelimiter("/")
                .withMaxDepth(5)
                .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
            PrefixLevelStorageMetrics()
                .withIsEnabled(true)
                .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
```

```

        .withAdvancedCostOptimizationMetrics(new
AdvancedCostOptimizationMetrics().withIsEnabled(true))
        .withAdvancedDataProtectionMetrics(new
AdvancedDataProtectionMetrics().withIsEnabled(true))
        .withDetailedStatusCodesMetrics(new
DetailedStatusCodesMetrics().withIsEnabled(true))
        .withPrefixLevel(new
PrefixLevel().withStorageMetrics(prefixStorageMetrics));
    AccountLevel accountLevel = new AccountLevel()
        .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
        .withAdvancedCostOptimizationMetrics(new
AdvancedCostOptimizationMetrics().withIsEnabled(true))
        .withAdvancedDataProtectionMetrics(new
AdvancedDataProtectionMetrics().withIsEnabled(true))
        .withDetailedStatusCodesMetrics(new
DetailedStatusCodesMetrics().withIsEnabled(true))
        .withBucketLevel(bucketLevel);

    Include include = new Include()
        .withBuckets(Arrays.asList("arn:aws:s3:::bucketName"))
        .withRegions(Arrays.asList("us-west-2"));

    StorageLensDataExportEncryption exportEncryption = new
StorageLensDataExportEncryption()
        .withSSES3(new SSES3());
    S3BucketDestination s3BucketDestination = new S3BucketDestination()
        .withAccountId(exportAccountId)
        .withArn(exportBucketArn)
        .withEncryption(exportEncryption)
        .withFormat(exportFormat)
        .withOutputSchemaVersion(OutputSchemaVersion.V_1)
        .withPrefix("Prefix");
    CloudWatchMetrics cloudWatchMetrics = new CloudWatchMetrics()
        .withIsEnabled(true);
    StorageLensDataExport dataExport = new StorageLensDataExport()
        .withCloudWatchMetrics(cloudWatchMetrics)
        .withS3BucketDestination(s3BucketDestination);

    StorageLensAwsOrg awsOrg = new StorageLensAwsOrg()
        .withArn(awsOrgARN);

    StorageLensConfiguration configuration = new StorageLensConfiguration()
        .withId(configurationId)
        .withAccountLevel(accountLevel)

```

```

        .withInclude(include)
        .withDataExport(dataExport)
        .withAwsOrg(awsOrg)
        .withIsEnabled(true);

    List<StorageLensTag> tags = Arrays.asList(
        new StorageLensTag().withKey("key-1").withValue("value-1"),
        new StorageLensTag().withKey("key-2").withValue("value-2")
    );

    AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
        .withCredentials(new ProfileCredentialsProvider())
        .withRegion(US_WEST_2)
        .build();

    s3ControlClient.putStorageLensConfiguration(new
PutStorageLensConfigurationRequest()
        .withAccountId(sourceAccountId)
        .withConfigId(configurationId)
        .withStorageLensConfiguration(configuration)
        .withTags(tags)
    );
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
}
}

```

Excluir uma configuração do S3 Storage Lens

Example Excluir 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();
        }
    }
}
```

Obter 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();
        }
    }
}
```

Listar configurações do S3 Storage Lens

Example Listar 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();
        }
    }
}
```

```
}
```

Adicionar etiquetas em uma configuração da Lente de Armazenamento do S3

Exemplo Adicionar etiquetas em uma configuração da Lente de Armazenamento do S3

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import
    com.amazonaws.services.s3control.model.PutStorageLensConfigurationTaggingRequest;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class PutDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "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")
            );

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            s3ControlClient.putStorageLensConfigurationTagging(new
                PutStorageLensConfigurationTaggingRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withTags(tags))
        }
    }
}
```

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

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 da Lente de Armazenamento do S3 com métricas e recomendações avançadas

Example Atualizar a configuração padrão da Lente de Armazenamento do S3 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.SSES3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class UpdateDefaultConfigWithPaidFeatures {

    public static void main(String[] args) {
        String configurationId = "default-account-dashboard"; // This configuration ID
        cannot be modified.
        String sourceAccountId = "Source Account ID";

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                .withDelimiter("/")
                .withMaxDepth(5)
                .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
            PrefixLevelStorageMetrics()
                .withIsEnabled(true)
                .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                .withPrefixLevel(new
            PrefixLevel().withStorageMetrics(prefixStorageMetrics));
            AccountLevel accountLevel = new AccountLevel()
                .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                .withBucketLevel(bucketLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                .withId(configurationId)
                .withAccountLevel(accountLevel)
                .withIsEnabled(true);

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
```



```
        .withRegion(US_WEST_2)
        .build();

        s3ControlClient.putStorageLensConfiguration(new
PutStorageLensConfigurationRequest()
            .withAccountId(sourceAccountId)
            .withConfigId(configurationId)
            .withStorageLensConfiguration(configuration)
        );

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

Note

Aplicam-se cobranças adicionais para métricas e recomendações avançadas. Para obter mais informações, consulte [Métricas e recomendações avançadas](#).

Anexar um grupo de Lente de Armazenamento a um painel da Lente de Armazenamento do S3

Example Anexar todos os grupos do Storage Lens a um painel

O exemplo de SDK para Java a seguir anexa todos os grupos de Lente de Armazenamento na conta **111122223333** ao painel *DashBoardConfigurationId*:

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
```

```
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensGroupLevel;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateDashboardWithStorageLensGroups {
    public static void main(String[] args) {
        String configurationId = "ExampleDashboardConfigurationId";
        String sourceAccountId = "111122223333";

        try {
            StorageLensGroupLevel storageLensGroupLevel = new StorageLensGroupLevel();

            AccountLevel accountLevel = new AccountLevel()
                .withBucketLevel(new BucketLevel())
                .withStorageLensGroupLevel(storageLensGroupLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                .withId(configurationId)
                .withAccountLevel(accountLevel)
                .withIsEnabled(true);

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            s3ControlClient.putStorageLensConfiguration(new
            PutStorageLensConfigurationRequest()
                .withAccountId(sourceAccountId)
                .withConfigId(configurationId)
                .withStorageLensConfiguration(configuration)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
        }
    }
}
```

```

        e.printStackTrace();
    }
}
}

```

Exemplo Anexar dois grupos de Lente de Armazenamento a um painel

O exemplo AWS SDK for Java a seguir anexa dois grupos de Lente de Armazenamento (*StorageLensGroupName1* e *StorageLensGroupName2*) ao painel *ExampleDashboardConfigurationId*.

```

package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensGroupLevel;
import com.amazonaws.services.s3control.model.StorageLensGroupLevelSelectionCriteria;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateDashboardWith2StorageLensGroups {
    public static void main(String[] args) {
        String configurationId = "ExampleDashboardConfigurationId";
        String storageLensGroupName1 = "StorageLensGroupName1";
        String storageLensGroupName2 = "StorageLensGroupName2";
        String sourceAccountId = "111122223333";

        try {
            StorageLensGroupLevelSelectionCriteria selectionCriteria = new
StorageLensGroupLevelSelectionCriteria()
                .withInclude(
                    "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
+ ":storage-lens-group/" + storageLensGroupName1,
                    "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
+ ":storage-lens-group/" + storageLensGroupName2);

```

```

System.out.println(selectionCriteria);
StorageLensGroupLevel storageLensGroupLevel = new StorageLensGroupLevel()
    .withSelectionCriteria(selectionCriteria);

AccountLevel accountLevel = new AccountLevel()
    .withBucketLevel(new BucketLevel())
    .withStorageLensGroupLevel(storageLensGroupLevel);

StorageLensConfiguration configuration = new StorageLensConfiguration()
    .withId(configurationId)
    .withAccountLevel(accountLevel)
    .withIsEnabled(true);

AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
    .withCredentials(new ProfileCredentialsProvider())
    .withRegion(US_WEST_2)
    .build();

s3ControlClient.putStorageLensConfiguration(new
PutStorageLensConfigurationRequest()
    .withAccountId(sourceAccountId)
    .withConfigId(configurationId)
    .withStorageLensConfiguration(configuration)
);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
}
}

```

Example Anexar todos os grupos de Lente de Armazenamento com exclusões

O exemplo de SDK para Java a seguir anexa todos os grupos de Lente de Armazenamento ao painel *ExampleDashboardConfigurationId*, exceto os dois especificados (*StorageLensGroupName1* e *StorageLensGroupName2*):

```
package aws.example.s3control;
```

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensGroupLevel;
import com.amazonaws.services.s3control.model.StorageLensGroupLevelSelectionCriteria;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateDashboardWith2StorageLensGroupsExcluded {
    public static void main(String[] args) {
        String configurationId = "ExampleDashboardConfigurationId";
        String storageLensGroupName1 = "StorageLensGroupName1";
        String storageLensGroupName2 = "StorageLensGroupName2";
        String sourceAccountId = "111122223333";

        try {
            StorageLensGroupLevelSelectionCriteria selectionCriteria = new
StorageLensGroupLevelSelectionCriteria()
                .withInclude(
                    "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
+ ":storage-lens-group/" + storageLensGroupName1,
                    "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
+ ":storage-lens-group/" + storageLensGroupName2);

            System.out.println(selectionCriteria);
            StorageLensGroupLevel storageLensGroupLevel = new StorageLensGroupLevel()
                .withSelectionCriteria(selectionCriteria);

            AccountLevel accountLevel = new AccountLevel()
                .withBucketLevel(new BucketLevel())
                .withStorageLensGroupLevel(storageLensGroupLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                .withId(configurationId)
                .withAccountLevel(accountLevel)
                .withIsEnabled(true);
```

```
AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
    .withCredentials(new ProfileCredentialsProvider())
    .withRegion(US_WEST_2)
    .build();

s3ControlClient.putStorageLensConfiguration(new
PutStorageLensConfigurationRequest()
    .withAccountId(sourceAccountId)
    .withConfigId(configurationId)
    .withStorageLensConfiguration(configuration)
);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Usar a Lente de Armazenamento do Amazon S3 com exemplos do AWS Organizations usando 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](#)
- [Desativar o acesso confiável das organizações para o S3 Storage Lens](#)
- [Registrar administradores delegados das Organizações para o S3 Storage Lens](#)
- [Cancelar o registro de administradores delegados do Organizations para o S3 Storage Lens](#)

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 = "111122223333"; // Account Id for the
delegated administrator.
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.registerDelegatedAdministrator(new
RegisterDelegatedAdministratorRequest()
                .withAccountId(delegatedAdminAccountId)
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
        }
    }
}
```

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 = "111122223333"; // Account Id for the
delegated administrator.
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.deregisterDelegatedAdministrator(new
DeregisterDelegatedAdministratorRequest()
                .withAccountId(delegatedAdminAccountId)
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
        }
    }
}
```

Trabalho com grupos de Lente de Armazenamento do S3

Um grupo de Lente de Armazenamento do Amazon S3 agrega métricas usando filtros personalizados com base nos metadados do objeto. Os grupos de Lente de Armazenamento ajudam a detalhar as características dos dados, como distribuição de objetos por idade, tipos de arquivo mais comuns e muito mais. Por exemplo, você pode filtrar métricas por tag de objeto para identificar conjuntos de dados de crescimento mais rápido ou visualizar armazenamento com base no tamanho e na idade do objeto para informar a estratégia de arquivamento de armazenamento. Como resultado, os grupos de Lente de Armazenamento do Amazon S3 ajudam a entender e otimizar o armazenamento do S3.

Ao usar grupos de Lente de Armazenamento, você pode analisar e filtrar as métricas de Lente de Armazenamento do S3 usando metadados de objetos, como prefixos, sufixos, [tags de objetos](#), tamanho do objeto ou idade do objeto. Também é possível aplicar uma combinação desses filtros. Depois de anexar o grupo de Lente de Armazenamento ao painel de Lente de Armazenamento do S3, você pode visualizar as métricas de Lente de Armazenamento do S3 agregadas pelos grupos de Lente de Armazenamento do Amazon S3 diretamente no seu painel.

Você também pode filtrar métricas por tamanho de objeto ou faixa etária para determinar qual parte do armazenamento consiste em objetos pequenos. Você pode usar essas informações com o S3 Intelligent-Tiering ou o S3 Lifecycle para fazer a transição de objetos pequenos para diferentes classes de armazenamento a fim de otimizar custos e armazenamento.

Tópicos

- [Como funcionam os grupos de Lente de Armazenamento do S3](#)
- [Uso de grupos de Lente de Armazenamento](#)

Como funcionam os grupos de Lente de Armazenamento do S3

Você pode usar grupos da Lente de Armazenamento para agregar métricas usando filtros personalizados com base em metadados de objetos. Ao definir um filtro personalizado, você pode usar prefixos, sufixos, tags de objeto, tamanhos de objeto, idade do objeto ou uma combinação desses filtros personalizados. Durante a criação do grupo da Lente de Armazenamento, você também pode incluir um único filtro ou várias condições de filtro. Para especificar várias condições de filtro, você usa os operadores lógicos And ou Or.

Ao criar e configurar um grupo da Lente de Armazenamento, o próprio grupo da Lente de Armazenamento atua como um filtro personalizado no painel ao qual você anexa o grupo. No painel,

você pode usar o filtro de grupo da Lente de Armazenamento para obter métricas de armazenamento com base no filtro personalizado definido no grupo.

Para visualizar os dados do grupo da Lente de Armazenamento no painel da Lente de Armazenamento do S3, você deve anexar o grupo ao painel depois de criá-lo. Depois que o grupo da Lente de Armazenamento for anexado ao painel da Lente de Armazenamento, o painel coletará métricas de uso de armazenamento em 48 horas. Visualize dados no painel da Lente de Armazenamento ou exporte por meio de uma exportação de métricas. Caso você se esqueça de anexar um grupo da Lente de Armazenamento a um painel, os dados do grupo da Lente de Armazenamento não serão capturados nem exibidos em nenhum lugar.

Note

- Ao criar um grupo da Lente de Armazenamento do S3, você está criando um recurso AWS. Portanto, cada grupo da Lente de Armazenamento tem o próprio nome do recurso da Amazon (ARN), que você pode especificar ao [anexá-lo ou excluí-lo de um painel da Lente de Armazenamento do S3](#).
- Se o seu grupo da Lente de Armazenamento não estiver anexado a um painel, você não incorrerá em nenhuma cobrança adicional pela criação de um grupo da Lente de Armazenamento.
- A Lente de Armazenamento do S3 agrega métricas de uso de um objeto em todos os grupos da Lente de Armazenamento correspondentes. Portanto, se um objeto corresponder às condições de filtro de dois ou mais grupos da Lente de Armazenamento, você verá contagens repetidas do mesmo objeto em todo o uso do armazenamento.

Você pode criar um grupo da Lente de Armazenamento no nível da conta em uma região inicial especificada (na lista de suportes Regiões da AWS). Em seguida, você pode anexar seu grupo da Lente de Armazenamento a vários painéis da Lente de Armazenamento, desde que os painéis estejam no mesmo Conta da AWS e região inicial. Você pode criar até 50 grupos da Lente de Armazenamento por região inicial em cada Conta da AWS.

Você pode criar e gerenciar grupos de Lente de Armazenamento do S3 usando o console do Amazon S3, AWS Command Line Interface (AWS CLI), AWS SDKs ou a API REST do Amazon S3.

Tópicos

- [Ver métricas agregadas do grupo da Lente de Armazenamento](#)

- [Permissões de grupos de Lente de Armazenamento](#)
- [Configuração de grupos da Lente de Armazenamento](#)
- [Tags de recursos AWS](#)
- [Exportação de métricas de grupos da Lente de Armazenamento](#)

Ver métricas agregadas do grupo da Lente de Armazenamento

Você pode ver as métricas agregadas dos seus grupos da Lente de Armazenamento ao anexar os grupos a um painel. Os grupos da Lente de Armazenamento que deseja anexar devem estar na região inicial designada na conta do painel.

Para anexar um grupo da Lente de Armazenamento a um painel, você deve especificar o grupo na Seção de agregação do grupo da Lente de Armazenamento da configuração do seu painel. Se você tiver vários grupos da Lente de Armazenamento, poderá filtrar os resultados da agregação do grupo da Lente de Armazenamento para incluir ou excluir apenas os grupos desejados. Para obter mais informações sobre como anexar grupos aos seus painéis, consulte [the section called “Anexe ou remova um grupo de Lente de Armazenamento”](#).

Depois de anexar seus grupos, você verá os dados adicionais de agregação de grupos da Lente de Armazenamento em seu painel dentro de 48 horas.

Note

Para ver métricas agregadas para seu grupo da Lente de Armazenamento, você deve anexar o grupo a um painel da Lente de Armazenamento do S3.

Permissões de grupos de Lente de Armazenamento

Os grupos de Lente de Armazenamento exigem determinadas permissões no AWS Identity and Access Management (IAM) para autorizar o acesso às ações do grupo de Lente de Armazenamento do S3. Para conceder essas permissões, você pode usar uma política do IAM baseada em identidade. Você pode vincular essa política a usuários, grupos ou funções do IAM para conceder permissões. Essas permissões podem incluir a capacidade de criar ou excluir grupos da Lente de Armazenamento, visualizar suas configurações ou gerenciar suas tags.

O usuário ou função do IAM ao qual você concede permissões deve pertencer à conta que criou ou é proprietária do grupo da Lente de Armazenamento.

Para usar grupos da Lente de Armazenamento e ver as métricas dos grupos da Lente de Armazenamento, primeiro você deve ter as permissões apropriadas para usar a Lente de Armazenamento do S3. Para ter mais informações, consulte [the section called “Permissões do S3 Storage Lens”](#).

Para criar e gerenciar grupos da Lente de Armazenamento do S3, você deve ter as seguintes permissões do IAM, dependendo das ações que deseja executar:

Ação	Permissões do IAM
Criar grupo da Lente de Armazenamento	<code>s3:CreateStorageLensGroup</code>
Criar grupo da Lente de Armazenamento com tags	<code>s3:CreateStorageLensGroup</code> , <code>s3:TagResource</code>
Atualizar um grupo da Lente de Armazenamento existente	<code>s3:UpdateStorageLensGroup</code>
Gerar os detalhes de uma configuração de grupo da Lente de Armazenamento	<code>s3:GetStorageLensGroup</code>
Listar todos os grupos da Lente de Armazenamento na região de origem	<code>s3:ListStorageLensGroups</code>
Excluir um grupo da Lente de Armazenamento	<code>s3>DeleteStorageLensGroup</code>
Listar as tags que foram adicionadas ao seu grupo da Lente de Armazenamento	<code>s3:ListTagsForResource</code>
Adicionar ou atualizar uma tag de grupo da Lente de Armazenamento para um grupo da Lente de Armazenamento existente	<code>s3:TagResource</code>
Excluir uma tag de um grupo da Lente de Armazenamento	<code>s3:UntagResource</code>

Veja um exemplo de como configurar a política do IAM na conta que cria o grupo da Lente de Armazenamento. Para usar essa política, substitua `us-east-1` pela região inicial em que seu grupo e Lente de Armazenamento está localizado. Substitua `111122223333` pelo seu Conta da AWS ID e

substitua *example-storage-lens-group* pelo nome do seu grupo da Lente de Armazenamento. Para aplicar essas permissões a todos os grupos da Lente de Armazenamento, substitua *example-storage-lens-group* por ***.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EXAMPLE-Statement-ID",
      "Effect": "Allow",
      "Action": [
        "s3:CreateStorageLensGroup",
        "s3:UpdateStorageLensGroup",
        "s3:GetStorageLensGroup",
        "s3:ListStorageLensGroups",
        "s3>DeleteStorageLensGroup",
        "s3:TagResource",
        "s3:UntagResource",
        "s3:ListTagsForResource"
      ],
      "Resource": "arn:aws:s3:us-east-1:111122223333:storage-lens-group/example-storage-lens-group"
    }
  ]
}
```

Para obter mais informações sobre as permissões de Lente de Armazenamento do S3, consulte [Permissões do Amazon S3 Storage Lens](#). Para obter mais informações sobre a linguagem da política do IAM, consulte [Políticas e permissões no Amazon S3](#).

Configuração de grupos da Lente de Armazenamento

Nome do grupo da Lente de Armazenamento do S3

Recomendamos dar nomes aos grupos da Lente de Armazenamento que indiquem a finalidade para determinar quais grupos anexar aos seus painéis. Para [anexar um grupo da Lente de Armazenamento a um painel](#), você deve especificar o grupo na Seção de agregação do grupo da Lente de Armazenamento da configuração do painel.

Os nomes dos grupos de Lente de Armazenamento devem ser exclusivos na conta. Eles não devem exceder 64 caracteres e podem conter apenas letras (a–z, A–Z), números (0–9), hifens (-) e sublinhados (_).

Região inicial

A região inicial é a Região da AWS em que o grupo da Lente de Armazenamento é criado e mantido. O grupo da Lente de Armazenamento é criado na mesma região inicial do painel da Lente de Armazenamento do Amazon S3. A configuração e as métricas do grupo da Lente de Armazenamento também são armazenadas nessa região. Você pode criar até 50 grupos da Lente de Armazenamento em uma região local.

Depois de criar seu grupo da Lente de Armazenamento, você não poderá editar a região inicial.

Escopo

Para incluir objetos no grupo da Lente de Armazenamento, eles devem estar no escopo do painel da Lente de Armazenamento do Amazon S3. O escopo do painel da Lente de Armazenamento é determinado pelos buckets incluídos no Escopo do painel da configuração do painel da Lente de Armazenamento do S3.

Você pode usar filtros diferentes para seus objetos a fim de definir o escopo do seu grupo da Lente de Armazenamento. Para ver essas métricas de grupo da Lente de Armazenamento no painel da Lente de Armazenamento do S3, os objetos devem corresponder aos filtros incluídos nos grupos da Lente de Armazenamento. Por exemplo, suponha que seu grupo da Lente de Armazenamento inclua objetos com o prefixo `marketing` e o sufixo `.png`, mas nenhum objeto corresponda a esses critérios. Nesse caso, as métricas desse grupo da Lente de Armazenamento não serão geradas na exportação diária de métricas e nenhuma métrica desse grupo ficará visível no seu painel.

Filtros

Você pode usar os seguintes filtros em um grupo da Lente de Armazenamento do S3:

- **Prefixos:** especifica o [prefixo](#) dos objetos incluídos, que é uma sequência de caracteres no início do nome da chave do objeto. Por exemplo, um valor de `images` para o filtro Prefixos inclui objetos com qualquer um dos seguintes prefixos: `images/`, `images-marketing` e `images/production`. O comprimento máximo de um prefixo é 1.024 bytes.
- **Sufixos:** especifica o sufixo dos objetos incluídos (por exemplo, `.png`, `.jpeg` ou `.csv`). O comprimento máximo de um sufixo é 1.024 bytes.
- **Tags de objeto:** especifica a lista de [tags de objeto](#) que você deseja filtrar. Uma chave de tag não pode exceder 128 caracteres Unicode, e um valor de tag não pode exceder 256 caracteres Unicode. Se o campo de valor da etiqueta do objeto ficar vazio, os grupos da Lente de

Armazenamento do S3 só associarão o objeto a outros objetos que também tenham valores de etiqueta vazios.

- **Idade:** especifica o intervalo de idade do objeto incluído em dias. Somente números inteiros são compatíveis.
- **Tamanho:** especifica o intervalo de tamanho dos objetos incluídos em bytes. Somente números inteiros são compatíveis. O valor máximo permitido é 5 TB.

Tags de objeto do grupo da Lente de Armazenamento

Você pode [criar um grupo da Lente de Armazenamento](#) que inclua até 10 filtros de tags de objetos. O exemplo a seguir inclui dois pares de valores-chave de tags de objeto como filtros para um grupo da Lente de Armazenamento denominado *Marketing-Department*. Para usar esse exemplo, substitua *Marketing-Department* pelo nome do seu grupo e substitua *object-tag-key-1*, *object-tag-value-1* e assim por diante pelos pares de valores-chave da tag de objeto que você deseja filtrar.

```
{
  "Name": "Marketing-Department",
  "Filter": {
    "MatchAnyTag": [
      {
        "Key": "object-tag-key-1",
        "Value": "object-tag-value-1"
      },
      {
        "Key": "object-tag-key-2",
        "Value": "object-tag-value-2"
      }
    ]
  }
}
```

Operadores lógicos (**And** ou **Or**)

Para incluir diversas condições de filtro em seu grupo da Lente de Armazenamento, use operadores lógicos (tanto And quanto Or). No exemplo a seguir, o grupo e Lente de Armazenamento *Marketing-Department* contém um operador And com os filtros Prefix, ObjectAge e ObjectSize. Como um operador And é usado, somente objetos que correspondam a todas as condições de filtro serão incluídos no escopo do grupo da Lente de Armazenamento.

Para usar esse exemplo, substitua *user input placeholders* pelos valores que você deseja filtrar.

```
{
  "Name": "Marketing-Department",
  "Filter": {
    "And": {
      "MatchAnyPrefix": [
        "prefix-1",
        "prefix-2",
        "prefix-3/sub-prefix-1"
      ],
      "MatchObjectAge": {
        "DaysGreaterThan": 10,
        "DaysLessThan": 60
      },
      "MatchObjectSize": {
        "BytesGreaterThan": 10,
        "BytesLessThan": 60
      }
    }
  }
}
```

Note

Se desejar incluir objetos que correspondam a qualquer uma das condições nos filtros, substitua o operador lógico And pelo operador lógico Or deste exemplo.

Tags de recursos AWS

Cada grupo da Lente de Armazenamento do S3 é contado como um recurso AWS com seu próprio nome do recurso da Amazon (ARN). Portanto, ao configurar o grupo da Lente de Armazenamento, você tem a opção de adicionar tags de recursos AWS ao grupo. Você pode adicionar até 50 tags para cada grupo da Lente de Armazenamento. Para criar um grupo da Lente de Armazenamento com tags, você deve ter as permissões `s3:CreateStorageLensGroup` e `s3:TagResource`.

Use tags de recursos AWS para categorizar os recursos de acordo com o departamento, a linha de negócios ou o projeto. Fazer isso é útil quando você tem muitos recursos do mesmo tipo. Ao aplicar

tags, é possível identificar rapidamente um grupo específico da Lente de Armazenamento com base nas tags atribuídas a ele. Você também pode usar tags para rastrear e alocar custos.

Além disso, ao adicionar uma tag de recursos AWS ao grupo da Lente de Armazenamento, você ativa o [controle de acesso por atributo \(ABAC\)](#). ABAC é uma estratégia de autorização que define permissões com base em atributos, neste caso, tags. É possível usar condições que especificam tags de recursos nas políticas do IAM para [controlar o acesso aos recursos AWS](#).

É possível editar chaves de tags e valores, e é possível remover as tags de um recurso a qualquer momento. Além disso, esteja ciente das seguintes limitações:

- As chaves e os valores das tags diferenciam maiúsculas de minúsculas.
- Ao adicionar uma tag que tenha a mesma chave de uma tag existente nesse recurso, o novo valor substituirá o antigo.
- Se você excluir um recurso, todas as tags do recurso também serão excluídas.
- Não inclua dados privados ou confidenciais nas tags de recursos AWS.
- As tags do sistema (ou tags com chaves de tag que começam com `aws:`) não são compatíveis.
- O comprimento de cada chave de tag não pode exceder 128 caracteres. O comprimento de cada valor de tag não pode exceder 256 caracteres.

Exportação de métricas de grupos da Lente de Armazenamento

As métricas do grupo da Lente de Armazenamento do S3 estão incluídas na [exportação de métricas de lente de armazenamento do Amazon S3](#) para o painel ao qual o grupo da Lente de Armazenamento está anexado. Para obter informações gerais sobre o recurso de exportação de métricas da Lente de Armazenamento, consulte [Exibição de métricas de lente de armazenamento do Amazon S3 usando uma exportação de dados](#).

A exportação de métricas para grupos da Lente de Armazenamento inclui quaisquer métricas da Lente de Armazenamento do S3 que estejam no escopo do painel ao qual você anexou o grupo da Lente de Armazenamento. A exportação também inclui dados de métricas adicionais para grupos da Lente de Armazenamento.

Depois de criar seu grupo da Lente de Armazenamento, sua exportação de métricas será enviada diariamente para o bucket selecionado ao configurar a exportação de métricas para o painel ao qual seu grupo está anexado. Pode levar até 48 horas para você receber a primeira exportação de métricas.

Para gerar métricas na exportação diária, os objetos devem corresponder aos filtros incluídos nos grupos da Lente de Armazenamento. Se nenhum objeto corresponder aos filtros incluídos no grupo da Lente de Armazenamento, nenhuma métrica será gerada. No entanto, se um objeto corresponder a dois ou mais grupos da Lente de Armazenamento, o objeto será listado separadamente para cada grupo quando aparecer na exportação de métricas.

Você pode identificar métricas para grupos da Lente de Armazenamento procurando um dos seguintes valores na coluna `record_type` de exportação de métricas do seu painel:

- `STORAGE_LENS_GROUP_BUCKET`
- `STORAGE_LENS_GROUP_ACCOUNT`

A coluna `record_value` mostra o ARN do recurso para o grupo da Lente de Armazenamento (por exemplo, `arn:aws:s3:us-east-1:111122223333:storage-lens-group/Marketing-Department`).

Uso de grupos de Lente de Armazenamento

A Lente de Armazenamento do Amazon S3 agrega métricas usando filtros personalizados com base nos metadados do objeto. Você pode analisar e filtrar as métricas de Lente de Armazenamento do S3 usando prefixos, sufixos, tags de objetos, tamanho do objeto ou idade do objeto. Com grupos de Lente de Armazenamento do Amazon S3, você também pode categorizar seu uso dentro de buckets e por meio deles no Amazon S3. Consequentemente, você poderá entender e otimizar melhor o armazenamento S3.

Para começar a visualizar os dados de um grupo de Lente de Armazenamento, você deve primeiro [anexar o grupo de Lente de Armazenamento a um painel de Lente de Armazenamento do S3](#). Se precisar gerenciar grupos de Lente de Armazenamento no painel, edite a configuração do painel. Para verificar quais grupos de Lente de Armazenamento estão na sua conta, você pode listá-los. Para verificar quais grupos de Lente de Armazenamento estão anexados ao seu painel, verifique a guia Grupos de Lente de Armazenamento no painel. Para revisar ou atualizar o escopo de um grupo de Lente de Armazenamento existente, visualize os detalhes. Você também pode excluir permanentemente um grupo de Lente de Armazenamento.

Para gerenciar permissões, você pode criar e adicionar tags de recursos AWS definidas pelo usuário aos seus grupos de Lente de Armazenamento. Use tags de recursos AWS para categorizar os recursos de acordo com o departamento, a linha de negócios ou o projeto. Fazer isso é útil quando

you have many resources of the same type. When you apply tags, it's possible to quickly identify a specific group of Storage Lens with tags assigned to it.

In addition, when you add an AWS resource tag to a Storage Lens group, you activate [attribute-based access control \(ABAC\)](#). ABAC is a strategy for authorization that defines permissions based on attributes, in this case, tags. It's possible to use conditions that specify tags of resources in IAM policies for [controlling access to AWS resources](#).

Tópicos

- [Criação de grupo da Lente de Armazenamento](#)
- [Anexar ao painel ou remover do painel grupos de Lente de Armazenamento do S3](#)
- [Visualização de dados dos grupos de Lente de Armazenamento](#)
- [Atualização do grupo de Lente de Armazenamento](#)
- [Gerenciamento de tags de recursos AWS com grupos de Lente de Armazenamento](#)
- [Listagem de todos os grupos de Lente de Armazenamento](#)
- [Visualização de detalhes do grupo de Lente de Armazenamento](#)
- [Exclusão de grupo de Lente de Armazenamento](#)

Criação de grupo da Lente de Armazenamento

The following examples demonstrate how to create a Storage Lens group on Amazon S3 using the Amazon S3 AWS Command Line Interface (AWS CLI) and AWS SDK for Java.

Uso do console do S3

Para criar um grupo da Lente 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 barra de navegação na parte superior da página, escolha o nome da região da AWS exibida no momento. Em seguida, escolha a região para a qual você deseja mudar.
3. No painel de navegação à esquerda, selecione Grupos de Lente de Armazenamento.
4. Selecione Criar grupo da Lente de Armazenamento.
5. Em Geral, visualize sua Região de origem e insira o Nome do grupo do Storage Lens.
6. Em Escopo, selecione o filtro para aplicar ao seu grupo da Lente de Armazenamento. Para aplicar vários filtros, selecione os filtros e, em seguida, escolha o operador lógico AND ou OR.

- Para o filtro Prefixos, selecione Prefixos e insira uma sequência de prefixo. Para adicionar vários prefixos, selecione Adicionar prefixo. Para remover um prefixo, selecione Remover ao lado do prefixo para remover.
 - Para o filtro Tags de objeto, selecione Tags de objeto e insira o par de valores-chave para seu objeto. Selecione Adicionar tag. Para remover uma tag, selecione Remover ao lado da tag para remover.
 - Para o filtro Sufixos, selecione Sufixos e insira uma sequência de sufixo. Para adicionar vários sufixos, selecione Adicionar sufixo. Para remover um sufixo, selecione Remover ao lado do sufixo para remover.
 - Para o filtro Idade, especifique a faixa etária do objeto em dias. Selecione Especificar idade mínima do objeto e insira a idade mínima do objeto. Em seguida, selecione Especificar idade máxima do objeto e insira a idade máxima do objeto.
 - Para o filtro Tamanho, especifique a faixa de tamanho do objeto e a unidade de medida. Selecione Especificar tamanho mínimo do objeto e insira o tamanho mínimo do objeto. Selecione Especificar tamanho máximo do objeto e insira o tamanho máximo do objeto.
7. (Opcional) Para tags de recursos AWS, adicione o par de valores-chave e selecione Adicionar tag.
 8. Selecione Criar grupo da Lente de Armazenamento.

Usando a AWS CLI

O comando de exemplo AWS CLI a seguir cria um grupo de Lente de Armazenamento. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control create-storage-lens-group --account-id 111122223333 \  
--region us-east-1 --storage-lens-group=file:///./marketing-department.json
```

O comando de exemplo AWS CLI a seguir cria um grupo da Lente de Armazenamento com duas tags de recursos AWS. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control create-storage-lens-group --account-id 111122223333 \  
--region us-east-1 --storage-lens-group=file:///./marketing-department.json \  
--tags Key=k1,Value=v1 Key=k2,Value=v2
```

Para obter configurações de exemplo JSON, consulte [Configuração de grupos da Lente de Armazenamento](#).

Usar o AWS SDK para Java

O exemplo AWS SDK for Java a seguir cria um grupo da Lente de Armazenamento. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

Example — Crie um grupo da Lente de Armazenamento com um único filtro

O exemplo a seguir cria um grupo de Lente de Armazenamento chamado *Marketing-Department*. Esse grupo tem um filtro de idade do objeto que especifica a faixa etária em *30* para *90* dias. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.MatchObjectAge;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;

public class CreateStorageLensGroupWithObjectAge {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            StorageLensGroupFilter objectAgeFilter = StorageLensGroupFilter.builder()
                .matchObjectAge(MatchObjectAge.builder()
                    .daysGreaterThan(30)
                    .daysLessThan(90)
                    .build())
                .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                .name(storageLensGroupName)
                .filter(objectAgeFilter)
```

```

        .build();

        CreateStorageLensGroupRequest createStorageLensGroupRequest =
CreateStorageLensGroupRequest.builder()
        .storageLensGroup(storageLensGroup)
        .accountId(accountId).build();

        S3ControlClient s3ControlClient = S3ControlClient.builder()
        .region(Region.US_WEST_2)
        .credentialsProvider(ProfileCredentialsProvider.create())
        .build();
        s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it and returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
}

```

Example — Crie um grupo da Lente de Armazenamento com um operador **AND** que inclua vários filtros

O exemplo a seguir cria um grupo de Lente de Armazenamento chamado *Marketing-Department*. Este grupo utiliza o operador AND para indicar que os objetos devem corresponder a todas as condições de filtro. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```

package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.MatchObjectAge;
import software.amazon.awssdk.services.s3control.model.MatchObjectSize;
import software.amazon.awssdk.services.s3control.model.S3Tag;

```



```
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupAndOperator;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;

public class CreateStorageLensGroupWithAndFilter {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            // Create object tags.
            S3Tag tag1 = S3Tag.builder()
                .key("object-tag-key-1")
                .value("object-tag-value-1")
                .build();
            S3Tag tag2 = S3Tag.builder()
                .key("object-tag-key-2")
                .value("object-tag-value-2")
                .build();

            StorageLensGroupAndOperator andOperator =
                StorageLensGroupAndOperator.builder()
                    .matchAnyPrefix("prefix-1", "prefix-2", "prefix-3/sub-prefix-1")
                    .matchAnySuffix(".png", ".gif", ".jpg")
                    .matchAnyTag(tag1, tag2)
                    .matchObjectAge(MatchObjectAge.builder()
                        .daysGreaterThan(30)
                        .daysLessThan(90).build())
                    .matchObjectSize(MatchObjectSize.builder()
                        .bytesGreaterThan(1000L)
                        .bytesLessThan(6000L).build())
                    .build();

            StorageLensGroupFilter andFilter = StorageLensGroupFilter.builder()
                .and(andOperator)
                .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                .name(storageLensGroupName)
                .filter(andFilter)
                .build();
        }
    }
}
```

```

        CreateStorageLensGroupRequest createStorageLensGroupRequest =
CreateStorageLensGroupRequest.builder()
    .storageLensGroup(storageLensGroup)
    .accountId(accountId).build();

    S3ControlClient s3ControlClient = S3ControlClient.builder()
        .region(Region.US_WEST_2)
        .credentialsProvider(ProfileCredentialsProvider.create())
        .build();
    s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
}
}

```

Example — Crie um grupo da Lente de Armazenamento com um operador **OR** que inclua vários filtros

O exemplo a seguir cria um grupo de Lente de Armazenamento chamado *Marketing-Department*. Este grupo usa um operador OR para aplicar um filtro de prefixo (*prefix-1*, *prefix-2*, *prefix3/sub-prefix-1*) ou um filtro de tamanho de objeto com intervalo de tamanho entre *1000* bytes e *6000* bytes. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```

package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.MatchObjectSize;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupOrOperator;

```

```
public class CreateStorageLensGroupWithOrFilter {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            StorageLensGroupOrOperator orOperator =
StorageLensGroupOrOperator.builder()
                .matchAnyPrefix("prefix-1", "prefix-2", "prefix-3/sub-prefix-1")
                .matchObjectSize(MatchObjectSize.builder()
                    .bytesGreaterThan(1000L)
                    .bytesLessThan(6000L)
                    .build())
                .build();

            StorageLensGroupFilter orFilter = StorageLensGroupFilter.builder()
                .or(orOperator)
                .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                .name(storageLensGroupName)
                .filter(orFilter)
                .build();

            CreateStorageLensGroupRequest createStorageLensGroupRequest =
CreateStorageLensGroupRequest.builder()
                .storageLensGroup(storageLensGroup)
                .accountId(accountId).build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
            s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

```
}
```

Example – Crie um grupo da Lente de Armazenamento com um único filtro e duas tags de recursos AWS

O exemplo a seguir cria um grupo da Lente de Armazenamento chamado *Marketing-Department*, que tem um filtro de sufixo. Este exemplo também adiciona duas tags de recursos AWS ao grupo da Lente de Armazenamento. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;
import software.amazon.awssdk.services.s3control.model.Tag;

public class CreateStorageLensGroupWithResourceTags {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            // Create AWS resource tags.
            Tag resourceTag1 = Tag.builder()
                .key("resource-tag-key-1")
                .value("resource-tag-value-1")
                .build();

            Tag resourceTag2 = Tag.builder()
                .key("resource-tag-key-2")
                .value("resource-tag-value-2")
                .build();

            StorageLensGroupFilter suffixFilter = StorageLensGroupFilter.builder()
                .matchAnySuffix(".png", ".gif", ".jpg")
                .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
```

```
        .name(storageLensGroupName)
        .filter(suffixFilter)
        .build();

        CreateStorageLensGroupRequest createStorageLensGroupRequest =
CreateStorageLensGroupRequest.builder()
        .storageLensGroup(storageLensGroup)
        .tags(resourceTag1, resourceTag2)
        .accountId(accountId).build();

        S3ControlClient s3ControlClient = S3ControlClient.builder()
        .region(Region.US_WEST_2)
        .credentialsProvider(ProfileCredentialsProvider.create())
        .build();
        s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it and returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

Para obter configurações de exemplo JSON, consulte [Configuração de grupos da Lente de Armazenamento](#).

Anexar ao painel ou remover do painel grupos de Lente de Armazenamento do S3

Depois de atualizar para o nível avançado na Lente de Armazenamento do Amazon S3, você poderá anexar um [grupo de Lente de Armazenamento](#) ao seu painel. Se você tiver vários grupos de Lente de Armazenamento, poderá incluir ou excluir os grupos desejados.

Seus grupos de Lente de Armazenamento devem residir na região inicial designada na conta do painel. Depois de anexar um grupo de Lente de Armazenamento ao painel, você receberá os dados adicionais de agregação do grupo de Lente de Armazenamento na exportação de métricas em até 48 horas.

Note

Para visualizar métricas agregadas ao grupo de Lente de Armazenamento, você deve anexá-las ao painel da Lente de Armazenamento. Para obter exemplos de arquivos de configuração JSON do grupo de Lente de Armazenamento, consulte [Exemplo de configuração da Lente de Armazenamento do S3 com grupos de Lente de Armazenamento em JSON](#).

Anexar um grupo de Lente de Armazenamento a um painel do Lente de Armazenamento do S3

Para anexar um grupo de Lente de Armazenamento a um painel da Lente de Armazenamento

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 Lente de Armazenamento, selecione Painéis.
3. Selecione botão de opção para o painel da Lente de Armazenamento ao qual você deseja anexar um grupo e Lente de Armazenamento.
4. Escolha Editar.
5. Em Seleção de métricas, escolha Métricas e recomendações avançadas.
6. Selecione a opção Agregação do grupo de Lente de Armazenamento.

Note

Por padrão, a opção Métricas avançadas também é selecionada. No entanto, você também pode desmarcar essa configuração, pois ela não é necessária para agregar dados de grupos de Lente de Armazenamento.

7. Role para baixo até Agregação do grupo de Lente de Armazenamento e especifique o grupo ou grupos de Lente de Armazenamento que você deseja incluir ou excluir na agregação de dados. Você pode usar as seguintes opções de filtro:
 - Para incluir determinados grupos de Lente de Armazenamento, escolha Incluir grupos de Lente de Armazenamento. Em Grupos de Lente de Armazenamento a serem incluídos, selecione seus grupos de Lente de Armazenamento.
 - Para incluir todos os grupos de Lente de Armazenamento, selecione Incluir todos os grupos de Lente de Armazenamento na região de origem nesta conta.

- Para excluir determinados grupos de Lente de Armazenamento, selecione Excluir grupos de Lente de Armazenamento. Em Grupos de Lente de Armazenamento a serem excluídos, selecione os grupos de Lente de Armazenamento que você deseja excluir.
8. Escolha Salvar alterações. Se tiver configurado seus grupos de Lente de Armazenamento corretamente, você verá os dados adicionais de agregação do grupo de Lente de Armazenamento em seu painel dentro de 48 horas.

Remover um grupo de Lente de Armazenamento de um painel da Lente de Armazenamento do S3

Como remover um grupo de Lente de Armazenamento de um painel da Lente de Armazenamento 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, em Lente de Armazenamento, selecione Painéis.
3. Selecione o botão de opção para o painel da Lente de Armazenamento do qual você deseja remover um grupo de Lente de Armazenamento.
4. Selecione Exibir configuração do painel.
5. Escolha Editar.
6. Role para baixo até a seção Seleção de métricas.
7. Em Agregação de grupo de Lente de Armazenamento, selecione o X ao lado do grupo de Lente de Armazenamento que você deseja remover. Isso remove o grupo de Lente de Armazenamento.

Se você tiver incluído todos os grupos de Lente de Armazenamento em seu painel, desmarque a caixa de seleção ao lado de Incluir todos os grupos de Lente de Armazenamento na região inicial nesta conta.

8. Escolha Salvar alterações.

Note

Levará até 48 horas para que seu painel mostre as atualizações de configuração.

Visualização de dados dos grupos de Lente de Armazenamento

Visualize os dados dos grupos de Lente de Armazenamento [anexando o grupo ao painel de Lente de Armazenamento do Amazon S3](#). Depois de incluir o grupo de Lente de Armazenamento na agregação do grupo na configuração do painel, pode levar até 48 horas para que os dados do grupo de Lente de Armazenamento sejam exibidos no painel.

Depois que a configuração do painel for atualizada, todos os grupos de Lente de Armazenamento recém-anexados aparecerão na lista de recursos disponíveis, na guia Grupos de Lente de Armazenamento. Você também pode analisar melhor o uso do armazenamento na guia Visão geral. É só dividir os dados em outra dimensão. Por exemplo, você pode selecionar um dos itens listados nas três principais categorias e escolher Analisar por para dividir os dados em outra dimensão. Não é possível aplicar a mesma dimensão do próprio filtro.

Note

Não é possível aplicar um filtro de grupo e Lente de Armazenamento junto com um filtro de prefixo e vice-versa. Você também não pode analisar de maneira mais aprofundada um grupo de Lente de Armazenamento usando um filtro de prefixo.

Você pode usar a guia Grupos de Lente de Armazenamento no painel de Lente de Armazenamento do Amazon S3 para personalizar a visualização de dados para os grupos de Lente de Armazenamento anexados ao seu painel. Você pode visualizar os dados de alguns grupos de Lente de Armazenamento que estão anexados ao seu painel ou de todos eles.

Ao visualizar os dados do grupo de Lente de Armazenamento no painel de Lente de Armazenamento do S3, esteja ciente do seguinte:

- A Lente de Armazenamento do S3 agrega métricas de uso de um objeto em todos os grupos de Lente de Armazenamento correspondentes. Portanto, se um objeto corresponder às condições de filtro de dois ou mais grupos de Lente de Armazenamento, você verá contagens repetidas do mesmo objeto em todo o uso do armazenamento.
- Os objetos devem corresponder aos filtros incluídos nos grupos de Lente de Armazenamento. Se nenhum objeto corresponder aos filtros incluídos no grupo de Lente de Armazenamento, nenhuma métrica será gerada. Para determinar se há objetos não atribuídos, verifique a contagem total de objetos no painel no nível da conta e do bucket.

Atualização do grupo de Lente de Armazenamento

Os exemplos a seguir demonstram como atualizar um grupo de Lente de Armazenamento do Amazon S3. Você pode atualizar um grupo de Lente de Armazenamento usando o console Amazon S3, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Uso do console do S3

Para atualizar um grupo de Lente de Armazenamento

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, selecione Grupos de Lente de Armazenamento.
3. Em Grupos de Lente de Armazenamento, selecione o grupo a ser atualizado.
4. Em Escopo, selecione Editar.
5. Na página Escopo, selecione o filtro a ser aplicado no grupo de Lente de Armazenamento. Para aplicar vários filtros, selecione os filtros e escolha o operador lógico AND ou OR.
 - Para o filtro Prefixos, selecione Prefixos e insira uma sequência de prefixo. Para adicionar vários prefixos, selecione Adicionar prefixo. Para remover um prefixo, selecione Remover ao lado do prefixo para remover.
 - Para o filtro Tags de objeto, insira o par de valores-chave ao objeto. Selecione Adicionar tag. Para remover uma tag, selecione Remover ao lado da tag para remover.
 - Para o filtro Sufixos, selecione Sufixos e insira uma sequência de sufixo. Para adicionar vários sufixos, selecione Adicionar sufixo. Para remover um sufixo, selecione Remover ao lado do sufixo para remover.
 - Para o filtro Idade, especifique a faixa etária do objeto em dias. Selecione Especificar idade mínima do objeto e insira a idade mínima do objeto. Em Especificar idade máxima do objeto, insira a idade máxima do objeto.
 - Para o filtro Tamanho, especifique a faixa de tamanho do objeto e a unidade de medida. Selecione Especificar tamanho mínimo do objeto e insira o tamanho mínimo do objeto. Em Especificar tamanho máximo do objeto, insira o tamanho máximo do objeto.
6. Escolha Salvar alterações. A página de detalhes do grupo de Lente de Armazenamento é exibida.
7. (Opcional) Para adicionar uma nova tag de recursos AWS, acesse a seção Tags de recursos AWS e selecione Adicionar tags. A página Adicionar tags é exibida.

Adicione o novo par de valores-chave e selecione Salvar alterações. A página de detalhes do grupo de Lente de Armazenamento é exibida.

- (Opcional) Para remover uma tag de recursos existente AWS, acesse a seção Tags de recurso AWS e selecione a tag de recursos. Em seguida, selecione Excluir. A caixa de diálogo Excluir tags AWS é exibida.

Selecione Excluir novamente para excluir permanentemente a tag de recursos AWS.

Note

Depois de excluir permanentemente uma tag de recursos AWS, ela não poderá ser restaurada.

Usar a AWS CLI

O comando de exemplo AWS CLI a seguir mostra os detalhes da configuração de um grupo de Lente de Armazenamento *marketing-department*. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control get-storage-lens-group --account-id 111122223333 \  
--region us-east-1 --name marketing-department
```

O exemplo AWS CLI a seguir atualiza um grupo de Lente de Armazenamento. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control update-storage-lens-group --account-id 111122223333 \  
--region us-east-1 --storage-lens-group=file://./marketing-department.json
```

Para obter configurações de exemplo JSON, consulte [Configuração de grupos da Lente de Armazenamento](#).

Usar o AWS SDK para Java

O exemplo AWS SDK for Java a seguir mostra os detalhes da configuração do grupo de Lente de Armazenamento *Marketing-Department* na conta *111122223333*. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupResponse;

public class GetStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            GetStorageLensGroupRequest getRequest =
                GetStorageLensGroupRequest.builder()
                    .name(storageLensGroupName)
                    .accountId(accountId).build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
            GetStorageLensGroupResponse response =
                s3ControlClient.getStorageLensGroup(getRequest);
            System.out.println(response);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

O exemplo a seguir atualiza o grupo de Lente de Armazenamento *Marketing-Department* na conta *111122223333*. Esse exemplo atualiza o escopo do painel para incluir objetos que

correspondam a qualquer um dos seguintes sufixos: *.png*, *.gif*, *.jpg* ou *.jpeg*. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;
import software.amazon.awssdk.services.s3control.model.UpdateStorageLensGroupRequest;

public class UpdateStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            // Create updated filter.
            StorageLensGroupFilter suffixFilter = StorageLensGroupFilter.builder()
                .matchAnySuffix(".png", ".gif", ".jpg", ".jpeg")
                .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                .name(storageLensGroupName)
                .filter(suffixFilter)
                .build();

            UpdateStorageLensGroupRequest updateStorageLensGroupRequest =
                UpdateStorageLensGroupRequest.builder()
                    .name(storageLensGroupName)
                    .storageLensGroup(storageLensGroup)
                    .accountId(accountId)
                    .build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
            s3ControlClient.updateStorageLensGroup(updateStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
```

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

Para obter configurações de exemplo JSON, consulte [Configuração de grupos da Lente de Armazenamento](#).

Gerenciamento de tags de recursos AWS com grupos de Lente de Armazenamento

Cada grupo de Lente de Armazenamento do Amazon S3 é contado como um recurso AWS com o próprio nome do recurso da Amazon (ARN). Portanto, ao configurar o grupo de Lente de Armazenamento, você tem a opção de adicionar tags de recursos AWS ao grupo. Você pode adicionar até 50 tags para cada grupo de Lente de Armazenamento. Para criar um grupo de Lente de Armazenamento com tags, você deve ter as permissões `s3:CreateStorageLensGroup` e `s3:TagResource`.

Use tags de recursos AWS para categorizar os recursos de acordo com o departamento, a linha de negócios ou o projeto. Fazer isso é útil quando você tem muitos recursos do mesmo tipo. Ao aplicar tags, é possível identificar rapidamente um grupo específico de Lente de Armazenamento com base nas tags atribuídas a ele. Você também pode usar tags para rastrear e alocar custos.

Além disso, ao adicionar uma tag de recursos AWS ao grupo de Lente de Armazenamento, você ativa o [controle de acesso por atributo \(ABAC\)](#). ABAC é uma estratégia de autorização que define permissões com base em atributos, neste caso, tags. É possível usar condições que especificam tags de recursos nas políticas do IAM para [controlar o acesso aos recursos AWS](#).

É possível editar chaves de tags e valores, e é possível remover as tags de um recurso a qualquer momento. Além disso, esteja ciente das seguintes limitações:

- As chaves e os valores das tags diferenciam maiúsculas de minúsculas.
- Ao adicionar uma tag que tenha a mesma chave de uma tag existente nesse recurso, o novo valor substituirá o antigo.
- Se você excluir um recurso, todas as tags do recurso também serão excluídas.

- Não inclua dados privados ou confidenciais nas tags de recursos AWS.
- As tags do sistema (com chaves de tags que começam com aws :) não são compatíveis.
- O comprimento de cada chave de tag não pode exceder 128 caracteres. O comprimento de cada valor de tag não pode exceder 256 caracteres.

Os exemplos a seguir demonstram como usar tags de recursos AWS com grupos de Lente de Armazenamento.

Tópicos

- [Adicionar tag de recursos AWS a um grupo de Lente de Armazenamento](#)
- [Atualização dos valores das tags de grupo de Lente de Armazenamento](#)
- [Exclusão de tag de recursos AWS do grupo de Lente de Armazenamento](#)
- [Listagem de tags de grupo de Lente de Armazenamento](#)

Adicionar tag de recursos AWS a um grupo de Lente de Armazenamento

Os exemplos a seguir demonstram como adicionar tags de recursos AWS a um grupo de Lente de Armazenamento do Amazon S3. Adicione tags de recurso usando o console do Amazon S3, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Uso do console do S3

Para adicionar uma tag de recursos AWS a um grupo de Lente de Armazenamento

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, selecione Grupos de Lente de Armazenamento.
3. Em Grupos de Lente de Armazenamento, selecione o grupo a ser atualizado.
4. Em Tags de recursos AWS, selecione Adicionar tags.
5. Na página Adicionar tags, adicione o novo par de valores-chave.

Note

Adicionar uma nova tag com a mesma chave de tag existente substitui o valor da tag anterior.

6. (Opcional) Para adicionar mais de uma tag nova, selecione Adicionar tag novamente para continuar adicionando novas entradas. Você pode adicionar até 50 tags de recursos AWS ao seu grupo de Lente de Armazenamento.
7. (Opcional) Para remover uma entrada recém-adicionada, selecione Remover ao lado da tag que você deseja remover.
8. Selecione Save changes.

Usar a AWS CLI

O comando de exemplo AWS CLI a seguir adiciona duas tags de recursos a um grupo de Lente de Armazenamento existente chamado *marketing-department*. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control tag-resource --account-id 111122223333 \  
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/marketing-  
department \  
--region us-east-1 --tags Key=k1,Value=v1 Key=k2,Value=v2
```

Usar o AWS SDK para Java

O exemplo AWS SDK for Java a seguir adiciona duas tags de recursos AWS a um grupo de Lente de Armazenamento existente. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
package aws.example.s3control;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;  
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.s3control.S3ControlClient;  
import software.amazon.awssdk.services.s3control.model.Tag;  
import software.amazon.awssdk.services.s3control.model.TagResourceRequest;  
  
public class TagResource {  
    public static void main(String[] args) {  
        String resourceARN = "Resource_ARN";  
        String accountId = "111122223333";  
  
        try {
```

```
Tag resourceTag1 = Tag.builder()
    .key("resource-tag-key-1")
    .value("resource-tag-value-1")
    .build();
Tag resourceTag2 = Tag.builder()
    .key("resource-tag-key-2")
    .value("resource-tag-value-2")
    .build();
TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
    .resourceArn(resourceARN)
    .tags(resourceTag1, resourceTag2)
    .accountId(accountId)
    .build();
S3ControlClient s3ControlClient = S3ControlClient.builder()
    .region(Region.US_WEST_2)
    .credentialsProvider(ProfileCredentialsProvider.create())
    .build();
s3ControlClient.tagResource(tagResourceRequest);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Atualização dos valores das tags de grupo de Lente de Armazenamento


Os exemplos a seguir demonstram como atualizar os valores das tags de grupo de Lente de Armazenamento usando o console Amazon S3, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Uso do console do S3

Para atualizar uma tag de recursos AWS de um grupo de Lente de Armazenamento

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, selecione Grupos de Lente de Armazenamento.

3. Em Grupos de Lente de Armazenamento, selecione o grupo a ser atualizado.
4. Em Tags de recursos AWS, selecione a tag a ser atualizada.
5. Adicione o novo valor da tag usando a mesma chave do par de valores-chave que você deseja atualizar. Escolha o ícone da marca de seleção para atualizar o valor da tag.

 Note

Adicionar uma nova tag com a mesma chave de tag existente substitui o valor da tag anterior.

6. (Opcional) Para adicionar novas tags, selecione Adicionar tag para adicionar novas entradas. A página Add tags (Adicionar tags) é exibida.

Adicione até 50 tags de recursos AWS ao grupo de Lente de Armazenamento. Quando terminar de adicionar tags, selecione Salvar alterações.

7. (Opcional) Para remover uma entrada recém-adicionada, selecione Remover ao lado da tag que você deseja remover. Ao finalizar a remoção de tags, selecione Salvar alterações.

Usar a AWS CLI

O comando de exemplo AWS CLI a seguir atualiza dois valores de tags para o grupo de Lente de Armazenamento *marketing-department*. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control tag-resource --account-id 111122223333 \  
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/marketing-  
department \  
--region us-east-1 --tags Key=k1,Value=v3 Key=k2,Value=v4
```

Usar o AWS SDK para Java

O exemplo AWS SDK for Java a seguir atualiza dois valores de tags do grupo de Lente de Armazenamento. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
package aws.example.s3control;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;
```

```
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.Tag;
import software.amazon.awssdk.services.s3control.model.TagResourceRequest;

public class UpdateTagsForResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
            Tag updatedResourceTag1 = Tag.builder()
                .key("resource-tag-key-1")
                .value("resource-tag-updated-value-1")
                .build();
            Tag updatedResourceTag2 = Tag.builder()
                .key("resource-tag-key-2")
                .value("resource-tag-updated-value-2")
                .build();
            TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
                .resourceArn(resourceARN)
                .tags(updatedResourceTag1, updatedResourceTag2)
                .accountId(accountId)
                .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
            s3ControlClient.tagResource(tagResourceRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Exclusão de tag de recursos AWS do grupo de Lente de Armazenamento

Os exemplos a seguir demonstram como excluir uma tag de recursos AWS de um grupo de Lente de Armazenamento. Exclua tags usando o console do Amazon S3, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Uso do console do S3

Para excluir uma tags de recurso AWS de um grupo de Lente de Armazenamento

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, selecione Grupos de Lente de Armazenamento.
3. Em Grupos de Lente de Armazenamento, selecione o grupo a ser atualizado.
4. Em Tags de recursos AWS, selecione o par de valores-chave a ser excluído.
5. Escolha Delete (Excluir). A caixa de diálogo Excluir tags de recursos AWS é exibida.

Note

Se tags forem usadas para controlar o acesso, prosseguir com essa ação poderá afetar os recursos relacionados. Depois de excluir permanentemente uma tag, ela não poderá ser restaurada.

6. Selecione Excluir para excluir permanentemente o par de valores-chave.

Usar a AWS CLI

O comando AWS CLI a seguir exclui duas tags de recursos AWS de um grupo de Lente de Armazenamento existente: para usar esse comando de exemplo, substitua *user input placeholders* por suas próprias informações.

```
aws s3control untag-resource --account-id 111122223333 \  
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/Marketing-  
Department \  
--region us-east-1 --tag-keys k1 k2
```

Usar o AWS SDK para Java

O exemplo AWS SDK for Java a seguir exclui duas tags de recursos AWS do nome do recurso da Amazon (ARN) do grupo de Lente de Armazenamento especificado na conta **111122223333**. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.UntagResourceRequest;

public class UntagResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
            String tagKey1 = "resource-tag-key-1";
            String tagKey2 = "resource-tag-key-2";
            UntagResourceRequest untagResourceRequest = UntagResourceRequest.builder()
                .resourceArn(resourceARN)
                .tagKeys(tagKey1, tagKey2)
                .accountId(accountId)
                .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
            s3ControlClient.untagResource(untagResourceRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

```
}
```

Listagem de tags de grupo de Lente de Armazenamento

Os exemplos a seguir demonstram como listar as tags de recursos AWS associadas ao grupo de Lente de Armazenamento. Você pode listar as tags usando o console do Amazon S3, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Uso do console do S3

Para revisar a lista de tags e valores de tags para o grupo de Lente de Armazenamento

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, selecione Grupos de Lente de Armazenamento.
3. Em Grupos de Lente de Armazenamento, selecione o grupo de Lente de Armazenamento da sua preferência.
4. Role para baixo até a seção Tags de recursos AWS. Todas as tags de recursos AWS definidas pelo usuário que são adicionadas ao grupo de Lente de Armazenamento são listadas junto com os valores de tag.

Usar a AWS CLI

O comando de exemplo AWS CLI a seguir lista todos os valores de tag do grupo de Lente de Armazenamento para o grupo de Lente de Armazenamento *marketing-department*. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control list-tags-for-resource --account-id 111122223333 \  
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/marketing-  
department \  
--region us-east-1
```

Usar o AWS SDK para Java

O exemplo AWS SDK for Java a seguir lista os valores de tag do grupo de Lente de Armazenamento para o nome do recurso da Amazon (ARN) do grupo de Lente de Armazenamento que você especifica. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.ListTagsForResourceRequest;
import software.amazon.awssdk.services.s3control.model.ListTagsForResourceResponse;

public class ListTagsForResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
            ListTagsForResourceRequest listTagsForResourceRequest =
ListTagsForResourceRequest.builder()
                .resourceArn(resourceARN)
                .accountId(accountId)
                .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
            ListTagsForResourceResponse response =
s3ControlClient.listTagsForResource(listTagsForResourceRequest);
            System.out.println(response);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Listagem de todos os grupos de Lente de Armazenamento

Os exemplos a seguir demonstram como listar todos os grupos de Lente de Armazenamento do Amazon S3 em uma Conta da AWS e região local. Esses exemplos mostram como listar todos os grupos de Lente de Armazenamento usando o console do Amazon S3, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Uso do console do S3

Para listar todos os grupos de Lente de Armazenamento em uma conta e região local

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, selecione Grupos de Lente de Armazenamento.
3. Em Grupos de Lente de Armazenamento, a lista é exibida na sua conta.

Usar a AWS CLI

O exemplo AWS CLI a seguir lista todos os grupos de Lente de Armazenamento da sua conta. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control list-storage-lens-groups --account-id 111122223333 \  
--region us-east-1
```

Usar o AWS SDK para Java

O exemplo AWS SDK for Java a seguir lista os grupos de Lente de Armazenamento da sua conta **111122223333**. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
package aws.example.s3control;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;  
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.s3control.S3ControlClient;  
import software.amazon.awssdk.services.s3control.model.ListStorageLensGroupsRequest;  
import software.amazon.awssdk.services.s3control.model.ListStorageLensGroupsResponse;
```

```
public class ListStorageLensGroups {
    public static void main(String[] args) {
        String accountId = "111122223333";

        try {
            ListStorageLensGroupsRequest listStorageLensGroupsRequest =
ListStorageLensGroupsRequest.builder()
                .accountId(accountId)
                .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
            ListStorageLensGroupsResponse response =
s3ControlClient.listStorageLensGroups(listStorageLensGroupsRequest);
            System.out.println(response);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Visualização de detalhes do grupo de Lente de Armazenamento

Os exemplos a seguir demonstram como visualizar detalhes de configuração do grupo de Lente de Armazenamento do Amazon S3. Visualize os detalhes usando o console do Amazon S3, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Uso do console do S3

Para visualizar detalhes de configuração do grupo de Lente de Armazenamento

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, selecione Grupos de Lente de Armazenamento.

3. Em Grupos de Lente de Armazenamento, selecione o botão ao lado do grupo de Lente de Armazenamento da sua preferência.
4. Escolha Exibir detalhes. Agora, você pode analisar os detalhes do grupo de Lente de Armazenamento.

Usar a AWS CLI

O exemplo AWS CLI a seguir replica os detalhes da configuração do grupo de Lente de Armazenamento. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control get-storage-lens-group --account-id 111122223333 \  
--region us-east-1 --name marketing-department
```

Usar o AWS SDK para Java

O exemplo AWS SDK for Java a seguir replica os detalhes de configuração do grupo de Lente de Armazenamento *Marketing-Department* na conta *111122223333*. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
package aws.example.s3control;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;  
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.s3control.S3ControlClient;  
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupRequest;  
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupResponse;  
  
public class GetStorageLensGroup {  
    public static void main(String[] args) {  
        String storageLensGroupName = "Marketing-Department";  
        String accountId = "111122223333";  
  
        try {  
            GetStorageLensGroupRequest getRequest =  
                GetStorageLensGroupRequest.builder()  
                    .name(storageLensGroupName)  
                    .accountId(accountId).build();  
            S3ControlClient s3ControlClient = S3ControlClient.builder()
```

```
        .region(Region.US_WEST_2)
        .credentialsProvider(ProfileCredentialsProvider.create())
        .build();

    GetStorageLensGroupResponse response =
s3ControlClient.getStorageLensGroup(getRequest);
    System.out.println(response);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Exclusão de grupo de Lente de Armazenamento

Os exemplos a seguir demonstram como excluir um grupo de Lente de Armazenamento do Amazon S3 usando o console do Amazon S3AWS Command Line Interface, (AWS CLI) e AWS SDK for Java.

Uso do console do S3

Para excluir um grupo de Lente de Armazenamento

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, selecione Grupos de Lente de Armazenamento.
3. Em Grupos de Lente de Armazenamento, selecione o botão ao lado do grupo de Lente de Armazenamento para excluir.
4. Escolha Excluir. Uma caixa de diálogo Excluir grupo de Lente de Armazenamento é exibida.
5. Selecione Excluir novamente para excluir permanentemente o grupo de Lente de Armazenamento.

Note

Depois de excluir um grupo de Lente de Armazenamento, ele não poderá ser restaurado.

Usar a AWS CLI

O exemplo AWS CLI a seguir exclui o grupo de Lente de Armazenamento *marketing-department*. Para usar esse exemplo de comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control delete-storage-lens-group --account-id 111122223333 \  
--region us-east-1 --name marketing-department
```

Usar o AWS SDK para Java

O exemplo AWS SDK for Java a seguir exclui o grupo de Lente de Armazenamento *Marketing-Department* na conta *111122223333*. Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
package aws.example.s3control;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;  
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.s3control.S3ControlClient;  
import software.amazon.awssdk.services.s3control.model.DeleteStorageLensGroupRequest;  
  
public class DeleteStorageLensGroup {  
    public static void main(String[] args) {  
        String storageLensGroupName = "Marketing-Department";  
        String accountId = "111122223333";  
  
        try {  
            DeleteStorageLensGroupRequest deleteStorageLensGroupRequest =  
DeleteStorageLensGroupRequest.builder()  
                .name(storageLensGroupName)  
                .accountId(accountId).build();  
            S3ControlClient s3ControlClient = S3ControlClient.builder()  
                .region(Region.US_WEST_2)  
                .credentialsProvider(ProfileCredentialsProvider.create())  
                .build();  
            s3ControlClient.deleteStorageLensGroup(deleteStorageLensGroupRequest);  
        } catch (AmazonServiceException e) {  
            // The call was transmitted successfully, but Amazon S3 couldn't process  
            // it and returned an error response.        }  
    }  
}
```

```
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
```

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](#)
- [Regiões disponíveis](#)

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 agrega 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](#)
- [Habilitar a hospedagem de sites](#)
- [Configurar um documento de índice](#)
- [Configurar um documento de erro personalizado](#)
- [Configuração de permissões para acesso ao site](#)
- [\(Opcional\) Registrar em log o tráfego da web](#)

- [\(Opcional\) Configurar um redirecionamento de uma página da Web](#)

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](#).

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](#).

Note

Para aumentar a segurança dos sites estáticos do Amazon S3, os domínios de endpoint do site do Amazon S3 (por exemplo, `s3-website-us-east-1.amazonaws.com` ou `s3-website-ap-south-1.amazonaws.com`) são registrados na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` - caso precise definir cookies confidenciais no nome de domínio de sites estáticos do Amazon S3. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

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 ter mais informações, consulte [Configuração de permissões para acesso ao site](#).

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](#).

Tópicos

- [Exemplos de endpoint de site](#)
- [Adicionar um registro DNS CNAME](#)
- [Usar um domínio personalizado com o Route 53](#)
- [Principais diferenças entre um endpoint de site e um endpoint de API REST](#)

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](#).

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 [Tutorial: Configurar um site estático usando um domínio personalizado registrado no Route 53](#).

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 apenas a conteúdo publicamente legível

Principal diferença	Endpoint de API REST	Endpoint de site
	Oferece suporte a conteúdo público e privado	
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 [Tutorial: Configurar um site estático usando um domínio personalizado registrado no Route 53](#).

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](#).

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 ter mais informações, consulte [Configurar um documento de erro personalizado](#).

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](#).

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](#).

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](#).

```
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](#).

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

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

use Aws\S3\S3Client;

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

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

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

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

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

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](#) e [Configuração de permissões para acesso ao site](#).

Também é possível configurar um [documento de erro](#), o [registro em log do tráfego da web](#) ou um [redirecionamento](#).

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](#).

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 de objetos 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 de objeto será `photos/sample.jpg`.

Se você criar um 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.

Note

Em um bucket com versionamento habilitado, é possível carregar várias cópias do `index.html`, mas somente a versão mais recente será resolvida. Para obter mais informações sobre como usar o versionamento do S3, consulte [Usando o versionamento em buckets do S3](#).

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 Documentos 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](#).

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](#).

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](#).

Também é possível configurar um [documento de erro](#), o [registro em log do tráfego da web](#) ou um [redirecionamento](#).

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](#)
- [Configurar um documento de erro personalizado](#)

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 (<a href="http://s3-website. <i>Region</i>.amazonaws.com/">http://s3-website. <i>Region</i>.amazonaws.com/), o Amazon S3 retornará uma resposta 301 Moved Permanently e redirecionará essas solicitações para https://aws.amazon.com/s3/ .
302 Found (302 Encontrado)	Quando o Amazon S3 recebe uma solicitação para uma chave <code>x</code> , <a href="http://<i>bucket-name</i>.s3-website. <i>Region</i>.amazonaws.com/x">http://<i>bucket-name</i>.s3-website. <i>Region</i>.amazonaws.com/x , sem uma barra no final, ele tenta localizar o objeto com o nome de chave <code>x</code> . Se o objeto não 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:

Código de erro HTTP	Descrição
	<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>
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: DdQezl9vkuw5luD5HKsFaTDm9KH4PZzCPRkW3igimLbTu1DiYlvXjgyd7pVxq32

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](#) e [Configurar um documento de erro personalizado](#).

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](#).

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 [Acelerar seu site com o Amazon](#)

[CloudFront](#) ou [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](#)
- [Etapa 2: Adicionar uma política de bucket](#)
- [Listas de controle de acesso de objetos](#)

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](#).

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](#) 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 bloqueie 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](#) 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 bloqueie todo o acesso público aos seus 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.

A Propriedade de objetos do S3 é uma configuração no nível do bucket do Amazon S3 que você pode usar para controlar a propriedade de objetos carregados no bucket e para desabilitar ou habilitar as ACLs. Por padrão, a Propriedade de Objetos está definida com a configuração Imposto pelo proprietário do bucket e todas as ACLs estão desabilitadas. Quando as ACLs são desabilitadas, o proprietário do bucket possui todos os objetos do bucket e gerencia o acesso a eles exclusivamente usando políticas de gerenciamento de acesso.

A maioria dos casos de uso modernos no Amazon S3 não exige mais o uso de ACLs. Recomendamos manter as ACLs desabilitadas, exceto em circunstâncias incomuns em que seja necessário controlar o acesso para cada objeto individualmente. Com as ACLs desabilitadas, é possível usar políticas para controlar o acesso a todos os objetos no bucket, independentemente de quem carregou os objetos para o bucket. Para ter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

Important

Se o bucket usar a configuração Imposto pelo proprietário do bucket para a Propriedade de Objetos do S3, será necessário usar políticas para conceder acesso ao bucket e aos objetos contidos nele. Quando a configuração Imposto pelo proprietário do bucket estiver habilitada, as solicitações para definir listas de controle de acesso (ACLs) ou atualizar ACLs falharão e

retornarão 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 \(ACL\)](#).

```
<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](#). Se planeja usar o Amazon CloudFront para [acelerar seu site](#), 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`).
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`).

Para ter mais informações, consulte [Acelerar seu site com o Amazon CloudFront](#).

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](#)
- [Configurar regras de redirecionamento para usar redirecionamentos condicionais avançados](#)
- [Redirecionar solicitações para um objeto](#)

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](#).

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](#).

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](#). 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](#).

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
RoutingRules	Contêiner para um conjunto de elementos RoutingRule .
RoutingRule	<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 RoutingRules deve ter pelo menos uma regra de roteamento.
Condition	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.

Nome	Descrição
KeyPrefixEquals	<p>O prefixo do nome da chave de objeto do qual as solicitações são redirecionadas.</p> <p>KeyPrefixEquals será obrigatório se HttpStatusCodeReturnedEquals não for especificado. Se KeyPrefixEquals e HttpStatusCodeReturnedEquals forem especificados, ambos deverão ser verdadeiros para a condição ser satisfeita.</p>
HttpStatusCodeReturnedEquals	<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>HttpStatusCodeReturnedEquals será obrigatório se KeyPrefixEquals não for especificado. Se KeyPrefixEquals e HttpStatusCodeReturnedEquals forem especificados, ambos deverão ser verdadeiros para a condição ser satisfeita.</p>
Redirect	<p>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 RoutingRule deve ter um elemento Redirect. Um elemento Redirect deve conter pelo menos um dos seguintes elementos irmãos: Protocol, HostName, ReplaceKeyPrefixWith , ReplaceKeyWith ou HttpRedirectCode .</p>
Protocol	<p>O protocolo, http ou https, que será usado no cabeçalho Location que é retornado na resposta.</p> <p>Se um dos irmãos for fornecido, Protocol não será obrigatório.</p>

Nome	Descrição
HostName	<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>
ReplaceKeyPrefixWith	<p>O prefixo do nome de chave de objeto que substitui o valor de <code>KeyPrefixEquals</code> na solicitação de redirecionamento.</p> <p>Se um dos irmãos for fornecido, <code>ReplaceKeyPrefixWith</code> não será obrigatório. Poderá ser fornecido somente se <code>ReplaceKeyWith</code> não for fornecido.</p>
ReplaceKeyWith	<p>A chave de objeto a ser usada no cabeçalho <code>Location</code> que é retornado na resposta.</p> <p>Se um dos irmãos for fornecido, <code>ReplaceKeyWith</code> não será obrigatório. Poderá ser fornecido somente se <code>ReplaceKeyPrefixWith</code> não for fornecido.</p>
HttpRedirectCode	<p>O código de redirecionamento HTTP a ser usado no cabeçalho <code>Location</code> que é retornado na resposta.</p> <p>Se um dos irmãos for fornecido, <code>HttpRedirectCode</code> não será obrigatório.</p>

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](#), 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](#).

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](#).

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.

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Tópicos

- [Fazer solicitações](#)
- [Desenvolvimento com o Amazon S3 usando a AWS CLI](#)
- [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers](#)
- [Desenvolver com o Amazon S3 usando a API REST](#)
- [Tratar erros de REST e SOAP](#)
- [Referência do desenvolvedor](#)

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 a obtenção de chaves de acesso, consulte [Como obter credenciais de segurança?](#) no 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](#)
- [Endpoints de solicitações](#)
- [Fazer solicitações para o Amazon S3 por meio do IPv6](#)
- [Fazer solicitações usando os AWS SDKs](#)
- [Fazer solicitações usando a API REST](#)

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 em sua Conta da AWS com suas próprias chaves de acesso e anexe políticas de usuário do IAM que concedam as permissões de acesso aos recursos apropriados para esses usuários. 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 usuário 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](#). Para obter informações sobre o envio de solicitações usando AWS SDKs, consulte [Fazer solicitações usando os AWS SDKs](#).

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 [Exigir MFA](#). 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 Endpoints os respectivos endpoints, 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](#)
- [Como usar endereços do IPv6 em políticas do IAM](#)
- [Testar a compatibilidade com endereços IP](#)
- [Usar endpoints de pilha dupla do Amazon S3](#)

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](#).
- 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](#).
- 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](#) e [Registrar em log as solicitações com registro em log de acesso ao servidor](#).

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. A performance deve ser a mesma também.

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

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](#).
- Os AWS SDKs, consulte [Usar endpoints de pilha dupla dos AWS SDKs](#).
- A API REST, consulte [Fazer solicitações para endpoints de pilha dupla usando a API REST](#).

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 [Gerenciamento de identidade e acesso para o Amazon S3](#).

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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
```



```
"Principal": "*",
"Action": "s3:*",
"Resource": "arn:aws:s3:::examplebucket/*",
"Condition": {
  "IpAddress": {"aws:SourceIp": "54.240.143.0/24"}
}
]
```

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

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

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

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](#).

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](#)
- [Usar endpoints de pilha dupla da AWS CLI](#)
- [Usar endpoints de pilha dupla dos AWS SDKs](#)
- [Usar endpoints de pilha dupla da API REST](#)

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](#).

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 [Acessar e listar um bucket do Amazon S3](#). Para obter uma lista dos endpoints do 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 ter mais informações, consulte [Conceitos básicos do Amazon S3 Transfer Acceleration](#).

Note

Os dois tipos de endpoint da VPC para acessar o Amazon S3 (endpoints de interface da VPC e endpoints de VPC do gateway) não são compatíveis com pilha dupla. Para obter mais informações sobre os endpoints da VPC para o Amazon S3, consulte [AWS PrivateLink for Amazon S3](#).

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](#).

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 [Usando a AWS CLI](#).

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 Exemplo do endpoint de pilha dupla do

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](#).

```
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](#).

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

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](#).

Fazer solicitações usando os AWS SDKs

Tópicos

- [Fazer solicitações usando credenciais de usuário do IAM ou da Conta da AWS](#)
- [Fazer solicitações usando credenciais temporárias de usuário do IAM](#)
- [Fazer solicitações usando credenciais temporárias de usuário federado](#)

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](#)
- [Enviar solicitações autenticadas usando os AWS SDKs](#)
- [Recursos relacionados](#)

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, 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. Para obter mais informações sobre o envio de solicitações autenticadas, consulte [Credenciais de segurança da AWS](#) ou [Autenticação do IAM Identity Center](#).

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](#).

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](#).

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 os usuários necessários. Você também pode gerenciar credenciais para esses usuários. 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 recuperar ativamente 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](#).

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](#) e [Visão geral dos buckets](#). 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](#).

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](#) 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\S3\Exception\S3Exception;
use Aws\S3\S3Client;

$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](#). 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](#).

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](#).

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
  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

# Example usage:
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

# Example usage:
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__
```

Go

Example

O exemplo a seguir usa credenciais da AWS carregadas automaticamente pelo SDK para Go pelo arquivo de credenciais compartilhado.

```
package main

import (
    "context"
    "fmt"

    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/s3"
)

// main uses the AWS SDK for Go V2 to create an Amazon Simple Storage Service
// (Amazon S3) client and list up to 10 buckets in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
    sdkConfig, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        fmt.Println("Couldn't load default configuration. Have you set up your AWS
account?")
        fmt.Println(err)
        return
    }
    s3Client := s3.NewFromConfig(sdkConfig)
    count := 10
    fmt.Printf("Let's list up to %v buckets for your account.\n", count)
    result, err := s3Client.ListBuckets(context.TODO(), &s3.ListBucketsInput{})
    if err != nil {
        fmt.Printf("Couldn't list buckets for your account. Here's why: %v\n", err)
        return
    }
    if len(result.Buckets) == 0 {
        fmt.Println("You don't have any buckets!")
    } else {
        if count > len(result.Buckets) {
            count = len(result.Buckets)
        }
        for _, bucket := range result.Buckets[:count] {
            fmt.Printf("\t\t%v\n", *bucket.Name)
        }
    }
}
```

Recursos relacionados

- [Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers](#)
- [AWS SDK for PHP for Amazon S3, classe Aws\S3\S3Client](#)
- [Documentação do AWS SDK for PHP](#)

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](#)) 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](#).

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](#).
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](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.AWSSStaticCredentialsProvider;
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 =
AWSecurityTokenServiceClientBuilder.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 = stsClient.assumeRole(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](#).

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](#).
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](#).

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

```
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](#) 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](#).

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](#).

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](#).

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

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

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

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

$sessionToken = $sts->getSessionToken();

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

```
]);  
  
$result = $s3->listBuckets();  
  
try {  
    // Retrieve a paginator for listing objects.  
    $objects = $s3->getPaginator('ListObjects', [  
        'Bucket' => $bucket  
    ]);  
  
    echo "Keys retrieved!" . PHP_EOL;  
  
    // List objects  
    foreach ($objects as $object) {  
        echo $object['Key'] . PHP_EOL;  
    }  
} catch (S3Exception $e) {  
    echo $e->getMessage() . PHP_EOL;  
}
```

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](#).

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

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](#).
- 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](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.AWSSStaticCredentialsProvider;
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.AWSSecurityTokenService;
import com.amazonaws.services.securitytoken.AWSSecurityTokenServiceClientBuilder;
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 {
            AWSSecurityTokenService stsClient = AWSSecurityTokenServiceClientBuilder
                .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](#).

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](#).
- 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](#).

```
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](#) 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](#).

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](#).

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\S3\Exception\S3Exception;  
use Aws\S3\S3Client;  
use Aws\Sts\StsClient;  
  
$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](#).

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

# Example usage:
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](#)
- [AWS SDK for PHP for Amazon S3, classe Aws\S3\S3Client](#)
- [Documentação do AWS SDK for PHP](#)

Fazer solicitações usando a API REST

Esta seção contém informações sobre como fazer solicitações para endpoints do Amazon S3 usando a API REST. Para obter uma lista dos endpoints do Amazon S3, consulte [Regiões e endpoints](#) 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 [Endpoints e cotas do Amazon S3](#) na 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](#).

Exemplo 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](#).

```
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
```

Exemplo 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](#).

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.

Exemplo 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
```

Exemplo 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](#).

Para obter mais informações sobre como fazer solicitações usando a API REST, consulte os tópicos abaixo.

Tópicos

- [Hospedagem virtual de buckets](#)
- [Redirecionamento de solicitação e a API REST](#)

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 [Endpoints e cotas do Amazon S3](#) na 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](#)
- [Solicitações no estilo de hospedagem virtual](#)
- [Especificação de bucket do cabeçalho Host HTTP](#)
- [Exemplos](#)
- [Personalizar URLs do Amazon S3 com registros CNAME](#)
- [Como associar um nome de host a um bucket do Amazon S3](#)
- [Limitações](#)
- [Compatibilidade retroativa](#)

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.

⚠ Warning

Ao hospedar o conteúdo de um site que será acessado de um navegador da web, evite usar URLs no estilo de caminho, que podem interferir no modelo de segurança da mesma origem do navegador. Para hospedar o conteúdo do site, recomendamos que você use os endpoints do site do S3 ou uma distribuição do CloudFront. Para obter mais informações, consulte [Endpoints de site](#) e [Implantar uma aplicação de página única baseado em reação no Amazon S3 e no CloudFront](#) nos Padrões de orientação de perspectiva 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](#). Para obter mais informações sobre chaves, consulte [Chaves](#).

Example – Solicitações e URLs no estilo de hospedagem virtual

Este exemplo usa o seguinte:

- Nome do bucket: DOC-EXAMPLE-BUCKET1
- Região: Europa (Irlanda)
- Nome da chave: homepage.html

O URL é o seguinte:

```
http://DOC-EXAMPLE-BUCKET1.s3.eu-west-1.amazonaws.com/homepage.html
```

A solicitação é a seguinte:

```
GET /homepage.html HTTP/1.1  
Host: DOC-EXAMPLE-BUCKET1.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](#).

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](#). Para criar um site estático usando um domínio personalizado, consulte [Tutorial: Configurar um site estático usando um domínio personalizado registrado no Route 53](#).

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 [Endpoints e cotas do Amazon S3](#) na 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](#).

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](#)
- [Redirecionamentos e 100-Continue](#)
- [Exemplo de redirecionamento](#)

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](#) 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: "a2c8d6b872054293afd41061e93bc289"
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. Depois, você pode acessar a AWS usando um 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 [Named profiles for the 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](#)
- [Especificar a versão da assinatura na autenticação de solicitações](#)
- [Usar a AWS SDK for Java](#)
- [Usar a AWS SDK for .NET](#)
- [Usar o AWS SDK for PHP e executar exemplos do PHP](#)
- [Usar o AWS SDK for Ruby - versão 3](#)

- [Usar a AWS SDK for Python \(Boto\)](#)
- [Usar os AWS Mobile SDKs for iOS e Android](#)
- [Uso da biblioteca JavaScript do AWS Amplify](#)
- [Usar a AWS SDK for JavaScript](#)

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

AWSO 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 do AWS SDK for C++
AWS SDK for Go	Exemplos de código do AWS SDK for Go
AWS SDK for Java	Exemplos de código do AWS SDK for Java
AWS SDK for JavaScript	Exemplos de código do AWS SDK for JavaScript
AWS SDK para Kotlin	Exemplos de código do AWS SDK para Kotlin
AWS SDK for .NET	Exemplos de código do AWS SDK for .NET
AWS SDK for PHP	Exemplos de código do AWS SDK for PHP
AWS SDK for Python (Boto3)	Exemplos de código do AWS SDK for Python (Boto3)
AWS SDK for Ruby	Exemplos de código do AWS SDK for Ruby
AWS SDK para Rust	Exemplos de código do AWS SDK para Rust
SDK da AWS para SAP ABAP	Exemplos de código do SDK da AWS para SAP ABAP
AWS SDK for Swift	Exemplos de código do AWS SDK for Swift

Para obter exemplos específicos deste serviço, consulte [Exemplos de código do Amazon S3 usando SDKs da AWS](#).

Exemplo de disponibilidade

Você não consegue encontrar o que precisa? Solicite um código de exemplo no link Fornecer feedback 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](#).

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	Adicione o seguinte ao código:

SDK	Solicitar o Signature versão 4 para autenticação de solicitações <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	Defina o parâmetro <code>signatureVersion</code> como <code>v4</code> ao criar o cliente: <pre>var s3 = new AWS.S3({signatureVersion: 'v4'});</pre>
SDK do PHP	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: <pre><?php \$client = S3Client::factory(['region' => 'YOUR-REGION', 'version' => 'latest', 'signature' => 'v4']);</pre> <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	Solicitar o Signature versão 4 para autenticação de solicitações
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>AWSConfigsS3.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 como assinar solicitações da AWS, consulte [Assinar 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 versão 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](#).

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](#).

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](#).

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.	Sim	Especificar a versão da assinatura na autenticação de solicitações
AWS SDK for Java v2	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	Sim	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
	ou versão superior.		
AWS SDK for JavaScript v3	Nenhuma outra ação é necessária no momento. Atualize para a versão principal V3 no terceiro trimestre de 2019.	Não	AWS SDK for JavaScript
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

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 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](#).

Tópicos

- [A organização da API Java](#)
- [Testar exemplos de código Java no Amazon S3](#)

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](#)).

Tópicos

- [A organização da API .NET](#)
- [Executar os exemplos de código do Amazon S3 .NET](#)

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](#)), 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.

Fornecer métodos que correspondem à API REST multipart upload do Amazon S3 (consulte [Uso dos REST API](#)).

- `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](#)
- [Executar exemplos do PHP](#)
- [Recursos relacionados](#)

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](#).

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](#)). 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	<p>Crie um script do SDK para Ruby e adicione as seguintes linhas à parte superior do script.</p> <pre>#!/usr/bin/env ruby require 'rubygems' require 'aws-sdk-s3'</pre> <p>A primeira linha é a diretiva do intérprete e as duas instruções <code>require</code> importam duas gems necessárias no script.</p>
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](#)
- [Fazer upload de objetos](#)

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](#)

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. O AWS Amplify fornece componentes de interface do usuário personalizáveis e uma interface declarativa para trabalhar com um bucket do S3, além de outras categorias de alto nível para serviços da .

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](#)

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 \(v2\)](#)
- [Referência da API do AWS SDK for JavaScript para o Amazon S3 \(v2\)](#)
- [O que é o AWS SDK for JavaScript? \(v3\)](#)
- [AWS SDK for JavaScript: exemplos do Amazon S3 \(v3\)](#)
- [Referência da API do AWS SDK for JavaScript para o Amazon S3 \(v3\)](#)

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 para criar, buscar e excluir bucket e objetos.

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

A API REST usa os cabeçalhos padrão e os códigos de status HTTP, para que os navegadores e os toolkits padrão funcionem como esperado. Em algumas áreas, adicionamos funcionalidade ao HTTP (por exemplo, adicionamos cabeçalhos para oferecer suporte ao controle de acesso). Nesses casos, fizemos o melhor para adicionar nova funcionalidade de uma forma que 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](#). Sobre algumas considerações que você deve ter em mente ao usar a API REST, consulte os tópicos abaixo.

Para obter mais informações sobre a API REST DO Amazon S3, consulte a [Referência da API do Amazon Simple Storage Service](#).

Tópicos

- [Roteamento de solicitação](#)

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 [Acessar e listar um bucket do Amazon S3](#).

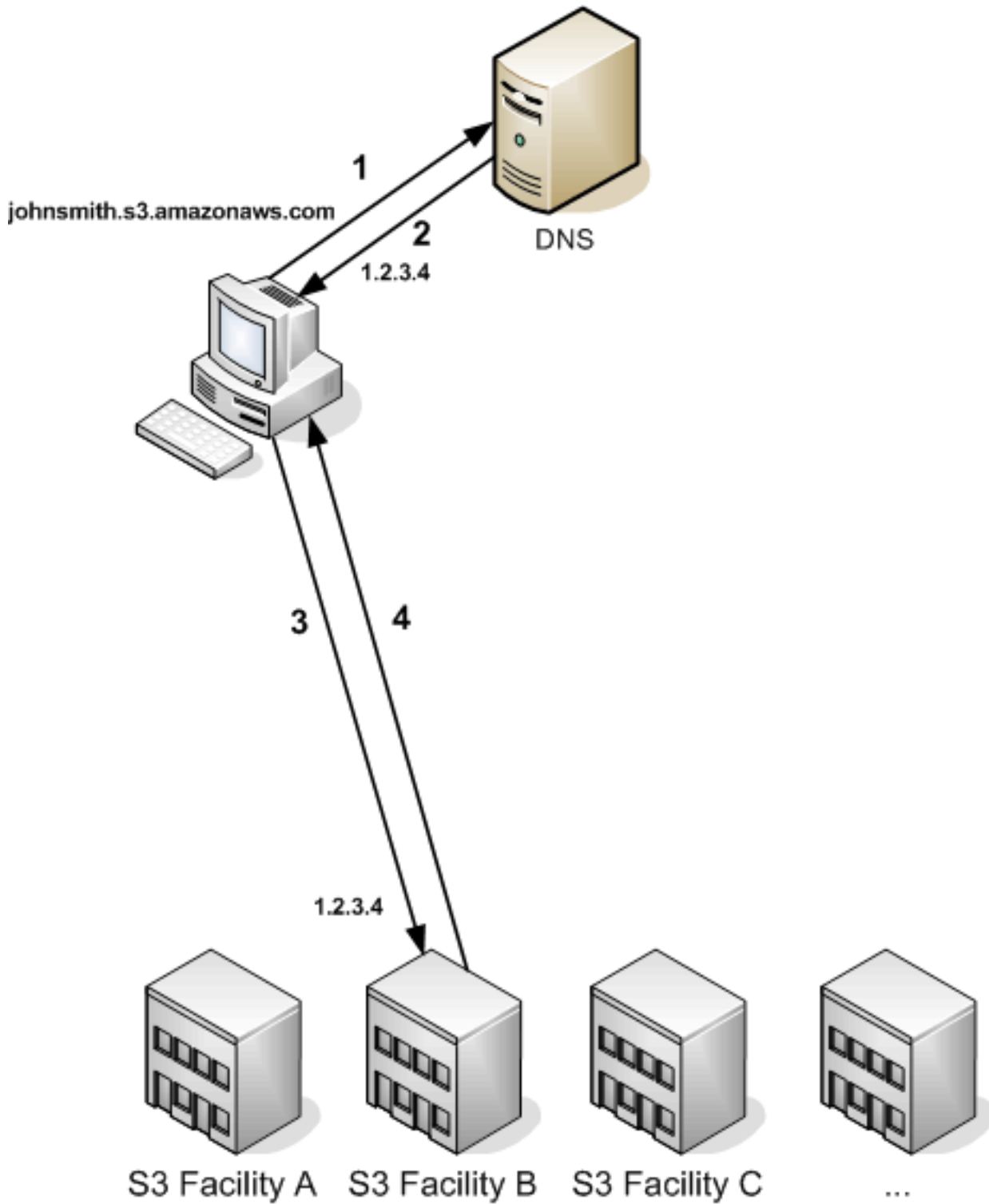
Para todas as regiões lançadas após 20 de março de 2019, se uma solicitação chega no local errado do Amazon S3, o Amazon S3 retorna um erro de solicitação incorreta HTTP 400. Para obter mais informações sobre como habilitar ou desabilitar uma Região da AWS, consulte [Regiões da AWS e endpoints](#) na Referência geral da AWS.

Tópicos

- [Roteamento de DNS](#)
- [Redirecionamento de solicitação temporário](#)
- [Redirecionamento permanente de solicitação](#)
- [Exemplos de redirecionamento de solicitação](#)

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.

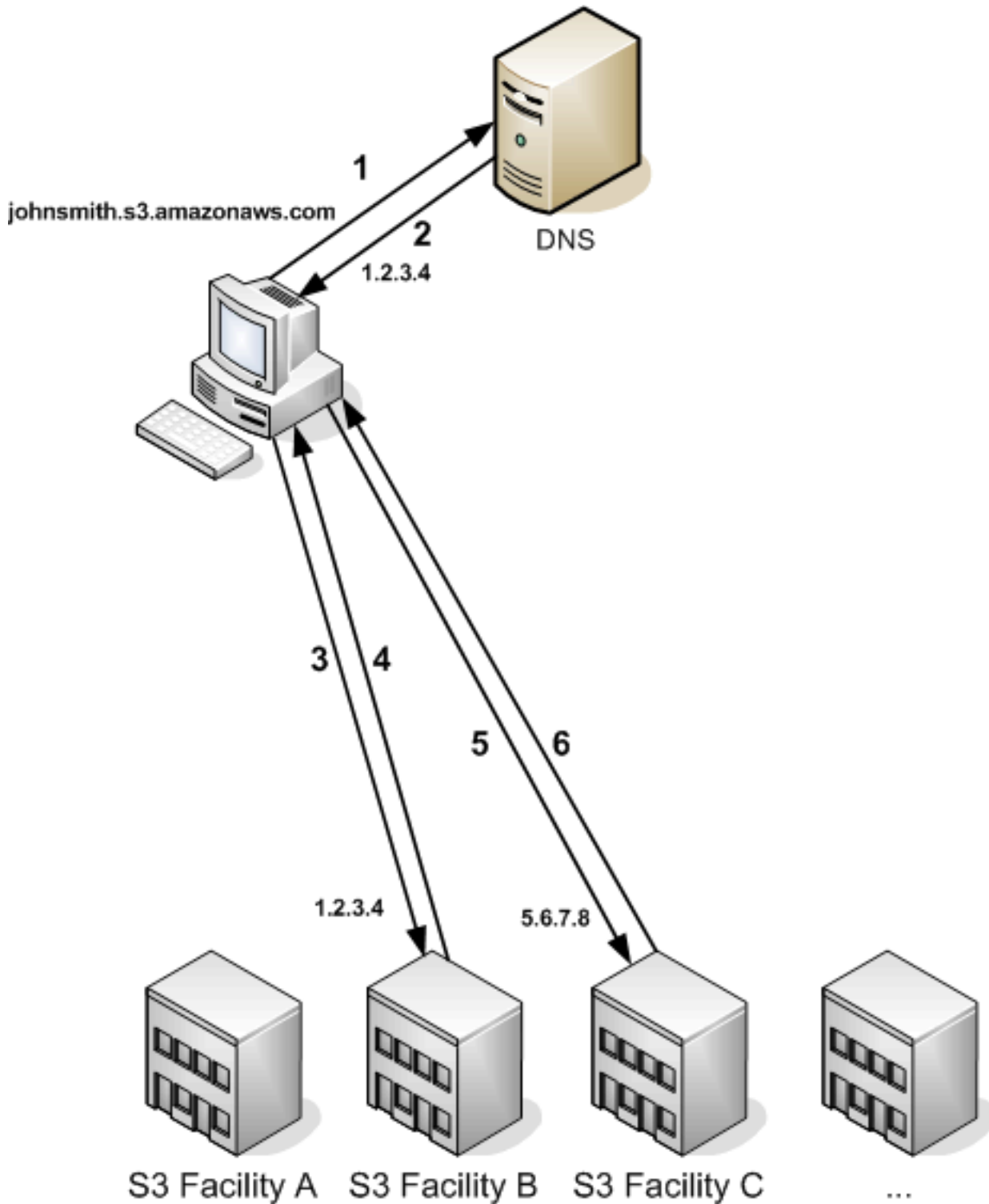
Por exemplo, se você criar um bucket novo e, em seguida, fizer uma solicitação a esse bucket, é possível que receba um redirecionamento temporário dependendo da restrição de localização do bucket. Se você criou o bucket na Região da AWS Leste dos EUA (Norte da Virgínia), não verá o redirecionamento, 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 [Acessar e listar um bucket do Amazon S3](#).

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-gztlb4pa9sq.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](#)
- [A resposta de erro de SOAP](#)
- [Melhores práticas com relação a erros do Amazon S3](#)

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](#)
- [Apêndice B: Autenticação de solicitações \(AWS Signature Version 2\)](#)

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](#)
- [Autenticar solicitações SOAP](#)
- [Configurar políticas de acesso padrão com SOAP](#)

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](#).

- **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 [Gerenciamento de identidade e acesso para o Amazon S3](#). 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>"828ef3fdfa96f00ad9f27c383fc9ac7f"</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](#)

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 de Signature Version 2 do Amazon S3 usando o CloudTrail](#) 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 Version 2 usando logs de acesso do Amazon S3](#)

Tópicos

- [Autenticar solicitações usando a API REST](#)
- [Assinar e autenticar as solicitações REST](#)
- [Uploads baseados no navegador usando POST \(AWS Signature Version 2\)](#)

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 pela AWS.

- 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.

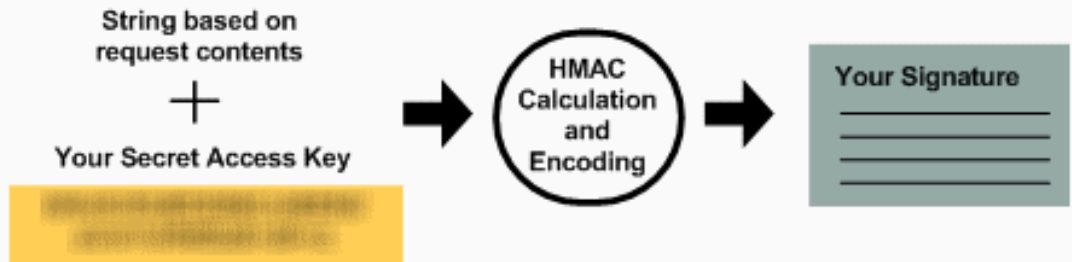
Veja a seguir as etapas gerais para autenticar solicitações ao Amazon S3. Pressupõe-se que você tem as credenciais de segurança necessárias, o ID de chave de acesso e a chave de acesso secreta.

You

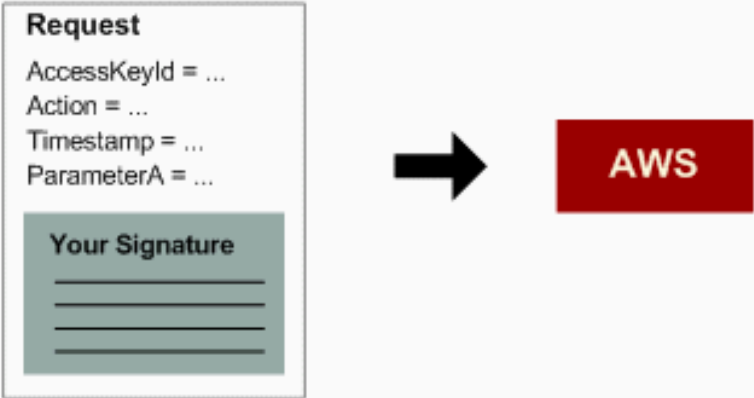
1 Create a request:



2 Create an HMAC-SHA1 signature:

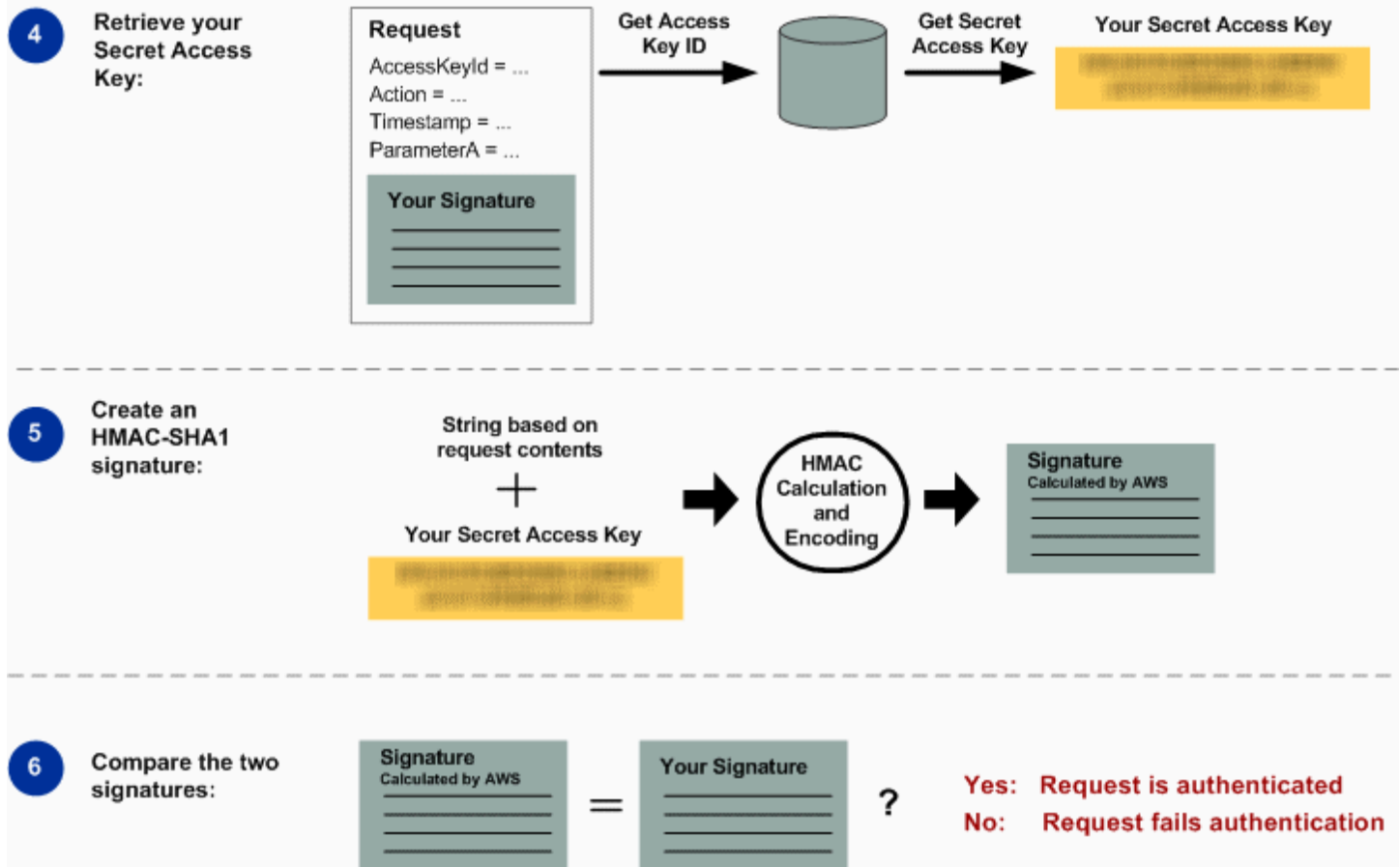


3 Send the request and signature to AWS:



- 1 Crie uma solicitação para a AWS.
- 2 Calcule a assinatura usando a chave de acesso secreta.
- 3 Envie a solicitação para o Amazon S3. Inclua o ID de chave de acesso e a assinatura na solicitação. O Amazon S3 executa as próximas três etapas.

AWS



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](#).

Assinar e autenticar as solicitações REST

Tópicos

- [Uso de credenciais de segurança temporárias](#)
- [Cabeçalho de autenticação](#)
- [Canonização de solicitação para assinatura](#)
- [Criar o elemento CanonicalizedResource](#)
- [Criar o elemento CanonicalizedAmzHeaders](#)
- [Elementos StringToSign de cabeçalho HTTP posicionais versus nomeados](#)
- [Requisito de time stamp](#)
- [Exemplos de autenticação](#)
- [Problemas de assinatura de solicitação REST](#)
- [Alternativa de autenticação de solicitação por string de consulta](#)

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 \(AWS Signature Version 2\)](#).

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.

Exemplo 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/oM7G3qLEjTH1a111g=
```

Uso de credenciais de segurança temporárias

Se você assinar sua solicitação usando credenciais de segurança temporárias (consulte [Fazer solicitações](#)), 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 `Authorization` 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 `Authorization` 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 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](#).

Para uma solicitação no estilo de hospedagem virtual "https://awsexamplebucket1.s3.us-west-1.amazonaws.com/photos/puppy.jpg", o CanonicalizedResource é "/awsexamplebucket1".

Para uma solicitação em estilo de caminho, "https://s3.us-west-1.amazonaws.com/awsexamplebucket1/photos/puppy.jpg", o CanonicalizedResource é "".

- 3 Adicione a parte do caminho de um URI de solicitação HTTP decodificado, até a query string, mas sem incluí-la.

Para uma solicitação no estilo de hospedagem virtual "https://awsexamplebucket1.s3.us-west-1.amazonaws.com/photos/puppy.jpg", o CanonicalizedResource é "/awsexamplebucket1/photos/puppy.jpg".

Para uma solicitação no estilo de hospedagem virtual "https://s3.us-west-1.amazonaws.com/awsexamplebucket1/photos/puppy.jpg", o CanonicalizedResource é "/awsexamplebucket1/photos/puppy.jpg". Neste ponto, o CanonicalizedResource é o mesmo para a solicitação em estilo hosted virtual e em estilo de caminho.

Para uma solicitação que não seja endereçada a um bucket, como [GET Service](#), adicione "/".

4 Se a solicitação endereça um sub-recurso, como `?versioning` , `?location` , `?acl`, `?lifecycle` ou `?versionid` , 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, `?acl&versionId=vaLue`.

Os sub-recursos que devem ser incluídos ao criar o elemento `CanonicalizedResource` são `acl`, `lifecycle`, `location`, `logging`, `notification`, `partNumber`, `policy`, `requestPayment`, `uploadId`, `uploads`, `versionId`, `versioning`, `versions` e `website`.

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](#).

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, 'X-Amz-Date ' torna-se 'x-amz-date '.
- 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 'x-amz-meta-username: fred ' e 'x-amz-meta-username: barney ' seriam combinados em único cabeçalho 'x-amz-meta-username: fred,barney '.
- 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 'x-amz-meta-username: fred,barney ' iria se tornar 'x-amz-meta-username:fred,barney '.
- 6 Finalmente, adicione um caractere de nova linha (U+000A) para cada cabeçalho canonizado na lista resultante. Crie o elemento CanonicalizedResource concatenando todos os cabeçalhos dessa lista em uma única string.

Elementos StringToSign de cabeçalho HTTP posicionais versus nomeados

Os primeiros elementos de cabeçalho do `StringToSign` (`Content-Type`, `Date` e `Content-MD5`) são de natureza posicional. `StringToSign` não inclui os nomes desses cabeçalhos, somente seus valores da solicitação. Em contraste, os elementos 'x-amz-' são nomeados. Os nomes de cabeçalho e os valores de cabeçalho aparecem em `StringToSign`.

Se um cabeçalho posicional chamado para a definição de `StringToSign` não estiver presente na sua solicitação (por exemplo, `Content-Type` ou `Content-MD5` são opcionais para solicitações PUT e sem sentido para solicitações GET), substitua a string vazia ("") para essa posição.

Requisito de time stamp

Um time stamp válido (usando o cabeçalho HTTP Date e 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](#).

Algumas bibliotecas de clientes HTTP não expõem a capacidade para configurar o cabeçalho Date e para uma solicitação. Se tiver problemas para incluir o valor do cabeçalho "Date" 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
AWSAccessKeyId	AKIAIOSFODNN7EXAMPLE
AWSecretAccessKey	wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY

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 AKIAIOSF0 DNN7EXAMPLE: qgk2+6Sv9/oM7G3qLEjTH1a1l1g=</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/bPxrFiCYEXAMPLEKEY'.encode("UTF-8")

string_to_sign = 'GET\n\n\nTue, 27 Mar 2007 19:36:42 +0000\n/awsexamplebucket1/
photos/puppy.jpg'.encode("UTF-8")
```

```
signature = base64.b64encode(
    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
<pre>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 Authorization: AWS AKIAIOSFODNN7EXAMPLE LE: iqRzw+ileNPu1fhspnRs8n0jjIA=</pre>	<pre>PUT\n \n image/jpeg\n Tue, 27 Mar 2007 21:15:45 +0000\n /awsexamplebucket1/photos/puppy.jpg</pre>

Observe o cabeçalho `Content-Type` na solicitação e em `StringToSign`. Também observe que o `Content-MD5` está em branco em `StringToSign` porque não está presente na solicitação.

Lista

Este exemplo lista o conteúdo do bucket `awsexamplebucket1`.

Solicitação	StringToSign
<pre>GET /?prefix=photos&max-keys=50&marker=puppy HTTP/1.1</pre>	<pre>GET\n \n</pre>

Solicitação	StringToSign
<pre>User-Agent: Mozilla/5.0 Host: awsexamplebucket1.s3.us-west-1.amazonaws.com Date: Tue, 27 Mar 2007 19:42:41 +0000 Authorization: AWS AKIAIOSFODNN7EXAMPLE: m0WP8eCtspQl5Ahe6L1SozdX9YA=</pre>	<pre>\n Tue, 27 Mar 2007 19:42:41 +0000\n /awsexamplebucket1/</pre>

Observe a barra final no CanonicalizedResource e a ausência de parâmetros de query string.

Fetch

Este exemplo busca o sub-recurso de política de controle de acesso para o bucket “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.

Delete

Este exemplo exclui um objeto do bucket “awsexamplebucket1” usando o estilo de caminho e a alternativa de data.

Solicitação	StringToSign
<pre>DELETE /awsexamplebucket1/photos/uppy.jpg HTTP/1.1</pre>	<pre>DELETE\n \n</pre>

Solicitação	StringToSign
<pre>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>\n Tue, 27 Mar 2007 21:20:26 +0000\n /awsexamplebucket1/photos/puppy.jpg</pre>

Observe como usamos o método alternativo 'x-amz-date' 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 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</pre>	<pre>PUT\n 4gJE4saaMU4BqNR0kLY+lw==\n application/x-download\n Tue, 27 Mar 2007 21:06:08 +0000\n x-amz-acl:public-read\n x-amz-meta-checksumalgorithm:crc32\n x-amz-meta-filechecksum:0x02661779\n x-amz-meta-reviewedby:joe@example.com,jane@example.com\n /static.example.com/db-backup.dat.gz</pre>

Solicitação	StringToSign
<pre>Authorization: AWS AKIAIOSFODNN7EXAMP LE: jtBQa0Aq+DkULFI8qrpwIjGEx0E=</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:qGdzdE RIC03wnaRNKh60qZehG9s=</pre>	<pre>GET\n \n \n Wed, 28 Mar 2007 01:29:59 +0000\n /</pre>

Chaves Unicode

Solicitação	StringToSign
<pre>GET /dictionary/fran%C3%A7ais/pr %c3%a9f%c3%a8re HTTP/1.1 Host: s3.us-west-1.amazonaws.com Date: Wed, 28 Mar 2007 01:49:49 +0000 Authorization: AWS AKIAIOSFODNN7EXAMP LE:DNEZGsoieTZ92F3bUfSPQcbGm1M=</pre>	<pre>GET\n \n \n Wed, 28 Mar 2007 01:49:49 +0000\n /dictionary/fran%C3%A7ais/pr %c3%a9f%c3%a8re</pre>

Note

Os elementos em `StringToSign` que foram derivados da URI de solicitação são obtidos literalmente, incluindo a codificação do URL e a capitalização.

Problemas de assinatura de solicitação REST

Quando a autenticação de solicitação REST falha, o sistema responde à solicitação com um documento de erro em XML. As informações contidas neste documento de erro têm o objetivo de ajudar os desenvolvedores a diagnosticar o problema. Especificamente, o elemento `StringToSign` do documento de erro `SignatureDoesNotMatch` diz exatamente que canonização de solicitação o sistema está usando.

Alguns toolkits inserem silenciosamente os cabeçalhos que você não conhece antecipadamente, como a adição do cabeçalho `Content-Type` durante um PUT. Na maioria desses casos, o valor do cabeçalho inserido permanece constante, permitindo que você descubra os cabeçalhos que faltam, usando ferramentas como `Ethereal` ou o `tcpmon`.

Alternativa de autenticação de solicitação por 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 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 com URLs pré-assinados](#).

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
AWSAccessKeyId	AKIAIOSFODNN7EXAMPLE	Seu ID de chave de acesso da AWS. Especifica a chave de acesso secreta da AWS usada para assinar a solicitação e, indiretamente, a identidade do desenvolvedor que fez a solicitação.
Expires	1141889120	O tempo quando a assinatura vai expirar, especificado como o número de segundos desde o epoch (00:00:00 UTC em 1º de janeiro de 1970). Uma requisição recebida depois desse tempo (de acordo com o servidor) será rejeitada.
Signature	vjbyPxybdZaNmGa%2ByT272YEAiv4%3D	A codificação do URL da codificação Base64 do HMAC-SHA1 de StringToSign.

O método de autenticação de solicitação por query string difere ligeiramente do método comum, mas somente no formato do parâmetro da solicitação Signature e no elemento StringToSign. A seguir está a pseudogramática que ilustra o método de autenticação de solicitação por query string.

```
Signature = URL-Encode( Base64( HMAC-SHA1( 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?AWSAccess KeyId=AKIAIOSFODNN7EXAMPLE& Signature=NpgCjnDzrM%2BWFzo ENXmpNDUsSn8%3D& Expires=1175139620 HTTP/1.1 Host: awsexamplebucket1.s3.us-wes t-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](#) do Amazon S3.

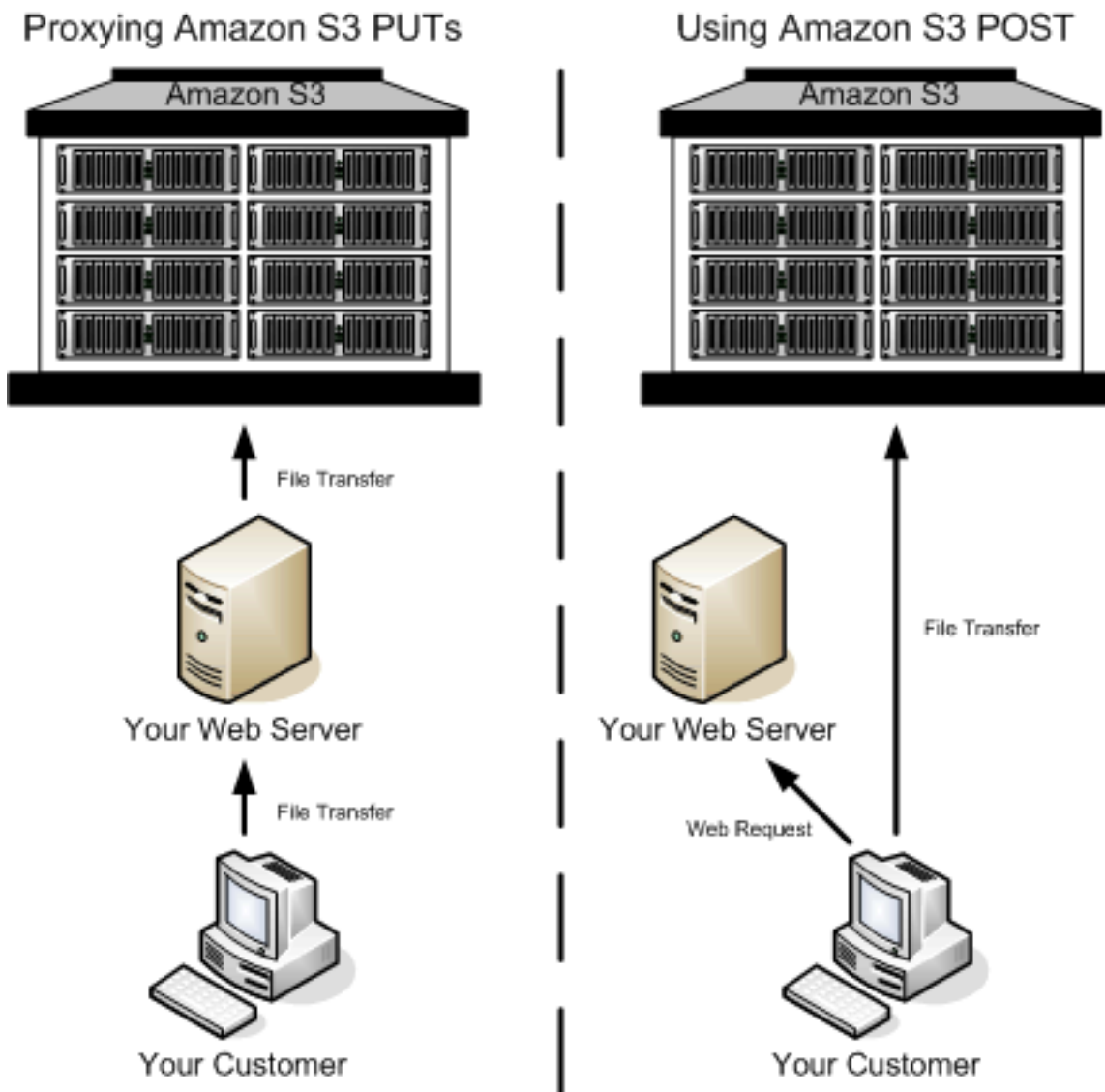
Uploads baseados no navegador usando POST (AWS Signature Version 2)

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.

i Note

Não há suporte para autenticação por query string para POST.

Formulários HTML (AWS Signature Version 2)

Tópicos

- [Codificação do formulário HTML](#)
- [Declaração do formulário HTML](#)
- [Campos do formulário HTML](#)
- [Criação de política](#)
- [Criar uma assinatura](#)
- [Redirecionamento](#)

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
AWSAccessKeyId	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.	Condicional
acl	Uma lista de controle de acesso (ACL) do Amazon S3. Se uma lista de controle de	Não

Nome do campo	Descrição	Obrigatório
	<p>acesso inválida for especificada, um erro será gerado. Para obter mais informações sobre as ACLs, consulte Listas de controle de acesso (ACLs).</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>	
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	<p>Cabeçalhos específicos para REST. Para obter mais informações, consulte Objeto PUT.</p>	Não
key	<p>O nome da chave carregada.</p> <p>Para usar o nome de arquivo fornecido pelo usuário, use a variável <code>\${filename}</code>. Por exemplo, se o usuário Betty carrega o arquivo lolcatz.jpg e você especifica <code>/user/betty/\${filename}</code>, o arquivo é armazenado como <code>/user/betty/lolcatz.jpg</code>.</p> <p>Para obter mais informações, consulte Trabalhar com metadados de objeto.</p>	Sim

Nome do campo	Descrição	Obrigatório
<code>policy</code>	<p>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.</p>	Não
<code>success_action_redirect, 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> <div data-bbox="604 1432 1269 1701" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"><p> Note</p><p>O nome do campo de redirecionamento está obsoleto e o suporte para ele será removido no futuro.</p></div>	Não

Nome do campo	Descrição	Obrigatório
<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>	Não

 **Note**

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 `success_action_status` como 201.

Nome do campo	Descrição	Obrigatório
signature	<p>A assinatura HMAC criada com a chave de acesso secreta correspondente ao <code>AWSAccessKeyId</code> fornecido. Este campo é necessário se um documento de política estiver incluso na solicitação.</p> <p>Para ter mais informações, consulte Gerenciamento de identidade e acesso para o Amazon S3.</p>	Condicional
x-amz-security-token	<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 x-amz-meta-	<p>Metadados especificados pelo usuário.</p> <p>O Amazon S3 não valida ou usa esses dados.</p> <p>Para obter mais informações, consulte Objeto PUT.</p>	Não

Nome do campo	Descrição	Obrigatório
file	<p>Conteúdo de arquivo ou texto.</p> <p>O arquivo ou conteúdo deve ser o último campo no formulário. Todos os campos abaixo deles serão ignorados.</p> <p>Não carregue mais de um arquivo por vez.</p>	Sim

Criação de política

Tópicos

- [Expiração](#)
- [Condições](#)
- [Correspondência de condição](#)
- [Caracteres de escape](#)

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 `AWSSignatureVersion`, 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](#).

A tabela a seguir descreve as condições do documento de política.

Nome do elemento	Descrição
acl	<p>Especifica as condições que a ACL deve satisfazer.</p> <p>Oferece suporte à correspondência exata e a <code>starts-with</code>.</p>
content-length-range	<p>Especifica os tamanhos mínimo e máximo permitidos para o conteúdo carregado.</p> <p>Oferece suporte à correspondência por intervalo.</p>
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	<p>Cabeçalhos específicos para REST.</p> <p>Oferece suporte à correspondência exata e a <code>starts-with</code>.</p>
chave	<p>O nome da chave carregada.</p> <p>Oferece suporte à correspondência exata e a <code>starts-with</code>.</p>
success_action_redirect, redirect	<p>O URL para o qual o cliente é redirecionado após um upload bem-sucedido.</p> <p>Oferece suporte à correspondência exata e a <code>starts-with</code>.</p>
success_action_status	<p>O código de status retornado ao cliente após o upload bem-sucedido se <code>success_action_redirect</code> não for especificado.</p> <p>Oferece suporte à correspondência exata.</p>
x-amz-security-token	<p>Token de segurança do Amazon DevPay.</p>

Nome do elemento	Descrição
	Cada solicitação que usa o Amazon DevPay requer dois campos do formulário <code>x-amz-security-token</code> : um para o token de produto e outro para o token de usuário. Conseqüentemente, os valores devem ser separados por vírgulas. Por exemplo, se o token de usuário é <code>eW91dHVIZQ==</code> e o token de produto for <code>b0hnNVNKWVJIQTA=</code> , defina a entrada da política para: <code>{ "x-amz-security-token": "eW91dHVIZQ==,b0hnNVNKWVJIQTA=" }</code> .
Outros nomes de campos com o prefixo <code>x-amz-meta-</code>	Metadados especificados pelo usuário. Oferece suporte à correspondência exata e a <code>starts-with</code> .

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
Correspon dências exatas	Correspondências exatas verificam se os campos correspondem a valores específicos. Este exemplo indica que a ACL deve ser definida como pública para leitura:

Condição	Descrição
	<pre data-bbox="375 212 1503 289">{"acl": "public-read" }</pre> <p data-bbox="375 352 1503 436">Este exemplo é uma forma alternativa para indicar que a ACL deve ser definida como pública para leitura:</p> <pre data-bbox="375 474 1503 552">["eq", "\$acl", "public-read"]</pre>
Inicia com	<p data-bbox="375 646 1503 730">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 data-bbox="375 768 1503 846">["starts-with", "\$key", "user/betty/"]</pre>
Corresponder qualquer conteúdo	<p data-bbox="375 940 1503 1077">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 data-bbox="375 1115 1503 1192">["starts-with", "\$success_action_redirect", ""]</pre>
Especificar intervalos	<p data-bbox="375 1283 1503 1419">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 data-bbox="375 1457 1503 1535">["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 [Gerenciamento de identidade e acesso para o Amazon S3](#).

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](#)
- [Upload de área de texto](#)

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.

```
eyJhZlZxhwaXJhdGlvbiI6IClYMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFoiLAogICJjb25kaXRpb25zIjogWwogICAgYeyJidW
```

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>
...
</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"

eyJhZXBhYXNjaWJhdGlub3I6IClYMDA3LTExVDEyOjAwOjAwLjAwMFoiLAogICJjb25kaXRpb25zIjogWwogICAgZidW
--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="39d459dfbc0faabbb5e179358dfb94c3"
Server: AmazonS3
```

Upload de área de texto

Tópicos

- [Criação de política e formulário](#)
- [Exemplo de solicitação](#)
- [Exemplo de resposta](#)

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.

```
eyAiZXhwaXJhdGlvbiI6IClyMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFoiLAogICJjb25kaXRpb25zIjogWwogICAgeyJidWNrZXQiOiAiam9obnNtaXRoIn0sCiAgICBbInN0YXJ0cy13aXRoIiwgIiRrZXkiLCAidXNlciLAogICAgeyJhY2wiOiAicHVibGljLXJlYWQifSwKICAgIHsic3VjY2Vzci19hY3Rpb25fcmVkaXJlY3QiOiAiaHR0cDovL2pC5zMy5hbWF6b25hd3MuY29tL25ld19wb3N0Lmh0bWwifSwKICAgIFsiZXEiLCAiJENvbnRlbnQtVHlwZSI6ICJ0ZXh0L2h0CAGIHsieC1hbXotbWV0YS11dWlkIjogIjE0MzY1MTIzNjUxMjc0In0sCiAgICBbInN0YXJ0cy13aXRoIiwgIiR4LWFtei1IsICIiXQogIF0KfQo=
```

Crie uma assinatura usando suas credenciais. Por exemplo, qA7FWXKq6VvU681I9KdveT1cWgF= 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"
eyJhZlZxhwaXJhdGlvbiI6IClYMDA3LTIyLTAxVDEyOjAwOjAwLjAwMFoiLAogICJjb25kaXRpb25zIjogWwogICAgZyJidW91
--178521717625888
Content-Disposition: form-data; name="Signature"

qA7FWXKq6VvU681I9KdveT1cWgF=
--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](#).

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, a 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 do Amazon S3. 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. O ajuste de escala, no caso de operações de leitura e gravação, ocorre gradualmente e não é algo instantâneo. Embora o Amazon S3 esteja escalando para sua nova taxa de solicitação mais alta, você pode ver alguns erros 503 (desaceleração). Esses erros serão dissipados quando o ajuste de escala for concluído. Para obter mais informações sobre como criar e usar prefixos, consulte [Organizar objetos usando prefixos](#).

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](#). 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 do lado do servidor com o AWS 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](#) e [Padrões de design de performance do Amazon S3](#) para obter as informações mais recentes sobre a otimização de performance do Amazon S3.

Note

Para obter mais informações sobre o uso da classe de armazenamento Amazon S3 Express One Zone com buckets de diretório, consulte [O que é a classe S3 Express One Zone?](#) e [Buckets de diretório](#).

Tópicos

- [Diretrizes de performance do Amazon S3](#)
- [Padrões de design de performance do Amazon S3](#)

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](#).

Para obter a melhor performance para a aplicação no Amazon S3, recomendamos as seguintes diretrizes.

Tópicos

- [Avaliar a performance](#)
- [Dimensionar conexões de armazenamento na horizontal](#)
- [Usar consulta na escala de bytes](#)

- [Solicitações de repetição para aplicativos sensíveis à latência](#)
- [Combinar o Amazon S3 \(armazenamento\) e o Amazon EC2 \(computação\) na mesma Região da AWS](#)
- [Usar o Amazon S3 Transfer Acceleration para minimizar a latência causada pela distância](#)
- [Usar a versão mais recente dos AWS SDKs](#)

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.

Para entender os requisitos de performance e otimizar a performance da aplicação, você também pode monitorar as respostas de erro 503 que você recebe. O monitoramento de determinadas métricas de performance pode gerar despesas adicionais. Para obter mais informações, consulte [Preço do Amazon S3](#).

Monitore o número de respostas de erro de status 503 (desaceleração)

Para monitorar o número de respostas de erro de status 503 que você recebe, use uma das seguintes opções:

- Use métricas de solicitação do Amazon CloudWatch para o Amazon S3. As métricas de solicitação do CloudWatch incluem uma métrica para respostas de status 5xx. Para obter mais informações sobre métricas de solicitação do CloudWatch, consulte [Monitoramento de métricas com o Amazon CloudWatch](#).
- Use a contagem de erros 503 (Serviço indisponível) disponível na seção de métricas avançadas da Lente de Armazenamento do Amazon S3. Para ter mais informações, consulte [Usar a Lente de Armazenamento do S3 para melhorar a performance](#).
- Use o registro em log de acesso ao servidor do Amazon S3. Com o registro de acesso ao servidor, é possível filtrar e revisar todas as solicitações que recebem respostas 503 (erro interno). Você também pode usar o Amazon Athena para analisar logs. 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](#).

Ao monitorar o número de códigos de erro de status HTTP 503, muitas vezes você pode obter informações valiosas sobre quais prefixos, chaves ou buckets estão recebendo mais solicitações de controle de utilização.

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 [Baixar objetos](#).

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](#) 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](#) 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](#)
- [Tempo limite e repetição para aplicativos sensíveis à latência](#)
- [Dimensionamento horizontal e paralelização de solicitações para alto throughput](#)
- [Usar o Amazon S3 Transfer Acceleration para acelerar transferências de dados geograficamente dispersas](#)

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 otimizar 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 [Repetições de erro e recuo exponencial na AWS](#) no 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 solucionadores 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](#).

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](#) 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](#).

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 no Outposts, você pode criar buckets do S3 em seu Outposts, além de armazenar e recuperar objetos facilmente on-premises. O S3 no Outposts fornece uma nova classe de armazenamento, OUP0STS, 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 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 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](#)
- [Recursos do S3 on Outposts](#)
- [Serviços relacionados](#)
- [Acessar o S3 no Outposts](#)
- [Pagar pelo S3 on Outposts](#)
- [Próximas etapas](#)

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 AWS service (Serviço da AWS) ou aplicação de cliente 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, use o ARN do ponto de acesso ou o alias do ponto de acesso. Para obter mais informações sobre alias de pontos de acesso, consulte [Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3 no Outposts](#).

O exemplo a seguir mostra o formato do ARN de ponto de acesso do S3 no Outposts, que inclui o `outpost-id`, o `account-id` e o nome do ponto de acesso:

```
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](#).

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 \(ou nome\)](#).

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](#).

Versionamento do S3

Use o versionamento do S3 em buckets do Outposts 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. O versionamento do S3 ajuda você a se recuperar de ações não intencionais de usuários e de falhas da aplicação.

Para obter mais informações, consulte [Gerenciar o versionamento do S3 para um bucket do S3 no Outposts](#).

ID da versão

Se você habilitar o versionamento do S3 em um bucket, o S3 no Outposts 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 [PutObject](#), os novos objetos receberão um ID de versão exclusivo.

Para obter mais informações, consulte [Gerenciar o versionamento do S3 para um bucket do S3 no Outposts](#).

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](#).

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 do Amazon S3](#).

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.

Ao especificar o bucket para operações de objeto, use o ARN do ponto de acesso ou o alias do ponto de acesso. Para obter mais informações sobre alias de pontos de acesso, consulte [Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3 no Outposts](#).

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](#).

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 \(IAM\)](#): o IAM é um serviço da Web que ajuda você a controlar de maneira segura o acesso aos recursos da AWS, incluindo seus recursos do S3 no Outposts. Com o IAM, é possível gerenciar, de maneira centralizada, permissões que controlam quais recursos da AWS os usuários poderão acessar. Você usa o IAM para controlar quem é autenticado (fez login) e autorizado (tem permissões) a usar os recursos.
- [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](#).

Serviços relacionados

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 o 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 do](#) . Para obter mais informações sobre os comandos que você pode usar com o S3 on Outposts, consulte [s3api](#), [s3control](#) e [es3outposts](#) 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](#).

Próximas etapas

Para obter mais informações sobre como trabalhar com o S3 on Outposts, consulte os seguintes tópicos:

- [Configurar seu Outpost](#)
- [O Amazon S3 on Outposts é diferente do Amazon S3?](#)
- [Conceitos básicos do Amazon S3 on Outposts](#)
- [Redes para S3 on Outposts](#)
- [Trabalhar com buckets do S3 on Outposts](#)
- [Trabalhar com objetos usando o S3 on Outposts](#)
- [Segurança no S3 on Outposts](#)
- [Gerenciar o armazenamento do S3 on Outposts](#)
- [Desenvolver com o Amazon S3 on Outposts](#)

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 verificar se o Outposts tem capacidade do S3, você pode usar a chamada de API [ListOutpostsWithS3](#). 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?](#).

Para obter mais informações, consulte os tópicos a seguir.

Tópicos

- [Pedir um novo Outpost](#)

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.

O Amazon S3 on Outposts é diferente do Amazon S3?

O Amazon S3 no Outposts fornece armazenamento de objetos para seu ambiente do AWS Outposts on-premises. Usar o S3 no Outposts ajuda você a atender às necessidades de processamento local, residência de dados e alto nível de performance, mantendo os dados próximos às aplicações on-premises. Como ele usa APIs e recursos do Amazon S3, o S3 no 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](#)
- [Operações de API compatíveis com o Amazon S3 no Outposts](#)
- [Recursos do Amazon S3 não compatíveis com o S3 no Outposts.](#)
- [Requisitos de rede do S3 on Outposts](#)

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](#).

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)
- 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
- Criptografia no lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS)
- 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?](#).

Tópicos

- [Configurar o IAM com o S3 on Outposts](#)
- [Conceitos básicos do uso do AWS Management Console](#)
- [Como começar a usar a AWS CLI e o SDK para Java](#)

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) para utilizar os recursos do Amazon S3 no Outposts. O IAM é um AWS service (Serviço da AWS) que pode ser usado sem custo adicional. Por padrão, os usuários não têm permissões para recursos e operações do S3 no Outposts. Para conceder permissões de acesso para recursos do S3 no Outposts e operações de API, você pode usar o IAM para criar [usuários](#), [grupos](#) ou [perfis](#) e anexar permissões.

Para fornecer o acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos no AWS IAM Identity Center:

Crie um conjunto de permissões. Siga as instruções em [Create a permission set](#) (Criação de um conjunto de permissões) no Guia do usuário do AWS IAM Identity Center.

- Usuários gerenciados no IAM usando um provedor de identidades:

Crie um perfil para a federação de identidades. Siga as instruções em [Criar um perfil para um provedor de identidades de terceiros \(federação\)](#) no Guia do usuário do IAM.

- Usuários do IAM:

- Crie um perfil que seu usuário possa assumir. Siga as instruções em [Creating a role for an IAM user](#) (Criação de um perfil para um usuário do IAM) no Guia do usuário do IAM.

- (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em [Adição de permissões a um usuário \(console\)](#) no Guia do usuário do IAM.

Além das políticas do IAM baseadas em identidade, o S3 no 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](#)
- [ARNs de recurso para S3 no Outposts](#)
- [Exemplo de políticas para S3 no Outposts](#)
- [Permissões para os endpoints do S3 on Outposts](#)
- [Perfis vinculados a serviço para o S3 no Outposts](#)

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](#).

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>: <i>account_id</i> :outpost / <i>outpost_id</i> / bucket/<i>bucket_name</i></code>	<code>arn:aws:s3-outposts: <i>us-west-2</i> :123456789012 :outpost/ <i>op-01ac5d28a6a232904</i> / bucket/<i>DOC-EXAMPLE-BUCKET1</i></code>
ARN do ponto de acesso	<code>arn:<i>partition</i> :s3-outposts: <i>region</i>: <i>account_id</i> :outpost / <i>outpost_id</i> /accesspoint/ <i>accesspoint_name</i></code>	<code>arn:aws:s3-outposts: <i>us-west-2</i> :123456789012 :outpost/ <i>op-01ac5d28a6a232904</i> /accesspoint/ <i>access-point-name</i></code>
ARN do objeto	<code>arn:<i>partition</i> :s3-outposts: <i>region</i>:</code>	<code>arn:aws:s3-outposts: <i>us-west-2</i></code>

ARN do Amazon S3 no Outposts	Formato ARN	Exemplo
	<i>account_id</i> :outpost / <i>outpost_id</i> / bucket/ <i>bucket_name</i> / object/ <i>object_key</i>	:123456789012 : outpost/ <i>op-01ac5d</i> 28a6a232904 / bucket/ <i>DOC-EXAMPLE-</i> <i>BUCKET1</i> /object/ <i>m</i> <i>yobject</i>
ARN do objeto de ponto de acesso do S3 on Outposts (usado nas políticas)	arn: <i>partition</i> :s3- outposts: <i>region</i> : <i>account_id</i> :outpost / <i>outpost_id</i> /accesspo int/ <i>accesspoi</i> <i>nt_name</i> / object/ <i>object_key</i>	arn:aws:s3-outpo sts: <i>us-west-2</i> : <i>123456789012</i> : outpost/ <i>op-01ac5d</i> 28a6a232904 /accesspo int/ <i>access-point-</i> <i>name/object/myobject</i>
ARN do S3 no Outposts	arn: <i>partition</i> :s3- outposts: <i>region</i> : <i>account_id</i> :outpost / <i>outpost_id</i>	arn:aws:s3-outpo sts: <i>us-west-2</i> : <i>123456789012</i> : outpost/ <i>op-01ac5d</i> 28a6a232904

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-outposts-
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-outposts-
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 ColP), você também precisa ter permissões para trabalhar com endereços IP de seu grupo de ColP, 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”](#).

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 ec2:DescribeSubnets ec2:CreateTags iam:CreateServiceLinkedRole Para endpoints que estão usando o tipo de acesso ao grupo de endereços IP de proprieda

Ação	Permissões do IAM
	<p>de 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>
DeleteEndpoint	<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>
ListEndpoints	<code>s3-outposts:ListEndpoints</code>

Note

Você pode usar tags de recurso em uma política do IAM para gerenciar permissões.

Perfis vinculados a serviço para o S3 no Outposts

O S3 no Outposts usa perfis vinculados a serviço do IAM para criar alguns recursos de rede em seu nome. Para obter mais informações, consulte [Usar perfis vinculados a serviço para o Amazon S3 no Outposts](#).

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?](#)

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](#)
- [Próximas etapas](#)

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](#).

 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. Em Bucket Versioning (Versionamento de bucket), defina o estado de versionamento do S3 para seu bucket do S3 no Outposts como uma das seguintes opções:
 - Disable (Desabilitar) (padrão): o bucket permanece sem versão.
 - Enable (Habilitar): habilita o versionamento do S3 para os objetos no bucket. Todos os objetos adicionados ao bucket recebem um ID de versão exclusivo.

Para obter mais informações sobre o S3 Versioning, consulte [Gerenciar o versionamento do S3 para um bucket do S3 no Outposts](#).

7. (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.

8. 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](#).

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](#).

9. 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](#).

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.

10. (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](#).

11. (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.

12. (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.
13. (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.
 14. Escolha Create Outposts bucket (Criar bucket do Outposts).

Note

Pode levar até cinco minutos para que seu endpoint do Outpost seja criado e seu bucket esteja pronto para uso. Para definir configurações de bucket adicionais, escolha View details (Visualizar detalhes).

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](#).

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?](#)

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](#).

Tópicos

- [Etapa 1: Criar um bucket](#)
- [Etapa 2: Criar um ponto de acesso](#)
- [Etapa 3: Criar um endpoint](#)
- [Etapa 4: Fazer upload de um objeto em um bucket do S3 on Outposts](#)

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-outposts-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-outposts-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. Para criar um endpoint, você precisará de uma conexão ativa com seu link de serviço com sua região de origem do 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](#). Você deve criar um endpoint para poder acessar seus buckets do Outposts e executar operações de objeto. Para obter mais informações, consulte [Endpoints](#).

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](#).

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, use o nome do recurso da Amazon (ARN) do ponto de acesso ou o alias do ponto de acesso. Para obter mais informações sobre alias de pontos de acesso, consulte [Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3 no Outposts](#).

O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 no 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 e o nome do ponto de acesso:

```
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](#).

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](#).

```
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](#)
- [Acessar buckets e objetos do S3 on Outposts](#)
- [Interfaces de rede elástica entre contas](#)

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 é acessível pela rede on-premises por meio do roteamento direto de VPC. [Para obter mais informações, consulte Tabelas de rotas do gateway local](#) no Guia do usuário do AWS Outposts.
- `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](#) e [Trabalhar com objetos usando o S3 on Outposts](#).

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 CoIP, o S3 on Outposts aloca e associa endereços IP à interface de rede elástica entre contas do grupo de CoIP 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?](#)

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](#).

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](#).

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	

API	Valor do parâmetro do S3 no Outposts
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](#)
- [Adicionar etiquetas aos buckets do S3 on Outposts](#)
- [Gerenciar o acesso a um bucket do Amazon S3 on Outposts usando uma política de bucket](#)
- [Listar buckets do Amazon S3 on Outposts](#)
- [Obter um bucket do S3 on Outposts usando a AWS CLI e o SDK para Java](#)
- [Excluir seu bucket do S3 on Outposts](#)
- [Trabalhar com pontos de acesso do Amazon S3 on Outposts](#)
- [Trabalhar com endpoints do Amazon S3 on Outposts](#)

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?](#)

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](#).

Se você quiser criar um bucket que use o AWS PrivateLink para fornecer acesso ao gerenciamento de buckets e endpoints por meio de endpoints da VPC de interface na nuvem privada virtual (VPC), consulte [AWS PrivateLink para S3 no Outposts](#).

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](#).

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. Em Bucket Versioning (Versionamento de bucket), defina o estado de versionamento do S3 para seu bucket do S3 no Outposts como uma das seguintes opções:
 - Disable (Desabilitar) (padrão): o bucket permanece sem versão.
 - Enable (Habilitar): habilita o versionamento do S3 para os objetos no bucket. Todos os objetos adicionados ao bucket recebem um ID de versão exclusivo.

Para obter mais informações sobre o S3 Versioning, consulte [Gerenciar o versionamento do S3 para um bucket do S3 no Outposts](#).

7. (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.

8. 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](#).

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](#).

9. 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](#).

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.

10. (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](#).

11. (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.

12. (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.
13. (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.
14. Escolha Create Outposts bucket (Criar bucket do Outposts).

Note

Pode levar até cinco minutos para que seu endpoint do Outpost seja criado e seu bucket esteja pronto para uso. Para definir configurações de bucket adicionais, escolha View details (Visualizar detalhes).

Usar a 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-outposts-bucket --outpost-id op-01ac5d28a6a232904
```

Usar o AWS SDK para 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-outposts-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-outposts-bucket --tagging 'TagSet=[{Key=organization,Value=marketing}]'
```

Para obter mais informações sobre esse comando, consulte [put-bucket-tagging](#) na Referência da AWS CLI.

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](#).

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](#)
- [Visualizar a política do bucket do Amazon S3 on Outposts](#)
- [Excluir a política do bucket do Amazon S3 on Outposts](#)
- [Exemplos de políticas de bucket](#)

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](#).

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](#).

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-outposts-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-outposts-bucket --policy file://policy1.json
```

Usar o AWS SDK para 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](#).

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](#).

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-outposts-bucket
```

Usar o AWS SDK para 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](#).

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-outposts-bucket
```

Exemplos de políticas de bucket

Com as políticas de bucket do S3 no Outposts, você pode proteger o acesso a objetos em seus buckets do S3 no Outposts, para que somente usuários com as permissões apropriadas possam acessá-los. Você pode até mesmo impedir que usuários autenticados sem as permissões apropriadas acessem seus recursos do S3 no Outposts.

Esta seção apresenta exemplos de casos de uso típicos de políticas de bucket do S3 no Outposts. Para testar essas políticas, substitua *user input placeholders* por suas informações (como o nome do seu bucket).

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

Para obter mais informações sobre a linguagem da política do AWS Identity and Access Management (IAM), consulte [Configurar o IAM com o S3 on Outposts](#).

Note

Ao testar as permissões [s3outposts](#) usando o console do Amazon S3, você deve conceder permissões adicionais exigidas pelo console, como `s3outposts:createendpoint`, `s3outposts:listendpoints` e assim por diante.

Recursos adicionais para criar políticas de bucket

- Para conferir uma lista de ações, recursos e chaves de condição de políticas do IAM que você pode usar ao criar uma política de bucket do S3 no Outposts, consulte [Actions, resources, and condition keys for Amazon S3 on Outposts](#).
- Para obter orientação sobre como criar uma política do S3 no Outposts, consulte [Adicionar ou editar uma política para um bucket do Amazon S3 on Outposts](#).

Tópicos

- [Gerenciar o acesso a um bucket do Amazon S3 no Outposts com base em endereços IP específicos](#)

Gerenciar o acesso a um bucket do Amazon S3 no Outposts com base em endereços IP específicos

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 ter mais informações, consulte [Política de bucket](#).

Restringir o acesso a endereços IP específicos

O exemplo a seguir impede que todos os usuários executem [operações do S3 no Outposts](#) em objetos nos buckets especificados, a menos que a solicitação tenha origem no intervalo de endereços IP especificado.

Note

Ao restringir o acesso a um endereço IP específico, especifique quais endpoints da VPC, endereços IP de origem da VPC ou endereços IP externos podem acessar o bucket do S3 no Outposts. Caso contrário, você poderá perder o acesso ao bucket se a sua política negar aos usuários a execução de qualquer operação [s3outposts](#) em objetos no bucket do S3 no Outposts sem as permissões adequadas já em vigor.

A instrução Condition dessa política identifica **192.0.2.0/24** como o intervalo de endereços IP 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. A chave de condição `aws:SourceIp` só pode ser usada para intervalos de endereços IP públicos. Para obter mais informações sobre essas chaves de condição, consulte [Actions, resources, and condition keys for S3 on Outposts](#). 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 do S3 no Outposts, substitua o intervalo de endereços IP **192.0.2.0/24** desse exemplo por um valor apropriado para o seu caso de uso. Caso contrário, você perderá a capacidade de acessar o bucket.

```
{
  "Version": "2012-10-17",
  "Id": "S3OutpostsPolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Deny",
```

```

    "Principal": "*",
    "Action": "s3outposts:*",
    "Resource": [
        "arn:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
accesspoint/EXAMPLE-ACCESS-POINT-NAME"
        "arn:aws:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
bucket/DOC-EXAMPLE-BUCKET"
    ],
    "Condition": {
        "NotIpAddress": {
            "aws:SourceIp": "192.0.2.0/24"
        }
    }
}

```

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. Isso ajudará a garantir que as políticas continuem funcionando à medida que você faz a transição para IPv6.

O exemplo de política de bucket do S3 no Outposts 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 permite acesso aos endereços IP `192.0.2.1` e `2001:DB8:1234:5678::1` e nega acesso aos endereços `203.0.113.1` e `2001:DB8:1234:5678:ABCD::1`.

A chave de condição `aws:SourceIp` só pode ser usada para intervalos de endereços IP públicos. 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 `2001: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 do S3 no Outposts. Caso contrário, você pode perder a capacidade de acessar seu bucket.

```

{
  "Id": "S3OutpostsPolicyId2",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowIPmix",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3outposts:*",
      "Resource": [
        "arn:aws:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/bucket/DOC-EXAMPLE-BUCKET",
        "arn:aws:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/bucket/DOC-EXAMPLE-BUCKET/*"
      ],
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "192.0.2.0/24",
            "2001:DB8:1234:5678::/64"
          ]
        },
        "NotIpAddress": {
          "aws:SourceIp": [
            "203.0.113.0/24",
            "2001:DB8:1234:5678:ABCD::/80"
          ]
        }
      }
    }
  ]
}

```

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?](#)

Para obter mais informações sobre como trabalhar com buckets no S3 on Outposts, consulte [Trabalhar com buckets do S3 on Outposts](#).

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 do AWS CLI a seguir 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?](#)

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](#).

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-outposts-  
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?](#)

Para obter mais informações sobre como trabalhar com buckets no S3 on Outposts, consulte [Trabalhar com buckets do S3 on Outposts](#).

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](#).
- 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-outposts-  
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](#)
- [Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3 no Outposts](#)
- [Exibir informações sobre uma configuração de ponto de acesso](#)
- [Visualizar uma lista dos pontos de acesso do Amazon S3 on Outposts](#)
- [Excluir um ponto de acesso](#)
- [Adicionar ou editar uma política de ponto de acesso](#)
- [Visualizar uma política para um ponto de acesso do S3 on Outposts](#)

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](#).

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-outposts-
bucket" --vpc-configuration VpcId=example-vpc-12345
```

Usar o AWS SDK para 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();
}
```

Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3 no Outposts

Com o S3 on Outposts, você deve usar pontos de acesso para acessar qualquer objeto em um bucket do Outposts. Toda vez que você cria um ponto de acesso para um bucket, o S3 no Outposts gera automaticamente um alias de ponto de acesso. É possível usar esse alias de ponto de acesso em vez de um ARN de ponto de acesso para qualquer operação de plano de dados. Por exemplo, você pode usar um alias de ponto de acesso para realizar operações em nível de objeto, como PUT, GET, LIST e muito mais. Para obter uma lista dessas operações, consulte [Operações de API do Amazon S3 para gerenciar objetos](#).

Os exemplos a seguir mostram um ARN e um alias de ponto de acesso para um ponto de acesso chamado *my-access-point*.

- ARN do ponto de acesso: `arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/my-access-point`
- Alias do ponto de acesso: `my-access-po-001ac5d28a6a232904e8xz5w8ijx1qzlb3i3kuse10--op-s3`

Para obter mais informações sobre ARNs, consulte [Nomes de recurso da Amazon \(ARNs\)](#) no Referência geral da AWS.

Para obter mais informações sobre alias de pontos de acesso, consulte os tópicos a seguir.

Tópicos

- [Alias de pontos de acesso](#)
- [Usar um alias de ponto de acesso em uma operação de objeto do S3 no Outposts](#)
- [Limitações](#)

Alias de pontos de acesso

Cria-se um alias de ponto de acesso dentro do mesmo namespace de um bucket do S3 no Outposts. Quando você cria um ponto de acesso, o S3 no Outposts gera automaticamente um alias de ponto de acesso que não poderá ser alterado. O alias de ponto de acesso atende a todos os requisitos de um nome de bucket válido do S3 no Outposts e consiste nas seguintes partes:

access point name prefix-metadata--op-s3

Note

O sufixo `--op-s3` é reservado para alias de pontos de acesso. Recomendamos que não o use para nomes de bucket ou ponto de acesso. Para obter mais informações sobre as regras para nomes de bucket do S3 no Outposts, consulte [Trabalhar com buckets do S3 on Outposts](#).

Encontrar o alias do ponto de acesso

Os exemplos a seguir mostram como encontrar um ponto de acesso usando o console do Amazon S3 e a AWS CLI.

Example : Encontrar e copiar um alias de ponto de acesso no console do Amazon S3

Depois de criar um ponto de acesso no console, você pode obter o alias do ponto de acesso na coluna Access Point alias (Alias de ponto de acesso) na lista Access Points (Pontos de acesso).

Como copiar um alias de ponto de acesso

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. Para copiar o alias do ponto de acesso, realize um dos seguintes procedimentos:
 - Na lista Access Points (Pontos de acesso), selecione o botão de opção ao lado do nome do ponto de acesso e escolha Copy Access Point alias (Copiar alias do ponto de acesso).
 - Escolha o nome do ponto de acesso. Depois, em Outposts access point overview (Visão geral dos pontos de acesso do Outposts), copie o alias do ponto de acesso.

Example : Criar um ponto de acesso usando a AWS CLI e encontrar o alias do ponto de acesso na resposta

O exemplo da AWS CLI a seguir para o comando `create-access-point` cria o ponto de acesso e retorna o alias do ponto de acesso gerado automaticamente. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control create-access-point --bucket example-outposts-bucket --name example-outposts-access-point --account-id 123456789012
```

```
{
  "AccessPointArn":
    "arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/
    accesspoint/example-outposts-access-point",
  "Alias": "example-outp-001ac5d28a6a232904e8xz5w8ijx1qz1bp3i3kuse10--op-s3"
}
```

Example : Obter um alias de ponto de acesso usando a AWS CLI

O exemplo a seguir da AWS CLI para o comando `get-access-point` retorna informações sobre o ponto de acesso especificado. Essas informações incluem o alias do ponto de acesso. Para executar esse comando, substitua os *user input placeholders* por suas próprias informações.

```
aws s3control get-access-point --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --name example-outposts-access-point --account-id 123456789012

{
  "Name": "example-outposts-access-point",
  "Bucket": "example-outposts-bucket",
  "NetworkOrigin": "Vpc",
  "VpcConfiguration": {
    "VpcId": "vpc-01234567890abcdef"
  },
  "PublicAccessBlockConfiguration": {
    "BlockPublicAcls": true,
    "IgnorePublicAcls": true,
    "BlockPublicPolicy": true,
    "RestrictPublicBuckets": true
  },
  "CreationDate": "2022-09-18T17:49:15.584000+00:00",
  "Alias": "example-outp-o0b1d075431d83bebde8xz5w8ijx1qzlbp3i3kuse10--op-s3"
}
```

Example : Listar os pontos de acesso para encontrar um alias de ponto de acesso usando a AWS CLI

O exemplo a seguir da AWS CLI para o comando `list-access-points` lista informações sobre o ponto de acesso especificado. Essas informações incluem o alias do ponto de acesso. 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-outposts-
bucket

{
  "AccessPointList": [
    {
      "Name": "example-outposts-access-point",
```

```

    "NetworkOrigin": "Vpc",
    "VpcConfiguration": {
      "VpcId": "vpc-01234567890abcdef"
    },
    "Bucket": "example-outposts-bucket",
    "AccessPointArn": "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point",
    "Alias": "example-outp-o0b1d075431d83bebde8xz5w8ijx1qzlp3i3kuse10--op-s3"
  }
]
}

```

Usar um alias de ponto de acesso em uma operação de objeto do S3 no Outposts

Ao adotar pontos de acesso, é possível usar alias de pontos de acesso sem exigir alterações extensas de código.

Este exemplo da AWS CLI mostra uma operação `get-object` para um bucket S3 no Outposts. Este exemplo usa o alias do ponto de acesso como valor para `--bucket` em vez do ARN completo do ponto de acesso.

```

aws s3api get-object --bucket my-access-po-
o0b1d075431d83bebde8xz5w8ijx1qzlp3i3kuse10--op-s3 --key testkey sample-object.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.

- Não é possível usar um alias de ponto de acesso para operações do ambiente de gerenciamento do S3 no Outposts. Para obter uma lista das operações do ambiente de gerenciamento do S3 no Outposts, consulte [Operações de API do Amazon S3 Control para gerenciar buckets](#).
- Não é possível usar alias em políticas do AWS Identity and Access Management (IAM).

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-outposts-bucket
```

Usar o AWS SDK para 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](#).

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](#).

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](#).

Para obter mais informações sobre como trabalhar com pontos de acesso no S3 on Outposts, consulte [Trabalhar com buckets do S3 on Outposts](#).

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](#).

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. Para criar um endpoint, você precisará de uma conexão ativa com seu link de serviço com sua região de origem do 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](#). Você deve criar um endpoint para poder acessar seus buckets do Outposts e executar operações de objeto. Para obter mais informações, consulte [Endpoints](#).

Depois de criar um endpoint, você pode usar o campo “Status” para entender o estado do endpoint. Se seu Outposts estiver offline, ele retornará um CREATE_FAILED. Você pode conferir sua conexão de link de serviço, excluir o endpoint e repetir a operação de criação depois que sua conexão for retomada. Para ter uma lista de códigos de erro adicionais, veja abaixo. Para obter mais informações, consulte [Endpoints](#).

API	Status	Código de erro do motivo da falha	Mensagem: motivo da falha
CreateEndpoint	Create_Failed	OutpostNotReachable	O endpoint não pôde ser criado porque a conexão do link de serviço com sua região de origem do Outposts está inativa. Confira sua conexão, exclua o endpoint e tente novamente.
CreateEndpoint	Create_Failed	InternalError	O endpoint não pôde ser criado devido a um erro interno. Exclua o endpoint e crie novamente.
DeleteEndpoint	Delete_Failed	OutpostNotReachable	O endpoint não pôde ser excluído porque a conexão do link de serviço com sua região de origem do Outposts está inativa. Confira sua conexão e tente novamente.
DeleteEndpoint	Delete_Failed	InternalError	O endpoint não pôde ser excluído devido a um erro interno. Tente novamente.

Para obter mais informações sobre como trabalhar com buckets no S3 on Outposts, consulte [Trabalhar com buckets do S3 on Outposts](#).

As seções a seguir descrevem como criar e gerenciar endpoints para o S3 on Outposts.

Tópicos

- [Criar um endpoint em um Outpost](#)
- [Visualizar uma lista de endpoints do Amazon S3 on Outposts](#)
- [Excluir um endpoint do Amazon S3 on Outposts](#)

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. Para criar um endpoint, você precisará de uma conexão ativa com seu link de serviço com sua região de origem do 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](#). Você deve criar um endpoint para poder acessar seus buckets do Outposts e executar operações de objeto. Para obter mais informações, consulte [Endpoints](#).

Permissões

Para obter mais informações sobre as permissões necessárias para criar um endpoint, consulte [Permissões para os endpoints do S3 on Outposts](#).

Ao criar um endpoint, o S3 no Outposts também cria um perfil vinculado a serviço na Conta da AWS. Para obter mais informações, consulte [Usar perfis vinculados a serviço para o Amazon S3 no Outposts](#).

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](#).

7. Escolha Create Outposts endpoint (Criar endpoint do Outposts).

Usar a 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. Para criar um endpoint, você precisará de uma conexão ativa com seu link de serviço com sua região de origem do 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](#). Você deve criar um endpoint para poder acessar seus buckets do Outposts e executar operações de objeto. Para obter mais informações, consulte [Endpoints](#).

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.

Usar a 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. Para criar um endpoint, você precisará de uma conexão ativa com seu link de serviço com sua região de origem do 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](#). Você deve criar um endpoint para poder acessar seus buckets do Outposts e executar operações de objeto. Para obter mais informações, consulte [Endpoints](#).

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).

Usar a 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 para 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.Arn;
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, use o nome do recurso da Amazon (ARN) do ponto de acesso ou o alias do ponto de acesso. Para obter mais informações sobre alias de pontos de acesso, consulte [Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3 no Outposts](#).

O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 no 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 e o nome do ponto de acesso:

```
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](#).

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](#). 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](#)
- [Obter um objeto de um bucket do Amazon S3 on Outposts](#)
- [Listar os objetos em um bucket do Amazon S3 on Outposts](#)
- [Excluir objetos em buckets do Amazon S3 on Outposts](#)
- [Usar o HeadBucket para determinar se existe um bucket do S3 on Outposts e se você tem permissões de acesso](#)
- [Realizar e gerenciar um carregamento fracionado com o SDK para Java](#)
- [Uso de URLs pré-assinados para o S3 no Outposts](#)
- [Amazon S3 no Outposts com Amazon EMR no Outposts local](#)
- [Armazenamento em cache de autorização e autenticação](#)

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, use o nome do recurso da Amazon (ARN) do ponto de acesso ou o alias do ponto de acesso. Para obter mais informações sobre alias de pontos de acesso, consulte [Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3 no Outposts](#).

O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 no 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 e o nome do ponto de acesso:

```
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](#).

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, use o nome do recurso da Amazon (ARN) do ponto de acesso ou o alias do ponto de acesso. Para obter mais informações sobre alias de pontos de acesso, consulte [Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3 no Outposts](#).

O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 no 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 e o nome do ponto de acesso:

```
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](#).

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, use o nome do recurso da Amazon (ARN) do ponto de acesso ou o alias do ponto de acesso. Para obter mais informações sobre alias de pontos de acesso, consulte [Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3 no Outposts](#).

O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 no 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 e o nome do ponto de acesso:

```
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](#).

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](#).

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, use o nome do recurso da Amazon (ARN) do ponto de acesso ou o alias do ponto de acesso. Para obter mais informações sobre alias de pontos de acesso, consulte [Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3 no Outposts](#).

O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 no 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 e o nome do ponto de acesso:

```
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](#).

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.txt` de um bucket do S3 on Outposts (`s3-outposts:DeleteObjects`) 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, use o nome do recurso da Amazon (ARN) do ponto de acesso ou o alias do ponto de acesso. Para obter mais informações sobre alias de pontos de acesso, consulte [Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3 no Outposts](#).

O exemplo a seguir mostra o formato do ARN para pontos de acesso do S3 no 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 e o nome do ponto de acesso:

```
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](#).

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?](#)

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](#)
- [Copiar um objeto grande em um bucket do S3 on Outposts usando carregamento fracionado](#)
- [Listar partes de um objeto em um bucket do S3 no Outposts](#)

- [Recuperar uma lista de multipart uploads em andamento em um bucket do S3 no Outposts](#)

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](#).

```
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](#).

```
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](#) 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": "444455556666"},
    "Action": "s3-outposts:*",
    "Resource": "arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/objec
t/*",
    "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 \(SigV4\)](#).

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 (por exemplo, políticas de usuário, grupo ou perfil). 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": "NetworkRestrictionForIAMPrincipal",
  "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](#).

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](#).

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](#)
- [Gerar um URL pré-assinado para carregar um objeto em um bucket do S3 no Outposts](#)

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](#).

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](#).

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](#). 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](#).

```
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](#). 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](#).

```
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 expirará 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](#).

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](#).

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](#).

```
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](#). 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()
```

Amazon S3 no Outposts com Amazon EMR no Outposts local

O Amazon EMR é uma plataforma de cluster gerenciada que simplifica a execução de frameworks de big data, como Apache Hadoop e Apache Spark, na AWS a fim de processar e analisar grandes volumes de dados. Ao usar essas estruturas e projetos de código aberto relacionados, é possível processar dados para finalidades analíticas e workloads de inteligência de negócios. O Amazon EMR também ajuda a transformar e mover grandes volumes de dados para dentro e para fora de outros datastores e bancos de dados da AWS, além de oferecer suporte ao Amazon S3 no Outposts. Para obter mais informações sobre o Amazon EMR, consulte [Amazon EMR no Outposts](#) no Guia de gerenciamento do Amazon EMR.

Para o Amazon S3 no Outposts, o Amazon EMR começou a oferecer suporte ao conector S3A do Apache Hadoop na versão 7.0.0. As versões anteriores do Amazon EMR não oferecem suporte ao S3 no Outposts local e o EMR File System (EMRFS) não é compatível.

Aplicações compatíveis

O Amazon EMR com o Amazon S3 no Outposts é compatível com as seguintes aplicações:

- Hadoop
- Spark
- Hue
- Hive
- Sqoop
- Pig
- Hudi
- Flink

Para obter mais informações, consulte o [Guia de versão do Amazon EMR](#).

Criar e configurar um bucket do Amazon S3 no Outposts

O Amazon EMR usa o AWS SDK for Java com o Amazon S3 no Outposts para armazenar dados de entrada e dados de saída. Seus arquivos de log do Amazon EMR são armazenados em um local regional do Amazon S3 selecionado por você e não são armazenados localmente no Outpost. Para obter mais informações, consulte [Logs do Amazon EMR](#) no Guia de gerenciamento do Amazon EMR.

Para estar em conformidade com os requisitos do Amazon S3 e de DNS, os buckets do S3 no Outposts têm restrições e limitações de nomenclatura. Para ter mais informações, consulte [Criar um bucket do S3 on Outposts](#).

Com o Amazon EMR versão 7.0.0 e posterior, você pode usar o Amazon EMR com o S3 no Outposts e o sistema de arquivos S3A.

Pré-requisitos

Permissões do S3 no Outposts: ao criar um perfil de instância do Amazon EMR, esse perfil deve conter o namespace do AWS Identity and Access Management (IAM) para o S3 no Outposts. O S3 no Outposts tem seu próprio namespace: `s3-outposts*`. Para conferir um exemplo de política que usa esse namespace, consulte [Configurar o IAM com o S3 on Outposts](#).

Conector S3A: para configurar um cluster do EMR para acessar dados de um bucket do Amazon S3 no Outposts, você deve usar o conector S3A do Apache Hadoop. Para usar o conector, garanta que todos os URIs do S3 usem o esquema do `s3a`. Caso contrário, você pode configurar a implementação do sistema de arquivos utilizado para o cluster do EMR de modo que os URIs do S3 funcionem com o conector S3A.

Para configurar a implementação do sistema de arquivos de modo que funcione com o conector S3A, use as propriedades de configuração `fs.file_scheme.impl` e `fs.AbstractFileSystem.file_scheme.impl` para o cluster do EMR, em que `file_scheme` corresponde ao tipo de URIs do S3 que você tem. Para usar o exemplo a seguir, substitua `user input placeholders` por suas próprias informações. Por exemplo, para alterar a implementação do sistema de arquivos para URIs do S3 que usam o esquema `s3`, especifique as seguintes propriedades de configuração do cluster:

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "fs.s3.impl": "org.apache.hadoop.fs.s3a.S3AFileSystem",
      "fs.AbstractFileSystem.s3.impl": "org.apache.hadoop.fs.s3a.S3A"
    }
  }
]
```

Para usar o S3A, defina a propriedade de configuração `fs.file_scheme.impl` como `org.apache.hadoop.fs.s3a.S3AFileSystem` e defina a propriedade `fs.AbstractFileSystem.file_scheme.impl` como `org.apache.hadoop.fs.s3a.S3A`.

Por exemplo, se você estiver acessando o caminho `s3a://bucket/...`, defina a propriedade `fs.s3a.impl` como `org.apache.hadoop.fs.s3a.S3AFileSystem` e defina a propriedade `fs.AbstractFileSystem.s3a.impl` como `org.apache.hadoop.fs.s3a.S3A`.

Conceitos básicos do Amazon EMR com o Amazon S3 no Outposts

Os tópicos a seguir explicam como começar a usar o Amazon EMR com o Amazon S3 no Outposts.

Tópicos

- [Criação de uma política de permissões](#)
- [Criação e configuração de um cluster](#)
- [Visão geral das configurações](#)
- [Considerações](#)

Criação de uma política de permissões

Antes de criar um cluster do EMR que use o Amazon S3 on Outposts, é necessário criar uma política do IAM para anexar ao perfil de instância do Amazon EC2 para o cluster. A política deve ter permissões para acessar o nome do recurso da Amazon (ARN) do ponto de acesso do S3 no Outposts. Para obter mais informações sobre como criar políticas do IAM para o S3 no Outposts, consulte [Configurar o IAM com o S3 on Outposts](#).

O exemplo de política a seguir mostra como conceder as permissões necessárias. Após criar a política, anexe-a à função do perfil de instância usada para criar seu cluster do EMR, conforme descrito na seção [the section called “Criação e configuração de um cluster”](#). Para usar esse exemplo, substitua os *user input placeholders* por suas próprias informações.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Resource": "arn:aws:s3-outposts:us-
west-2:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/access-point-name,
```

```
        "Action": [  
            "s3-outposts:*"  
        ]  
    }  
]  
}
```

Criação e configuração de um cluster

Para criar um cluster que execute o Spark com o S3 no Outposts, conclua as etapas a seguir no console.

Como criar um cluster que execute o Spark com o S3 no Outposts

1. Abra o console do Amazon EMR em <https://console.aws.amazon.com/elasticmapreduce/>.
2. No painel de navegação à esquerda, escolha Clusters.
3. Selecione Criar cluster.
4. Em Versão do Amazon EMR, escolha emr-7.0.0 ou posterior.
5. Em Pacote de aplicativos, escolha Spark interativo. Selecione todas as outras aplicações compatíveis que você queira incluir no cluster.
6. Para habilitar o Amazon S3 no Outposts, insira suas configurações.

Exemplo de configuração

Para usar o exemplo de configuração a seguir, substitua *user input placeholders* por suas próprias informações.

```
[  
  {  
    "Classification": "core-site",  
    "Properties": {  
      "fs.s3a.bucket.DOC-EXAMPLE-BUCKET.accesspoint.arn": "arn:aws:s3-outposts:us-  
west-2:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/access-point-name"  
      "fs.s3a.committer.name": "magic",  
      "fs.s3a.select.enabled": "false"  
    }  
  },  
  {
```

```
"Classification": "hadoop-env",
"Configurations": [
  {
    "Classification": "export",
    "Properties": {
      "JAVA_HOME": "/usr/lib/jvm/java-11-amazon-corretto.x86_64"
    }
  }
],
"Properties": {}
},
{
  "Classification": "spark-env",
  "Configurations": [
    {
      "Classification": "export",
      "Properties": {
        "JAVA_HOME": "/usr/lib/jvm/java-11-amazon-corretto.x86_64"
      }
    }
  ],
  "Properties": {}
},
{
  "Classification": "spark-defaults",
  "Properties": {
    "spark.executorEnv.JAVA_HOME": "/usr/lib/jvm/java-11-amazon-corretto.x86_64",
    "spark.sql.sources.fastS3PartitionDiscovery.enabled": "false"
  }
}
]
```

7. Na seção Redes, escolha uma nuvem privada virtual (VPC) e uma sub-rede que estejam no seu rack do AWS Outposts. Para obter mais informações sobre o Amazon EMR no Outposts, consulte [Clusters do EMR no AWS Outposts](#) no Guia de gerenciamento do Amazon EMR.
8. Na seção Perfil de instância do EC2 para o Amazon EMR, escolha o perfil do IAM que tem a [política de permissões que você criou anteriormente](#) anexada.
9. Defina as configurações restantes do cluster e escolha Criar cluster.

Visão geral das configurações

As tabelas a seguir descrevem as configurações do S3A e do Spark e os valores a serem especificados para os respectivos parâmetros ao configurar um cluster que usa o S3 no Outposts com o Amazon EMR.

Configurações do S3A

Parâmetro	Valor padrão	Valor necessário para o S3 no Outposts	Explicação
<code>fs.s3a.aws.credentials.provider</code>	Se não for especificado, o S3A procurará o S3 no bucket de regiões com o nome de bucket Outposts.	O ARN do ponto de acesso do bucket do S3 no Outposts.	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.
<code>fs.s3a.committer.name</code>	<code>file</code>	<code>magic</code>	O confirmador Magic é o único confirmador compatível com o S3 no Outposts.
<code>fs.s3a.select.enabled</code>	<code>TRUE</code>	<code>FALSE</code>	O S3 Select não é compatível com o Outposts.
<code>JAVA_HOME</code>	<code>/usr/lib/jvm/java-8</code>	<code>/usr/lib/jvm/java-11-amazon-corretto.x86_64</code>	O S3 no Outposts em S3A requer Java versão 11.

Configurações do Spark

Parâmetro	Valor padrão	Valor necessário para o S3 no Outposts	Explicação
<code>spark.sql.sources.fastS3PartitionDiscovery.enabled</code>	TRUE	FALSE	O S3 no Outposts não oferece suporte à partição rápida.
<code>spark.executorEnv.JAVA_HOME</code>	<code>/usr/lib/jvm/java-8</code>	<code>/usr/lib/jvm/java-11-amazon-corretto.x86_64</code>	O S3 no Outposts em S3A requer Java versão 11.

Considerações

Considere os seguintes pontos ao integrar o Amazon EMR a buckets do S3 no Outposts:

- O Amazon S3 no Outposts é compatível com o Amazon EMR 7.0.0 e posterior.
- O conector S3A é necessário para usar o S3 no Outposts com o Amazon EMR. Somente o S3A tem os recursos necessários para interagir com buckets do S3 no Outposts. Para obter informações sobre a configuração do conector S3A, consulte [Prerequisites](#).
- O Amazon S3 no Outposts só oferece suporte à criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) com o Amazon EMR. Para ter mais informações, consulte [the section called “Criptografia de dados”](#).
- O Amazon S3 no Outposts não oferece suporte a gravações com o `FileOutputCommitter` do S3A. As gravações com o `FileOutputCommitter` do S3A em buckets do S3 no Outposts resultam no seguinte erro: `InvalidStorageClass`: a classe de armazenamento especificada não é válida.
- O Amazon S3 no Outposts não é compatível com o Amazon EMR Sem Servidor nem com o Amazon EMR no EKS.
- Os logs do Amazon EMR são armazenados em um local regional do Amazon S3 selecionado por você e não são armazenados localmente no bucket do S3 no Outposts.

Armazenamento em cache de autorização e autenticação

O S3 no Outposts armazena em cache os dados de autenticação e autorização com segurança e localmente nos racks do Outposts. O cache remove idas e voltas à Região da AWS pai para cada solicitação. Isso elimina a variabilidade introduzida pelas idas e voltas da rede. Com o cache de autenticação e autorização no S3 no Outposts, você tem latências consistentes que são independentes da latência da conexão entre o Outposts e a Região da AWS.

Quando você faz uma solicitação de API do S3 no Outposts, os dados de autenticação e autorização são armazenados em cache com segurança. Os dados em cache são então usados para autenticar as solicitações subseqüentes da API de objetos do S3. O S3 no Outposts só armazena em cache os dados de autenticação e autorização quando a solicitação é assinada usando o Signature versão 4A (SigV4A). O cache é armazenado localmente no Outposts dentro do serviço S3 no Outposts. Ele é atualizado de forma assíncrona quando você faz uma solicitação de API do S3. O cache é criptografado e nenhuma chave criptográfica de texto simples é armazenada no Outposts.

O cache é válido por até 10 minutos quando o Outpost está conectado à Região da AWS. Quando você faz uma solicitação de API do S3 no Outposts, ele é atualizado de forma assíncrona para garantir que as políticas mais recentes sejam usadas. Se o Outpost for desconectado da Região da AWS, o cache será válido por até 12 horas.

Configurar o cache de autorização e autenticação

O S3 no Outposts armazena automaticamente em cache os dados de autenticação e autorização para solicitações assinadas com o algoritmo SigV4A. Consulte mais informações em [Assinar solicitações de API da AWS](#) no Guia do usuário do AWS Identity and Access Management. O algoritmo SigV4A está disponível nas versões mais recentes dos SDKs da AWS. É possível obtê-lo por meio de uma dependência em [AWS Common Runtime \(CRT\) libraries](#).

Você precisa usar a versão mais recente do SDK da AWS e instalar a versão mais recente do CRT. Por exemplo, você pode executar `pip install awscrt` para obter a versão mais recente do CRT com o Boto3.

O S3 no Outposts não armazena em cache os dados de autenticação e autorização para solicitações assinadas com o algoritmo SigV4.

Validar a assinatura do SigV4A

Você pode usar o AWS CloudTrail para validar se as solicitações foram assinadas com o SigV4A. Consulte mais informações sobre a configuração do CloudTrail para o S3 no Outposts em [Monitoramento do S3 no Outposts com logs do AWS CloudTrail](#).

Depois de configurar o CloudTrail, é possível verificar como uma solicitação foi assinada no campo `SignatureVersion` dos logs do CloudTrail. As solicitações que foram assinadas com SigV4A terão um `SignatureVersion` definido como `AWS4-ECDSA-P256-SHA256`. As solicitações que foram assinadas com o SigV4 terão `SignatureVersion` definido como `AWS4-HMAC-SHA256`.

Segurança no S3 on Outposts

A segurança para com a 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](#)
- [AWS PrivateLink para S3 on Outposts](#)

- [Chaves de política específicas de autenticação do AWS Signature Version 4 \(SigV4\)](#)
- [Políticas gerenciadas da AWS para o Amazon S3 no Outposts](#)
- [Usar perfis vinculados a serviço para o Amazon S3 no Outposts](#)

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 [Usar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 \(SSE-S3\)](#).

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 [Como usar criptografia do lado do servidor com chaves fornecidas pelo cliente \(SSE-C\)](#).

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

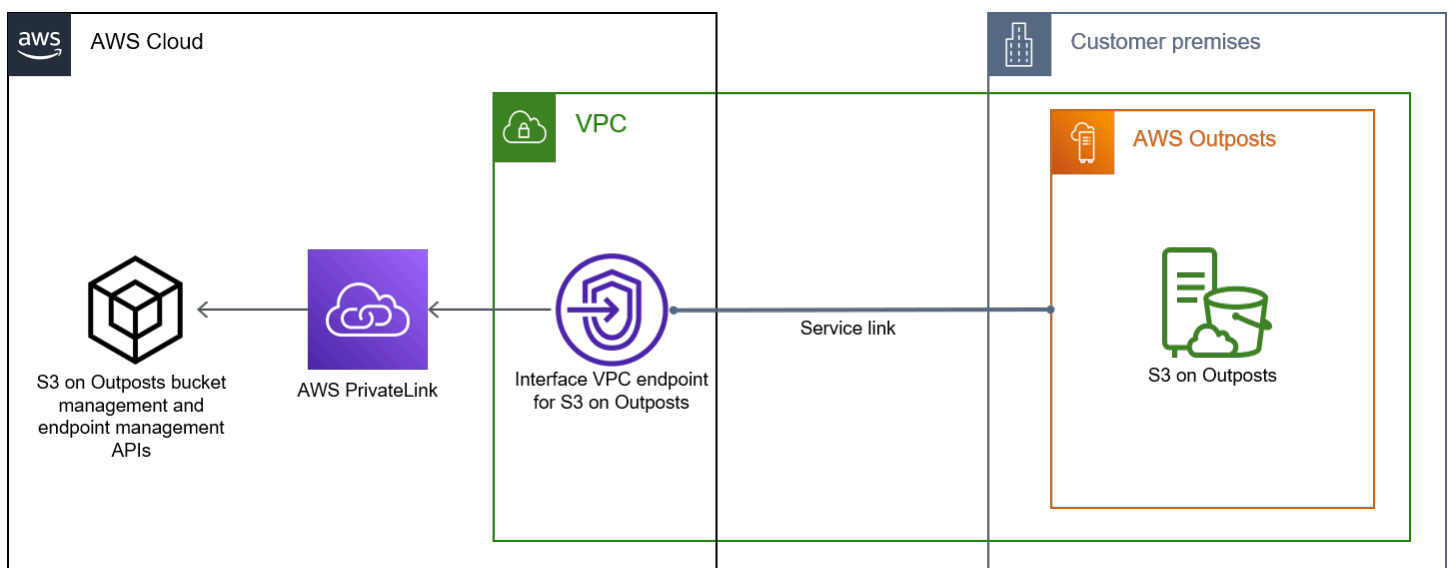
O S3 no Outposts é compatível com o AWS PrivateLink, que fornece acesso direto de gerenciamento ao armazenamento do S3 no Outposts por meio de um endpoint privado na rede privada virtual. Isso permite que você simplifique a arquitetura de rede interna e realize operações de gerenciamento no armazenamento de objetos do Outposts usando endereços IP privados na nuvem privada virtual (VPC). Usar o AWS PrivateLink elimina a necessidade de usar endereços IP públicos ou servidores proxy.

Com o AWS PrivateLink para o Amazon S3 no Outposts, você pode provisionar endpoints da VPC de interface na nuvem privada virtual (VPC) para acessar as APIs de [gerenciamento de bucket](#) e [gerenciamento de endpoint](#) do S3 no Outposts. Os endpoints da VPC de interface são diretamente acessíveis por aplicações implantadas na VPC ou no ambiente on-premises por meio da rede privada virtual (VPN) ou do AWS Direct Connect. Você pode acessar as APIs de gerenciamento de buckets e endpoints por meio do AWS PrivateLink. O AWS PrivateLink não é

compatível com operações de [transferência de dados](#) da API, como GET, PUT e APIs semelhantes. Essas operações já são transferidas de forma privada por meio da configuração do ponto de acesso e do endpoint do S3 no Outposts. Para ter mais informações, consulte [Redes para S3 on Outposts](#).

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 AWS. 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 AWS e por meio do 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](#)
- [Acessar endpoints da interface do S3 on Outposts](#)
- [Atualizar uma configuração de DNS on-premises](#)
- [Criar um endpoint da VPC para o S3 on Outposts](#)

- [Criar políticas de bucket e políticas de endpoint da VPC para S3 on Outposts](#)

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](#); por exemplo, GET, PUT e operações de API de objetos semelhantes.
- DNS privado

Acessar endpoints da interface do S3 on Outposts

Para acessar APIs de gerenciamento de buckets e endpoints do S3 on Outposts usando o AWS PrivateLink, você deve atualizar as 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.

⚠ Important

Os endpoints da interface do S3 no Outposts são resolvidos pelo domínio DNS público. O S3 no Outposts não é compatível com DNS privado. Use o parâmetro `--endpoint-url` para todas as APIs de gerenciamento de buckets e endpoints.

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'
```

```
)
```

Para obter mais informações, consulte [AWS PrivateLink para Amazon S3](#) no Guia do desenvolvedor do Boto3.

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()
```

Para obter mais informações, consulte [S3ControlClient](#) na Referência de APIs do AWS SDK for Java.

Atualizar uma configuração de DNS on-premises

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 resolvidor 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 bloqueie 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](#)

- [Exemplo: negar acesso usando um endpoint da VPC específico em uma política de bucket do S3 on Outposts](#)

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](#). 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 Authorization seja usado na autenticação de solicitação.</p> <p>Valores válidos:</p> <p>REST-HEADER</p>

Chaves aplicáveis	Descrição
s3-outposts:signatureAge	<p data-bbox="467 212 792 243">REST-QUERY-STRING</p> <p data-bbox="467 296 1442 373">A duração em milissegundos da validade de uma assinatura em uma solicitação autenticada.</p> <p data-bbox="467 422 1312 453">Essa condição funciona somente para URLs pré-assinados.</p> <p data-bbox="467 506 1507 772">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 data-bbox="467 821 847 852">Valor de exemplo: 600000</p>

Chaves aplicáveis	Descrição
<p>s3-outposts:x-amz-content-sha256</p>	<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, define 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-PAYLOAD</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-assinadas. 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"
        }
      }
    }
  ]
}

```

Políticas gerenciadas da AWS para o Amazon S3 no Outposts

Uma política gerenciada pela AWS é uma política independente criada e administrada pela AWS. As políticas gerenciadas pela AWS são criadas para fornecer permissões a vários casos de uso comuns a fim de que você possa começar a atribuir permissões a usuários, grupos e perfis.

Lembre-se de que as políticas gerenciadas pela AWS podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque estão disponíveis para todos os clientes da

AWS usarem. Recomendamos que você reduza ainda mais as permissões definindo [políticas gerenciadas pelo cliente](#) específicas para seus casos de uso.

Você não pode alterar as permissões definidas em políticas gerenciadas pela AWS. Se a AWS atualiza as permissões definidas em um política gerenciada pela AWS, a atualização afeta todas as identidades de entidades principais (usuários, grupos e perfis) às quais a política está vinculada. É mais provável que a AWS atualize uma política gerenciada pela AWS quando um novo AWS service (Serviço da AWS) é lançado ou novas operações de API são disponibilizadas para os serviços existentes.

Para obter mais informações, consulte [Políticas gerenciadas pela AWS](#) no Manual do usuário do IAM.

Política gerenciada da AWS: AWSS3OnOutpostsServiceRolePolicy

Ajuda a gerenciar recursos de rede para você como parte do perfil vinculado a serviço `AWSServiceRoleForS3OnOutposts`.

Para visualizar as permissões dessa política, consulte [AWSS3OnOutpostsServiceRolePolicy](#).

Atualizações do S3 no Outposts em políticas gerenciadas da AWS

Visualize detalhes sobre atualizações em políticas gerenciadas da AWS do S3 no Outposts desde o momento em que esse serviço começou a monitorar essas alterações.

Alteração	Descrição	Data
O S3 no Outposts adicionou <code>AWSS3OnOutpostsServiceRolePolicy</code>	O S3 no Outposts adicionou <code>AWSS3OnOutpostsServiceRolePolicy</code> como parte do perfil vinculado a serviço <code>AWSServiceRoleForS3OnOutposts</code> , que ajuda a gerenciar os recursos de rede para você.	3 de outubro de 2023
O S3 no Outposts começou a monitorar alterações	O S3 no Outposts começou a monitorar alterações nas políticas gerenciadas da AWS.	3 de outubro de 2023

Usar perfis vinculados a serviço para o Amazon S3 no Outposts

O Amazon S3 no Outposts usa [perfis vinculados a serviço](#) do AWS Identity and Access Management (IAM). O perfil vinculado a serviço é um tipo exclusivo de perfil do IAM vinculado diretamente ao S3 no Outposts. Os perfis vinculados a serviço são predefinidos pelo S3 no Outposts e incluem todas as permissões que o serviço requer para chamar outros serviços da AWS em seu nome.

Um perfil vinculado a serviço facilita a configuração do S3 no Outposts porque você não precisa adicionar as permissões necessárias manualmente. O S3 no Outposts define as permissões dos perfis vinculados a serviço. Além disso, exceto se definido de outra forma, somente o S3 no Outposts pode assumir os perfis. 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.

Uma função vinculada ao serviço poderá ser excluída somente após excluir seus recursos relacionados. Isso protege os recursos do S3 no Outposts, pois não é possível remover por engano a permissão para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com perfis vinculados aos serviços, consulte [Serviços da AWS que funcionam com o IAM](#) e procure os serviços que apresentam Sim na coluna Funções vinculadas aos serviços. Escolha um Sim com um link para visualizar a documentação da função vinculada a esse serviço.

Permissões de perfil vinculado a serviço para o S3 no Outposts

O S3 no Outposts usa o perfil vinculado a serviço denominado `AWSServiceRoleForS3OnOutposts` para ajudar a gerenciar recursos de rede para você.

A função vinculada ao serviço `AWSServiceRoleForS3OnOutposts` confia nos seguintes serviços para assumir a função:

- `s3-outposts.amazonaws.com`

A política de permissões do perfil denominada `AWSS3OnOutpostsServiceRolePolicy` permite que o S3 no Outposts conclua as seguintes ações nos recursos especificados:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
```

```

        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcs",
        "ec2:DescribeCoipPools",
        "ec2:GetCoipPoolUsage",
        "ec2:DescribeAddresses",
        "ec2:DescribeLocalGatewayRouteTableVpcAssociations"
    ],
    "Resource": "*",
    "Sid": "DescribeVpcResources"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface"
    ],
    "Resource": [
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:security-group/*"
    ],
    "Sid": "CreateNetworkInterface"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface"
    ],
    "Resource": [
        "arn:aws:ec2:*:*:network-interface/*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/CreatedBy": "S3 On Outposts"
        }
    },
    "Sid": "CreateTagsForCreateNetworkInterface"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:AllocateAddress"
    ],
    "Resource": [

```

```

        "arn:aws:ec2:*:*:ipv4pool-ec2/*"
    ],
    "Sid": "AllocateIpAddress"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:AllocateAddress"
    ],
    "Resource": [
        "arn:aws:ec2:*:*:elastic-ip/*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/CreatedBy": "S3 On Outposts"
        }
    },
    "Sid": "CreateTagsForAllocateIpAddress"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:CreateNetworkInterfacePermission",
        "ec2>DeleteNetworkInterface",
        "ec2>DeleteNetworkInterfacePermission",
        "ec2:DisassociateAddress",
        "ec2:ReleaseAddress",
        "ec2:AssociateAddress"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/CreatedBy": "S3 On Outposts"
        }
    },
    "Sid": "ReleaseVpcResources"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "*",

```

```
    "Condition": {
      "StringEquals": {
        "ec2:CreateAction": [
          "CreateNetworkInterface",
          "AllocateAddress"
        ],
        "aws:RequestTag/CreatedBy": [
          "S3 On Outposts"
        ]
      }
    },
    "Sid": "CreateTags"
  }
]
}
```

É necessário configurar permissões para que uma entidade do IAM (por exemplo, um perfil) crie, edite ou exclua um perfil vinculado a serviço. Para ter mais informações, consulte [Service-linked role permissions](#) (Permissões de nível vinculado a serviços) no Guia do usuário do IAM.

Criar um perfil vinculado a serviço para o S3 no Outposts

Não é necessário criar manualmente uma função vinculada ao serviço. Quando você cria um endpoint do S3 no Outposts no AWS Management Console, a AWS CLI ou a API da AWS, o S3 no Outposts cria o perfil vinculado a serviço para você.

Se excluir essa função vinculada ao serviço e precisar criá-la novamente, você poderá usar esse mesmo processo para recriar a função em sua conta. Quando você cria um endpoint do S3 no Outposts, o S3 no Outposts cria o perfil vinculado a serviço para você novamente.

Também é possível usar o console do IAM para criar um perfil vinculado a serviço com o caso de uso do S3 no Outposts. Na AWS CLI ou na API do AWS, crie uma função vinculada ao serviço com o nome de serviço `s3-outposts.amazonaws.com`. Para obter mais informações, consulte [Criar uma função vinculada ao serviço](#) no Manual do usuário do IAM. Se você excluir essa função vinculada ao serviço, será possível usar esse mesmo processo para criar a função novamente.

Editar um perfil vinculado a serviço para o S3 no Outposts

O S3 no Outposts não permite editar o perfil vinculado a serviço `AWSServiceRoleForS3OnOutposts`. Isso inclui o nome do perfil, pois várias entidades podem

fazer referência a ele. No entanto, será possível editar a descrição da função usando o IAM. Para ter mais informações, consulte [Editar uma função vinculada a serviço](#) no Guia do usuário do IAM.

Excluir um perfil vinculado a serviço para o S3 no Outposts

Se você não precisar mais usar um recurso ou serviço que requer uma função vinculada a serviço, é recomendável excluí-la. 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 excluí-la manualmente.

Note

Se o serviço do S3 no Outposts estiver usando o perfil quando você tentar excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Como excluir recursos do S3 no Outposts usados pelo perfil `AWSServiceRoleForS3OnOutposts`

1. [Exclua os endpoints do S3 no Outposts](#) na Conta da AWS em todas as Regiões da AWS.
2. Exclua o perfil vinculado a serviço usando o IAM.

Use o console do IAM, a AWS CLI ou a API da AWS para excluir a função vinculada ao serviço `AWSServiceRoleForS3OnOutposts`. Para obter mais informações, consulte [Excluir uma função vinculada ao serviço](#) no Guia do usuário do IAM.

Regiões compatíveis com perfis vinculados a serviço do S3 no Outposts

O S3 no Outposts aceita o uso de perfis vinculados a serviço em todas as Regiões da AWS em que o serviço está disponível. Para obter mais informações, consulte [Regiões e endpoints do S3 no Outposts](#).

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

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?](#)

Para obter mais informações sobre como gerenciar e compartilhar sua capacidade de armazenamento do Amazon S3 on Outposts, consulte os tópicos a seguir.

Tópicos

- [Gerenciar o versionamento do S3 para um bucket do S3 no Outposts](#)
- [Criar e gerenciar uma configuração de ciclo de vida para um bucket do Amazon S3 on Outposts](#)
- [Replicar objetos para o S3 no Outposts](#)
- [Compartilhar o S3 on Outposts usando o AWS RAM](#)
- [Outros Serviços da AWS que usam o S3 on Outposts](#)

Gerenciar o versionamento do S3 para um bucket do S3 no Outposts

Quando habilitado, o versionamento do S3 salva várias cópias distintas de um objeto no mesmo bucket. O versionamento do S3 pode ser usado para preservar, recuperar e restaurar todas as versões de cada objeto armazenado em buckets do Outposts. O versionamento do S3 ajuda você a se recuperar de ações não intencionais de usuários e de falhas da aplicação.

Os buckets do Amazon S3 no Outposts têm três estados de versionamento:

- **Unversioned (Sem versionamento):** se você nunca habilitou ou suspendeu o versionamento do S3 em seu bucket, ele não tem versionamento e não retorna nenhum status de versionamento do S3. Para obter mais informações sobre o S3 Versioning, consulte [Usando o versionamento em buckets do S3](#).
- **Enabled (Habilitado):** habilita o versionamento do S3 para os objetos no bucket. Todos os objetos adicionados ao bucket recebem um ID de versão exclusivo. 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 [PutObject](#), os novos objetos receberão um ID de versão exclusivo.

- **Suspended (Suspensão):** suspende o versionamento do S3 para os objetos no bucket. Todos os objetos adicionados ao bucket depois que o versionamento é suspenso recebem o ID de versão null. Para mais informações, consulte [Adicionar objetos a buckets com versionamento suspenso](#).

Depois que você habilita o versionamento do S3 para um bucket do S3 no Outposts, ele nunca pode voltar a um estado sem versionamento. No entanto, você pode suspender o versionamento. Para obter mais informações sobre o S3 Versioning, consulte [Usando o versionamento em buckets do S3](#).

Para cada objeto em um bucket, há uma versão atual e zero ou mais versões desatualizadas. Para reduzir os custos de armazenamento, você pode configurar regras de ciclo de vida do bucket do S3 para expirar as versões desatualizadas após um período especificado. Para mais informações, consulte [Criar e gerenciar uma configuração de ciclo de vida para um bucket do Amazon S3 on Outposts](#).

Os exemplos a seguir mostram como habilitar ou suspender o versionamento para um bucket do S3 no Outposts usando o AWS Management Console e a AWS Command Line Interface (AWS CLI). Para criar um bucket com versionamento do S3 habilitado, consulte [Criar um bucket do S3 on Outposts](#).

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 mais informações, consulte [Especificações do Amazon S3 no Outposts](#).

Uso do console do S3

Como editar as configurações de versionamento do S3 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 Outposts buckets (Buckets do Outposts).
3. Escolha o bucket do Outposts para o qual você deseja habilitar o versionamento do S3.
4. Escolha a guia Properties (Propriedades).

5. Em Bucket Versioning (Versionamento de bucket), escolha Edit (Editar).
6. Edite as configurações de versionamento do S3 para o bucket escolhendo uma das seguintes opções:
 - Para suspender o versionamento do S3 e interromper a criação de versões de objetos, escolha Suspend (Suspender).
 - Para habilitar o versionamento do S3 e salvar várias cópias distintas de cada objeto, escolha Enable (Habilitar).
7. Selecione Save changes.

Uso do AWS CLI

Para habilitar ou suspender o versionamento do S3 para um bucket usando a AWS CLI, use o comando `put-bucket-versioning` conforme mostrado nos exemplos a seguir. Para usar esses exemplos, substitua cada *user input placeholder* por suas próprias informações.

Para obter mais informações, consulte [put-bucket-versioning](#) na Referência da AWS CLI.

Example : Habilitar o versionamento do S3

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket --versioning-configuration Status=Enabled
```


Example : Suspender o versionamento do S3

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket --versioning-configuration Status=Suspended
```

Criar e gerenciar uma configuração de ciclo de vida para um bucket do Amazon S3 on Outposts

Você pode usar o Ciclo de Vida do S3 para otimizar a capacidade de armazenamento do Amazon S3 no Outposts. Você pode criar regras de ciclo de vida para expirar objetos à medida que envelhecem ou quando são substituídos por versões mais recentes. Você pode criar, habilitar, desabilitar e excluir uma regra de ciclo de vida.

Para obter mais informações sobre o ciclo de vida do S3, consulte [Gerenciando seu ciclo de vida de armazenamento](#).

 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](#)
- [Criar e gerenciar uma configuração de ciclo de vida usando a AWS CLI e o SDK para Java](#)

Criar e gerenciar uma regra de ciclo de vida usando o AWS Management Console

Você pode usar o Ciclo de Vida do S3 para otimizar a capacidade de armazenamento do Amazon S3 no Outposts. Você pode criar regras de ciclo de vida para expirar objetos à medida que envelhecem ou quando são substituídos por versões mais recentes. Você pode criar, habilitar, desabilitar e excluir uma regra de ciclo de vida.

Para obter mais informações sobre o ciclo de vida do S3, consulte [Gerenciando seu ciclo de vida de armazenamento](#).

 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 S3 no Outposts usando o AWS Management Console, consulte os tópicos a seguir.

Tópicos

- [Criar uma regra de ciclo de vida](#)
- [Habilitar uma regra de ciclo de vida](#)


- [Editar uma regra de ciclo de vida](#)
- [Excluir uma regra de ciclo de vida](#)

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. Insira um valor para Lifecycle rule name (Nome da regra do ciclo de vida).
6. Em Rule scope (Escopo da regra), escolha uma das seguintes opções:
 - Para limitar o escopo a filtros específicos, escolha Limit the scope of this rule using one or more filters (Limitar o escopo desta regra usando um ou mais filtros). Depois, adicione um filtro de prefixos, etiquetas ou tamanho de objeto.
 - Para aplicar a regra a todos os objetos no bucket, escolha Apply to all objects in the bucket (Aplicar a todos os objetos no bucket).
7. Em Lifecycle rule actions (Ações de regra de ciclo de vida), escolha uma das seguintes opções:
 - Expire current versions of objects (Expirar versões atuais de objetos): para buckets com versionamento habilitado, o S3 no Outposts adiciona um marcador de exclusão e retém os objetos como versões não atuais. Para buckets que não usam o versionamento do S3, o S3 no Outposts exclui permanentemente os objetos.
 - Permanently delete noncurrent versions of objects (Excluir permanentemente versões de objetos desatualizadas): o S3 no Outposts exclui permanentemente as versões de objetos desatualizadas.
 - Delete expired object delete markers or incomplete multipart uploads (Excluir marcadores de exclusão de objetos expirados ou multipart uploads incompletos): o S3 no Outposts exclui permanentemente marcadores de exclusão de objetos expirados ou multipart uploads incompletos.

Se você limitar o escopo da sua regra de ciclo de vida usando etiquetas de objeto, não poderá escolher a opção Delete expired object delete markers (Excluir marcadores de exclusão

de objetos expirados). Também não poderá escolher a opção Delete expired object delete markers (Excluir marcadores de exclusão de objetos expirados) se escolher a opção Expire current object versions (Expirar as versões atuais do objeto).

 Note

Filtros baseados em tamanho não podem ser usados com marcadores de exclusão e multipart uploads incompletos.

8. Se você escolheu a opção Expire current versions of objects (Expirar versões atuais de objetos) ou Permanently delete noncurrent versions of objects (Excluir permanentemente versões não atuais de objetos), configure o gatilho de regra com base em uma data específica ou com base na idade do objeto.
9. Se você escolheu a opção Delete expired object delete markers (Excluir marcadores de exclusão de objetos expirados), para confirmar que deseja excluir os marcadores de exclusão de objetos expirados, selecione a opção Delete expired object delete markers (Excluir marcadores de exclusão de objetos expirados).
10. Em Timeline Summary (Resumo da linha do tempo), revise sua regra de ciclo de vida e escolha Create rule (Criar regra).

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, em Lifecycle rule (Regra de ciclo de vida), selecione a regra que 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. (Opcional) Atualize o valor de Lifecycle rule name (Nome da regra de ciclo de vida).
6. Em Rule scope (Escopo da regra), edite o escopo conforme necessário:
 - Para limitar o escopo a filtros específicos, escolha Limit the scope of this rule using one or more filters (Limitar o escopo desta regra usando um ou mais filtros). Depois, adicione um filtro de prefixos, etiquetas ou tamanho de objeto.
 - Para aplicar a regra a todos os objetos no bucket, escolha Apply to all objects in the bucket (Aplicar a todos os objetos no bucket).
7. Em Lifecycle rule actions (Ações de regra de ciclo de vida), escolha uma das seguintes opções:
 - Expire current versions of objects (Expirar versões atuais de objetos): para buckets com versionamento habilitado, o S3 no Outposts adiciona um marcador de exclusão e retém os objetos como versões não atuais. Para buckets que não usam o versionamento do S3, o S3 no Outposts exclui permanentemente os objetos.
 - Permanently delete noncurrent versions of objects (Excluir permanentemente versões de objetos desatualizadas): o S3 no Outposts exclui permanentemente as versões de objetos desatualizadas.
 - Delete expired object delete markers or incomplete multipart uploads (Excluir marcadores de exclusão de objetos expirados ou multipart uploads incompletos): o S3 no Outposts exclui permanentemente marcadores de exclusão de objetos expirados ou multipart uploads incompletos.

Se você limitar o escopo da sua regra de ciclo de vida usando etiquetas de objeto, não poderá escolher a opção Delete expired object delete markers (Excluir marcadores de exclusão de objetos expirados). Também não poderá escolher a opção Delete expired object delete markers (Excluir marcadores de exclusão de objetos expirados) se escolher a opção Expire current object versions (Expirar as versões atuais do objeto).

Note

Filtros baseados em tamanho não podem ser usados com marcadores de exclusão e multipart uploads incompletos.

8. Se você escolheu a opção *Expire current versions of objects* (Expirar versões atuais de objetos) ou *Permanently delete noncurrent versions of objects* (Excluir permanentemente versões não atuais de objetos), configure o gatilho de regra com base em uma data específica ou com base na idade do objeto.
9. Se você escolheu a opção *Delete expired object delete markers* (Excluir marcadores de exclusão de objetos expirados), para confirmar que deseja excluir os marcadores de exclusão de objetos expirados, selecione a opção *Delete expired object delete markers* (Excluir marcadores de exclusão de objetos expirados).
10. Escolha *Save* (Salvar).

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, em *Lifecycle rule* (Regra de ciclo de vida), selecione a regra que deseja excluir.
5. Escolha *Delete* (Excluir).

Criar e gerenciar uma configuração de ciclo de vida usando a AWS CLI e o SDK para Java

Você pode usar o Ciclo de Vida do S3 para otimizar a capacidade de armazenamento do Amazon S3 no Outposts. Você pode criar regras de ciclo de vida para expirar objetos à medida que envelhecem ou quando são substituídos por versões mais recentes. Você pode criar, habilitar, desabilitar e excluir uma regra de ciclo de vida.

Para obter mais informações sobre o ciclo de vida do S3, consulte [Gerenciando seu ciclo de vida de armazenamento](#).

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](#)
- [Executar GET na configuração de ciclo de vida em um bucket do S3 on Outposts](#)

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"
            }
          ]
        }
      }
    }
  ]
}
```

```

        }
    ],
    "ObjectSizeGreaterThan": 1000,
    "ObjectSizeLessThan": 5000
  }
},
"Status": "Enabled",
"Expiration": {
  "Days": 10
}
}
]
}

```

- 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-outposts-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("mytagkey1");
    S3Tag tag2 = new S3Tag().withKey("mytagkey2").withValue("mytagkey2");

    LifecycleRuleFilter lifecycleRuleFilter = new LifecycleRuleFilter()
        .withAnd(new LifecycleRuleAndOperator()

```

```
        .withPrefix("myprefix")
        .withTags(tag1, tag2))
        .withObjectSizeGreaterThan(1000)
        .withObjectSizeLessThan(5000);

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-outposts-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());
}
```

Replicar objetos para o S3 no Outposts

Com a replicação do S3 no AWS Outposts, você pode configurar o Amazon S3 no Outposts para replicar automaticamente objetos do S3 em diferentes Outposts ou entre buckets no mesmo Outpost. Você pode usar a replicação do S3 no Outposts para manter várias réplicas dos dados no mesmo ou em diferentes Outposts, ou em contas diferentes, para ajudar a atender às necessidades de residência de dados. A replicação do S3 no Outposts ajuda a atender às suas necessidades de armazenamento compatíveis e ao compartilhamento de dados entre contas. Se você precisa garantir que as réplicas sejam idênticas aos dados de origem, poderá usar a replicação do S3 no Outposts para fazer réplicas dos objetos que retêm todos os metadados, como o tempo de criação do objeto original, as tags e os IDs de versão.

A replicação do S3 no Outposts também fornece métricas e notificações detalhadas para monitorar o status da replicação de objetos entre buckets. Você pode usar o Amazon CloudWatch para monitorar o andamento da replicação rastreando bytes pendentes de replicação, operações pendentes de replicação e latência de replicação entre os buckets de origem e destino. Para diagnosticar e corrigir rapidamente os problemas de configuração, você também pode configurar o Amazon EventBridge para receber notificações sobre falhas de objetos de replicação. Para saber mais, consulte [Gerenciar sua replicação](#).

Tópicos

- [Configuração de replicação](#)
- [Requisitos para a replicação do S3 no Outposts](#)
- [O que é replicado?](#)
- [O que não é replicado?](#)
- [O que não é compatível com a replicação do S3 no Outposts?](#)
- [Configuração da replicação](#)
- [Gerenciar sua replicação](#)

Configuração de replicação

O S3 no Outposts 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 S3 no Outposts não pode replicar objetos sem sua permissão. Você concede permissões do S3 no Outposts com o perfil do IAM especificada na configuração de replicação. O S3 no Outposts assume esse perfil do IAM para replicar objetos em seu nome. Você precisa conceder as permissões

necessárias ao perfil do IAM antes de iniciar a replicação. Para obter mais informações sobre essas permissões para Outposts S3 no Outposts, consulte [Criar um perfil do IAM](#).

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/`.

Para obter mais informações sobre a configuração de replicação e as regras de replicação do S3 no Outposts, consulte [ReplicationConfiguration](#) na Referência de API do Amazon Simple Storage Service.

Requisitos para a replicação do S3 no Outposts

A replicação exige o seguinte:

- O intervalo CIDR do Outpost de destino deve estar associado à tabela de sub-rede do Outpost de origem. Para obter mais informações, consulte [Pré-requisitos para criar regras de replicação](#).
- Tanto o bucket de origem quanto o de destino devem ter o versionamento do S3 ativado. Para obter mais informações sobre versionamento, consulte [Gerenciar o versionamento do S3 para um bucket do S3 no Outposts](#).
- O Amazon S3 no Outposts deve ter permissão para replicar objetos do bucket de origem para o bucket de destino em seu nome. Isso significa que você deve criar um perfil de serviço para delegar as permissões GET e PUT ao S3 no Outposts.
 1. Antes de criar o perfil de serviço, você deve ter a permissão GET no bucket de origem e a permissão PUT no bucket de destino.
 2. Para criar o perfil de serviço para delegar permissões ao S3 no Outposts, você deve primeiro configurar as permissões para permitir que uma entidade do IAM (um usuário ou perfil) execute as ações `iam:CreateRole` e `iam:PassRole`. Depois, permita que a entidade do IAM crie

um perfil de serviço. Para fazer com que o S3 no Outposts assuma o perfil de serviço em seu nome e delegue as permissões GET e PUT ao S3 no Outposts, você deve atribuir as políticas de permissões e confiança necessárias ao perfil. Para obter mais informações sobre essas permissões para Outposts S3 no Outposts, consulte [Criar um perfil do IAM](#). Para obter mais informações sobre como criar um perfil de serviço, consulte [Criar um perfil de serviço](#).

O que é replicado?

Por padrão, o S3 no Outposts replica o seguinte:

- Objetos criados depois de adicionar uma configuração de replicação.
- 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 [Status da replicação se a sincronização de modificação de réplica do Amazon S3 no Outposts estiver ativada](#).
- Tags de objeto, se houver.

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 S3 no Outposts adicionará um marcador de exclusão. O S3 no Outposts lida com o marcador de exclusão da seguinte forma:
 - O S3 no Outposts não replica 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 sobre como habilitar a replicação do marcador de exclusão na configuração de replicação, consulte [Uso do console do S3](#).
- Se você especificar um ID de versão do objeto para excluir na solicitação DELETE, o S3 no Outposts excluirá permanentemente essa versão do objeto no bucket de origem. No entanto, ele não replica a exclusão nos buckets de destino. Em outras palavras: ele não exclui a mesma versão do objeto dos buckets de destino. Esse comportamento protege os dados contra exclusões mal-intencionadas.

O que não é replicado?

Por padrão, o S3 no Outposts 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.
- 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 S3 no Outposts não replicará os objetos novamente.
- Objetos criados com criptografia do lado do servidor com as chaves de criptografia fornecidas pelo cliente (SSE-C).
- 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 você habilitar uma configuração de ciclo de vida somente no bucket de origem e configurar ações de expiração, o S3 no Outposts criará marcadores de exclusão para objetos expirados no bucket de origem, mas não replicará esses marcadores aos buckets de destino. 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](#).

O que não é compatível com a replicação do S3 no Outposts?

Os recursos de replicação do S3 a seguir não são compatíveis com o S3 on Outposts:

- Controle de Tempo de Replicação do S3 (S3 RTC). O S3 RTC não é compatível porque o tráfego de objetos na replicação do S3 no Outposts passa pela rede on-premises (o gateway local). Para ter mais informações sobre gateways locais, consulte [Trabalhar com o gateway local](#) no Guia do usuário do AWS Outposts.
- Replicação do S3 para operações em lote.

Configuração da replicação

Note

Objetos que já existiam em seu bucket antes de você configurar uma regra de replicação não são replicados automaticamente. Em outras palavras, o Amazon S3 no Outposts não replica os objetos retroativamente. Para replicar objetos que foram criados antes da configuração de replicação, você pode usar a operação da API `CopyObject` para copiá-los no mesmo bucket. Depois que os objetos são copiados, eles aparecem como objetos “novos” no bucket e a configuração de replicação será aplicada a eles. Para ter mais informações sobre como copiar um objeto, consulte [Copiar um objeto em um bucket do Amazon S3 no Outposts usando o AWS SDK for Java](#) e [CopyObject](#) na Referência da API do Amazon Simple Storage Service.

Para habilitar a replicação do S3 em Outposts, adicione uma regra de replicação ao seu bucket de origem do Outposts. A regra de replicação diz ao S3 no Outposts para replicar objetos, conforme especificado. Na regra de replicação, você deve fornecer o seguinte:

- O ponto de acesso do bucket do Outposts de origem: o nome de recurso da Amazon (ARN) do ponto de acesso ou o alias do ponto de acesso do bucket do qual você deseja que o S3 no Outposts replique os objetos. Para ter mais informações sobre como usar aliases de ponto de acesso, consulte [Usar um alias em estilo de bucket para um ponto de acesso de bucket do S3 no Outposts](#).
- Os objetos que você deseja replicar: você pode replicar todos os objetos do bucket do Outposts de origem ou um subgrupo. Identifique um subgrupo fornecendo, na configuração, um [prefixo do nome da chave](#), uma ou mais tags de objeto ou ambos.

Por exemplo, se você configurar uma regra de replicação para replicar somente objetos com o prefixo de nome da chave `Tax/`, o S3 no Outposts replicará objetos com chaves como `Tax/doc1` ou `Tax/doc2`. Porém, ele não replica objetos com a chave `Legal/doc3`. Se você especificar um prefixo e uma ou mais tags, o S3 no Outposts replicará somente os objetos com o prefixo das chaves e as tags específicas.

- O bucket do Outposts de destino: o ARN ou o alias do ponto de acesso do bucket para o qual você deseja que o S3 no Outposts replique os objetos.

Você pode configurar a regra de replicação usando a API REST, os AWS SDKs, a AWS Command Line Interface (AWS CLI) ou o console do Amazon S3.

O S3 no Outposts também fornece operações de API para 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:

- [PutBucketReplication](#)
- [GetBucketReplication](#)
- [DeleteBucketReplication](#)

Tópicos

- [Pré-requisitos para criar regras de replicação](#)
- [Criar regras de replicação no Outposts](#)

Pré-requisitos para criar regras de replicação

Tópicos

- [Conectar suas sub-redes do Outpost de origem e destino](#)
- [Criar um perfil do IAM](#)

Conectar suas sub-redes do Outpost de origem e destino

Para que seu tráfego de replicação vá do Outpost de origem para o Outpost de destino pelo gateway local, você deve adicionar uma nova rota para configurar a rede. É necessário conectar os intervalos de rede de Encaminhamento Entre Domínios Sem Classificação (CIDR) de seus pontos de acesso. Para cada par de pontos de acesso, você precisa configurar essa conexão apenas uma vez.

Algumas etapas para configurar a conexão são diferentes, dependendo do tipo de acesso de seus endpoints do Outposts associados aos seus pontos de acesso. O tipo de acesso para endpoints é privado (roteamento direto de nuvem virtual privada [VPC] para AWS Outposts) ou IP de propriedade do cliente (um grupo de endereços IP de propriedade do cliente [grupo de CoIP] em sua rede on-premises).

Etapa 1: Encontrar o intervalo CIDR de seu endpoint do Outposts de origem

Como encontrar o intervalo CIDR de seu endpoint de origem que está associado ao seu ponto de acesso de origem

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. Na lista Buckets do Outposts, selecione o bucket de origem desejado para replicação.
4. Selecione a guia Pontos de acesso do Outposts e o ponto de acesso do Outposts do bucket de origem para sua regra de replicação.
5. Selecione endpoint do Outposts.
6. Copie o ID da sub-rede para uso na [Etapa 5](#).
7. O método utilizado para encontrar o intervalo CIDR do endpoint do Outposts de origem depende do tipo de acesso de seu endpoint.

Na seção Visão geral do endpoint do Outposts, consulte o Tipo de acesso.

- Se o tipo de acesso for Privado, copie o valor do Encaminhamento Entre Domínios sem Classificação (CIDR) para ser utilizado na [Etapa 6](#).
- Se o tipo de acesso for IP de propriedade do cliente, faça o seguinte:
 1. Copie o valor do Grupo IPv4 de propriedade do cliente para ser utilizado como ID do grupo de endereços posteriormente.
 2. Abra o console do AWS Outposts em <https://console.aws.amazon.com/outposts/>.
 3. No painel de navegação, selecione Tabelas de rotas do gateway local.
 4. Selecione o valor de ID da tabela de rotas do gateway local de seu Outpost de origem.
 5. No painel de detalhes, selecione a guia Grupos de CoIP. Cole o valor de seu ID do grupo de CoIP que você copiou anteriormente na caixa de pesquisa.
 6. Para o grupo de CoIP correspondente, copie o valor CIDRs correspondente de seu endpoint do Outposts de origem para uso na [Etapa 6](#).

Etapa 2: Encontrar o ID da sub-rede e o intervalo CIDR de seu endpoint do Outposts de destino

Para encontrar o ID da sub-rede e o intervalo CIDR de seu endpoint de destino associado ao seu ponto de acesso de destino, siga as mesmas subetapas indicadas na [Etapa 1](#) e altere seu endpoint

do Outposts de origem para o endpoint do Outposts de destino ao realizar essas subetapas. Copie o valor do ID da sub-rede de seu endpoint do Outposts de destino para uso na [Etapa 6](#). Copie o valor CIDR de seu endpoint do Outposts de destino para uso na [Etapa 5](#).

Etapa 3: Encontrar o ID do gateway local do seu Outpost de origem

Como encontrar o ID do gateway local de seu Outpost de origem

1. Abra o console do AWS Outposts em <https://console.aws.amazon.com/outposts/>.
2. No painel de navegação à esquerda, selecione Gateways locais.
3. Na página Gateways locais, encontre o ID do Outpost de seu Outpost de origem a ser utilizado para replicação.
4. Copie o valor do ID do gateway local de seu Outpost de origem para uso na [Etapa 5](#).

Para ter mais informações sobre o gateway local, consulte [Gateway local](#) no Guia do usuário do AWS Outposts.

Etapa 4: Encontrar o ID do gateway local de seu Outpost de destino

Para encontrar o ID do gateway local de seu Outpost de destino, siga as mesmas subetapas indicadas na [Etapa 3](#), exceto para procurar o ID de seu Outpost de destino. Copie o valor do ID do gateway local de seu Outpost de destino para uso na [Etapa 6](#).

Etapa 5: Configurar a conexão da sub-rede do Outpost de origem com a sub-rede do Outpost de destino

Como estabelecer conexão da sub-rede do Outpost de origem com a sub-rede do Outpost de destino

1. Faça login no AWS Management Console e abra o console da VPC em <https://console.aws.amazon.com/vpc/>.
2. No painel de navegação à esquerda, escolha Subnets (Sub-redes).
3. Na caixa de pesquisa, insira o ID da sub-rede de seu endpoint do Outposts de origem que você encontrou na [Etapa 1](#). Selecione a sub-rede com o ID de sub-rede correspondente.
4. Para o item de sub-rede correspondente, selecione o valor da tabela de rotas dessa sub-rede.
5. Na página com uma tabela de rotas selecionada, selecione Ações e, depois, Editar rotas.
6. Na página Editar rotas, selecione Editar rota.
7. Em Destino, insira o intervalo CIDR de seu endpoint do Outposts de destino que você encontrou na [Etapa 2](#).

8. Em Destino, selecione Gateway local do Outpost e insira o ID do gateway local de seu Outpost de origem que você encontrou na [Etapa 3](#).
9. Escolha Save changes (Salvar alterações).
10. O status da rota deve ser ativo.

Etapa 6: Configurar a conexão da sub-rede do Outpost de destino com a sub-rede do Outpost de origem

1. Faça login no AWS Management Console e abra o console da VPC em <https://console.aws.amazon.com/vpc/>.
2. No painel de navegação à esquerda, escolha Subnets (Sub-redes).
3. Na caixa de pesquisa, insira o ID da sub-rede de seu endpoint do Outposts de destino que você encontrou na [Etapa 2](#). Selecione a sub-rede com o ID de sub-rede correspondente.
4. Para o item de sub-rede correspondente, selecione o valor da tabela de rotas dessa sub-rede.
5. Na página com uma tabela de rotas selecionada, selecione Ações e, depois, Editar rotas.
6. Na página Editar rotas, selecione Editar rota.
7. Em Destino, insira o intervalo CIDR de seu endpoint do Outposts de origem que você encontrou na [Etapa 1](#).
8. Em Destino, selecione Gateway local do Outpost e insira o ID do gateway local de seu Outpost de destino que você encontrou na [Etapa 4](#).
9. Escolha Save changes (Salvar alterações).
10. O status da rota deve ser ativo.

Depois de conectar os intervalos de rede CIDR de seus pontos de acesso de origem e destino, você deve criar um perfil do AWS Identity and Access Management (IAM).

Criar um perfil do IAM

Por padrão, todos os recursos do S3 no Outposts: buckets, objetos e sub-recursos relacionados, são privados, e somente o proprietário do recurso pode acessá-lo. O S3 no Outposts precisa de permissões de leitura e replicação de objetos a partir do bucket do Outposts de origem. Você concede essas permissões criando um perfil do IAM perfil de serviço e especificando esse perfil 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 [Criar regras de replicação no Outposts](#). Para obter mais informações sobre funções do IAM, consulte [Funções do IAM](#) no Manual do usuário do IAM.

- O exemplo a seguir mostra uma política de confiança na qual você identifica o S3 no Outposts como a entidade principal do serviço que pode assumir a função.

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

- 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 S3 no Outposts assumir o perfil, ele terá as permissões que você especificar nessa política. Para utilizar essa política, substitua os *user input placeholders* por suas próprias informações. Substitua-os pelos IDs dos Outposts de origem e destino e pelos nomes dos bucket e nomes de pontos de acesso de seus buckets dos Outposts de origem e destino.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3-outposts:GetObjectVersionForReplication",
        "s3-outposts:GetObjectVersionTagging"
      ],
      "Resource":[
        "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/
        bucket/SOURCE-OUTPOSTS-BUCKET/object/*",

```

```

        "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/
accesspoint/SOURCE-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3-outposts:ReplicateObject",
      "s3-outposts:ReplicateDelete"
    ],
    "Resource": [
      "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/
bucket/DESTINATION-OUTPOSTS-BUCKET/object/*",
      "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/
accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
    ]
  }
]
}

```

A política de acesso concede permissões às seguintes ações:

- `s3-outposts:GetObjectVersionForReplication`: a permissão para essa ação é concedida em todos os objetos para permitir que o S3 no Outposts obtenha uma versão específica associada a cada objeto.
- `s3-outposts:GetObjectVersionTagging`: a permissão para essa ação em objetos no bucket do `SOURCE-OUTPOSTS-BUCKET` (o bucket de origem) permite que o S3 no Outposts leia tags de objeto para replicação. Para obter mais informações, consulte [Adicionar etiquetas aos buckets do S3 on Outposts](#). Se o S3 no Outposts não tiver essa permissão, ele replicará os objetos, mas não as tags de objeto.
- `s3-outposts:ReplicateObject` e `s3-outposts:ReplicateDelete`: as permissões para essas ações em todos os objetos no bucket do `DESTINATION-OUTPOSTS-BUCKET` (o bucket de destino) permitem que o S3 no Outposts replique objetos ou marcadores de exclusão no bucket do Outposts de destino. Para obter mais informações sobre marcadores de exclusão, consulte [Como a exclusão de operações afeta a replicação](#).

Note

- A permissão para a ação `s3-outposts:ReplicateObject` no bucket do `DESTINATION-OUTPOSTS-BUCKET` (o bucket de destino) também permite a

replicação de tags de objeto. Portanto, você não precisa conceder permissão explícita para a ação `s3-outposts:ReplicateTags`.

- Para replicação entre contas, o proprietário do bucket do Outposts de destino deve atualizar sua política de bucket para conceder permissão para a ação `s3-outposts:ReplicateObject` no ***DESTINATION-OUTPOSTS-BUCKET***. A ação `s3-outposts:ReplicateObject` permite que o S3 no Outposts replique objetos e tags de objetos no bucket do Outposts de destino.

Para obter uma lista das ações do S3 no Outposts, consulte [Ações definidas pelo S3 no Outposts](#).

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 do Outposts de origem contenha objetos pertencentes a outra Conta da AWS. O proprietário dos objetos deverá conceder as permissões exigidas à Conta da AWS proprietária do perfil do IAM por meio da política do bucket e da política de ponto de acesso. Caso contrário, o S3 no Outposts não conseguirá acessar os objetos e ocorrerá uma falha na replicação dos objetos.

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 S3 no Outposts.

Conceder permissões quando os buckets do Outposts de origem e destino pertencerem a Contas da AWS diferentes

Quando os buckets do Outposts de origem e destino não pertencem às mesmas contas, o proprietário do bucket do Outposts de destino deve atualizar as políticas de bucket e ponto de acesso do bucket de destino. Essas políticas devem conceder ao proprietário do bucket do Outposts de origem e ao perfil de serviço do IAM permissões para realizar ações de replicação, conforme mostrado nos exemplos de políticas a seguir, ou ocorrerá uma falha na replicação. Nestes exemplos de política, ***DESTINATION-OUTPOSTS-BUCKET*** é o bucket de destino. Para usar esses exemplos de política, substitua os *user input placeholders* por suas próprias informações.

Se você estiver criando o perfil de serviço do IAM manualmente, defina o caminho do perfil como `role/service-role/`, conforme mostrado nos exemplos de políticas a seguir. Para obter mais informações, consulte [ARNs do IAM](#) no Guia do usuário do IAM.

```
{
  "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-outposts:ReplicateDelete",
        "s3-outposts:ReplicateObject"
      ],
      "Resource": [
        "arn:aws:s3-outposts:region:DestinationBucket-account-ID:outpost/DESTINATION-OUTPOST-ID/bucket/DESTINATION-OUTPOSTS-BUCKET/object/*"
      ]
    }
  ]
}
```

```
{
  "Version": "2012-10-17",
  "Id": "PolicyForDestinationAccessPoint",
  "Statement": [
    {
      "Sid": "Permissions on objects",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::SourceBucket-account-ID:role/service-role/source-account-IAM-role"
      },
      "Action": [
        "s3-outposts:ReplicateDelete",
        "s3-outposts:ReplicateObject"
      ],
    }
  ]
}
```

```
    "Resource" : [
      "arn:aws:s3-outposts:region:DestinationBucket-account-
      ID:outpost/DESTINATION-OUTPOST-ID/accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT/
      object/*"
    ]
  }
]
```

Note

Se os objetos no bucket do Outposts de origem estiverem marcados, observe o seguinte: Se o proprietário do bucket do Outposts de origem conceder ao S3 no Outposts permissão para as ações `s3-outposts:GetObjectVersionTagging` e `s3-outposts:ReplicateTags` para replicação de tags de objeto (pelo perfil 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](#).

Criar regras de replicação no Outposts

A replicação do S3 no Outposts é a replicação assíncrona e automática de objetos em buckets no mesmo AWS Outposts ou em outro. A replicação copia os objetos recém-criados e as atualizações de objeto de um bucket do Outposts de origem para um bucket do Outposts de destino. Para obter mais informações, consulte [Replicar objetos para o S3 no Outposts](#).

Note

Objetos que já existiam no bucket do Outposts de origem antes de você definir regras de replicação não são replicados. Em outras palavras, o S3 no Outposts não replica os objetos retroativamente. Para replicar objetos que foram criados antes da configuração de replicação, você pode usar a operação da API `CopyObject` para copiá-los no mesmo bucket. Depois que os objetos são copiados, eles aparecem como objetos “novos” no bucket e a configuração de replicação será aplicada a eles. Para ter mais informações sobre como copiar um objeto, consulte [Copiar um objeto em um bucket do Amazon S3 no Outposts usando o AWS SDK for Java](#) e [CopyObject](#) na Referência da API do Amazon Simple Storage Service.

Ao configurar a replicação, você adiciona regras de replicação ao bucket do Outposts de origem. As regras de replicação definem quais objetos do bucket do Outposts de origem devem ser replicados e o bucket do Outposts 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 do Outposts de destino pode estar no mesmo Outpost que o bucket do Outposts de origem ou pode estar em outro Outpost.

Para as regras de replicação do S3 no Outposts, você deve fornecer o nome do recurso da Amazon (ARN) do ponto de acesso do bucket do Outposts de origem e o ARN do ponto de acesso do bucket do Outposts de destino em vez dos nomes dos buckets do Outposts de origem e destino.

Se você especificar um ID da versão do objeto a ser excluído, o S3 no Outposts excluirá essa versão do objeto no bucket do Outposts de origem. No entanto, ele não replica a exclusão no bucket do Outposts de destino. Em outras palavras, ele não exclui a mesma versão do objeto do bucket do Outposts de destino. Esse comportamento protege os dados contra exclusões mal-intencionadas.

Quando você adiciona uma regra de replicação a um bucket do Outposts, ela fica ativada por padrão para que ela comece a funcionar assim que é salva.

Neste exemplo, você configura a replicação para os buckets de origem e destino que estão em Outposts diferentes e 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. Para ter informações sobre permissões de replicação do S3 no Outposts entre contas, consulte [Conceder permissões quando os buckets do Outposts de origem e destino pertencerem a Contas da AWS diferentes](#).

Para saber os pré-requisitos para as regras de replicação do S3 no Outposts, consulte [Pré-requisitos para criar regras de replicação](#).

Uso do console do S3

Siga estas etapas para configurar uma regra de replicação quando o bucket do Amazon S3 no Outposts de destino estiver em um Outpost diferente do bucket do Outposts de origem.

Se o bucket do Outposts de destino estiver em uma conta diferente do bucket do Outposts de origem, você deverá adicionar uma política ao bucket do Outposts de destino. Assim, será possível conceder ao proprietário da conta do bucket do Outposts de origem permissão para replicar objetos no bucket do Outposts de destino. Para obter mais informações, consulte [Conceder permissões quando os buckets de origem e de destino pertencerem a Contas da AWS diferentes](#).

Como criar uma regra 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 do Outposts, selecione o nome do bucket a ser utilizado como o bucket de origem.
3. Selecione a guia Gerenciamento, role para baixo até a seção Regras de replicação e, depois, selecione Criar regra de replicação.
4. Em Nome da regra de replicação, insira um nome para a regra, ajudando a identificá-la posteriormente. O nome é obrigatório e precisa ser exclusivo dentro do bucket.
5. Em Status, Habilitado é selecionado por padrão. Uma regra ativada começa a funcionar assim que você a salva. Se você quiser ativar a regra posteriormente, selecione Desabilitado.
6. Em Prioridade, o valor da prioridade da regra determina qual regra aplicar se houver regras sobrepostas. Quando objetos são incluídos no escopo de mais de uma regra de replicação, o S3 no Outposts utiliza esses valores de prioridade para evitar conflitos. Por padrão, novas regras são adicionadas à configuração de replicação com a prioridade mais alta. Quanto maior o número, maior a prioridade.

Para alterar a prioridade da regra, depois de salvá-la, selecione o nome da regra na lista de regras de replicação, selecione Ações e, depois, Editar prioridade.


7. Em Bucket de origem, você tem as seguintes opções para definir a origem da replicação:
 - Para replicar todo o bucket, selecione Aplicar a todos os objetos no bucket.
 - Para aplicar a filtragem de prefixo ou tag à fonte de replicação, selecione Limitar o escopo dessa regra utilizando um ou mais filtros. Você pode combinar um prefixo e tags.
 - Para replicar todos os objetos com o mesmo prefixo, em Prefixo, insira um prefixo na caixa. O uso do filtro Prefixo limita a replicação para todos os objetos que têm nomes que começam com a mesma string (por exemplo, pictures).

Se você inserir um prefixo que seja o nome de uma pasta, use uma / (barra) como o último caractere (por exemplo, pictures/).

- Para replicar todos os objetos que têm uma ou mais tags de objeto, selecione Adicionar tag) e insira o par de chave-valor nas caixas. Para adicionar outra etiqueta, repita o procedimento. Para obter mais informações sobre etiquetas de objeto, consulte [Adicionar etiquetas aos buckets do S3 on Outposts](#).

8. Para acessar seu bucket de origem do S3 no Outposts para replicação, em Nome do ponto de acesso de origem, selecione um ponto de acesso que esteja conectado ao bucket de origem.
9. Em Destino, selecione o ARN do ponto de acesso do bucket do Outposts de destino em que você deseja que o S3 no Outposts replique objetos. O bucket do Outposts de destino pode estar na mesma Conta da AWS que o bucket do Outposts de origem ou em outra.

Se o bucket de destino estiver em uma conta diferente do bucket do Outposts de origem, você deverá adicionar uma política ao bucket do Outposts de destino. Assim, será possível conceder ao proprietário da conta do bucket do Outposts de origem permissão para replicar objetos no bucket do Outposts de destino. Para obter mais informações, consulte [Conceder permissões quando os buckets do Outposts de origem e destino pertencerem a Contas da AWS diferentes](#).

 Note

Se o versionamento não estiver ativado no bucket do Outposts de destino, você receberá um aviso que contém um botão Habilitar versionamento. Escolha esse botão para habilitar o versionamento no bucket.

10. Configure um perfil de serviço do AWS Identity and Access Management (IAM) que o S3 no Outposts possa assumir para replicar objetos em seu nome.

Para configurar um perfil do IAM, em Perfil do IAM, faça o seguinte:

- Para que o S3 no Outposts crie um perfil do IAM para sua configuração de replicação, selecione Escolher entre os perfis do IAM existentes e, depois, Criar perfil. Quando você salva a regra, uma nova política é gerada para o perfil do IAM que coincide com os buckets do Outposts de origem e destino que você escolheu. Recomendamos que você selecione Criar perfil.
- Você também pode optar por utilizar um perfil do IAM existente. Se fizer isso, escolha um perfil que conceda ao S3 no Outposts as permissões necessárias para a replicação. Se esse perfil não conceder ao S3 no Outposts permissões suficientes para seguir sua regra de replicação, ocorrerá uma falha na replicação.

Para selecionar um perfil existente, selecione Escolher entre perfis do IAM existentes e, depois, o perfil no menu suspenso. Você também pode escolher Inserir um ARN de perfil do IAM e, depois, inserir o nome do recurso da Amazon (ARN) do perfil do IAM.

⚠ Important

Quando você adiciona uma regra de replicação a um bucket do S3 no Outposts, é preciso ter as permissões `iam:CreateRole` e `iam:PassRole` para poder criar e transmitir o perfil do IAM que concede permissões de replicação do S3 no Outposts. Para ter mais informações, consulte [Conceder permissões ao usuário para transmitir um perfil a um AWS service \(Serviço da AWS\)](#) no Guia do usuário do IAM.

11. Todos os objetos nos buckets do Outposts são criptografados por padrão. Para ter mais informações sobre o S3 no Outposts, consulte [Criptografia de dados no S3 on Outposts](#). Somente objetos que são criptografados com a criptografia no lado do servidor com chaves gerenciadas do Amazon S3 (SSE-S3) podem ser replicados. A replicação de objetos que são criptografados com a criptografia no lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS) ou a criptografia do lado do servidor com chaves de criptografia fornecidas pelo cliente (SSE-C) não é compatível.
12. Conforme necessário, ative as seguintes opções adicionais ao definir a configuração da regra de replicação:
 - Se você quiser habilitar métricas de replicação do S3 no Outposts na configuração de replicação, selecione Métricas de replicação. Para obter mais informações, consulte [Monitorar o andamento com métricas de replicação](#).
 - 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 [Como a exclusão de operações afeta a replicação](#).
 - Se você quiser replicar as alterações de metadados feitas nas réplicas de volta aos objetos de origem, selecione Sincronização de modificação de réplica. Para obter mais informações, consulte [Status da replicação se a sincronização de modificação de réplica do Amazon S3 no Outposts estiver ativada](#).
13. Para finalizar, selecione Criar regra.

Depois de salvar sua regra, você pode editar, habilitar, desabilitar ou excluir sua regra. Para fazer isso, acesse a guia Gerenciamento do bucket do Outposts de origem, role para baixo até a seção Regras de replicação, selecione sua regra e Editar regra.

Usar a AWS CLI

Para usar a AWS CLI para configurar a replicação quando os buckets do Outposts de origem e destino pertencem à mesma Conta da AWS, faça o seguinte:

- Crie os buckets do Outposts de origem e destino.
- Habilite o versionamento nos dois buckets.
- Crie um perfil do IAM que conceda ao S3 permissão para replicar objetos
- Adicione a configuração de replicação ao bucket do Outposts de origem.

Para verificar sua configuração, teste-a.

Como configurar a replicação quando os buckets do Outposts 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 ter mais informações sobre a definição de perfis de 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 o perfil de serviço do IAM que o S3 no Outposts pode assumir. Você só poderá fazer isso se o perfil utilizado tiver as permissões `iam:CreateRole` e `iam:PassRole`. Para ter mais informações, consulte [Conceder permissões ao usuário para transmitir um perfil a um AWS service \(Serviço da AWS\)](#) no Guia do usuário do IAM. Se você usar credenciais de administrador para criar um perfil nomeado, este terá a permissão necessária para realizar todas as tarefas.

2. Crie um bucket de *origem* e habilite o versionamento nele. O comando `create-bucket` a seguir cria um bucket do `SOURCE-OUTPOSTS-BUCKET` na região Leste dos EUA (Norte da Virgínia) (`us-east-1`). Para usar esse comando, substitua *user input placeholders* por suas informações.

```
aws s3control create-bucket --bucket SOURCE-OUTPOSTS-BUCKET --outpost-id SOURCE-OUTPOST-ID --profile acctA --region us-east-1
```

O comando `put-bucket-versioning` a seguir habilita o versionamento no bucket do *SOURCE-OUTPOSTS-BUCKET*. Para usar esse comando, substitua *user input placeholders* por suas informações.

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/bucket/SOURCE-OUTPOSTS-BUCKET --versioning-configuration Status=Enabled --profile acctA
```

3. Crie um bucket de *destino* e habilite o versionamento nele. O comando `create-bucket` a seguir cria um bucket do *DESTINATION-OUTPOSTS-BUCKET* na região Oeste dos EUA (Oregon) (`us-west-2`). Para usar esse comando, substitua *user input placeholders* por suas informações.

Note

Para definir uma configuração da replicação quando os buckets do Outposts de origem e destino estiverem na mesma Conta da AWS, use o mesmo perfil nomeado. 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.

```
aws s3control create-bucket --bucket DESTINATION-OUTPOSTS-BUCKET --create-bucket-configuration LocationConstraint=us-west-2 --outpost-id DESTINATION-OUTPOST-ID --profile acctA --region us-west-2
```

O comando `put-bucket-versioning` a seguir habilita o versionamento no bucket do *DESTINATION-OUTPOSTS-BUCKET*. Para usar esse comando, substitua *user input placeholders* por suas informações.

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/bucket/DESTINATION-OUTPOSTS-BUCKET --versioning-configuration Status=Enabled --profile acctA
```

4. Crie um perfil de serviço do IAM. Posteriormente na configuração da replicação, você vai adicionar esse perfil de serviço ao bucket do *SOURCE-OUTPOSTS-BUCKET*. O S3 no Outposts assume esse perfil para replicar objetos em seu nome. A função do IAM é criada em duas etapas:

- a. Crie um perfil do IAM.
 - i. Copie a política de confiança a seguir e salve-a em um arquivo chamado `s3-on-outposts-role-trust-policy.json` no diretório atual do computador local. Essa política concede à entidade principal do serviço do S3 no Outposts permissões para assumir o perfil de serviço.

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

- ii. Execute o comando da a seguir para criar a função. Substitua os *user input placeholders* por suas próprias informações.

```
aws iam create-role --role-name replicationRole --assume-role-policy-document file://s3-on-outposts-role-trust-policy.json --profile acctA
```

- b. Anexe uma política de permissões ao perfil de serviço.
 - i. Copie a política de permissões a seguir e salve-a em um arquivo com o nome `s3-on-outposts-role-permissions-policy.json` no diretório atual do computador local. Essa política concede permissões para várias ações de bucket e objetos do S3 no Outposts. Para utilizar essa política, substitua os *user input placeholders* por suas próprias informações.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3-outposts:GetObjectVersionForReplication",

```

```

        "s3-outposts:GetObjectVersionTagging"
    ],
    "Resource": [
        "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-
        OUTPOST-ID/bucket/SOURCE-OUTPOSTS-BUCKET/object/*",
        "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-
        OUTPOST-ID/accesspoint/SOURCE-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3-outposts:ReplicateObject",
      "s3-outposts:ReplicateDelete"
    ],
    "Resource": [
        "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-
        OUTPOST-ID/bucket/DESTINATION-OUTPOSTS-BUCKET/object/*",
        "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-
        OUTPOST-ID/accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
    ]
  }
]
}

```

- ii. Execute o comando a seguir para criar uma política e ligá-la à função. Substitua os *user input placeholders* por suas próprias informações.

```

aws iam put-role-policy --role-name replicationRole --policy-
document file://s3-on-outposts-role-permissions-policy.json --policy-
name replicationRolePolicy --profile acctA

```

5. Adicione uma configuração de replicação ao bucket *SOURCE-OUTPOSTS-BUCKET*.
 - a. Embora a API do S3 no Outposts exija uma configuração de replicação no formato XML, a AWS CLI requer que você especifique a configuração da replicação no formato JSON. Salve o JSON a seguir em um arquivo chamado `replication.json` no diretório local do seu computador. Para usar essa configuração, substitua os *user input placeholders* por suas próprias informações.

```

{
  "Role": "IAM-role-ARN",

```

```

"Rules": [
  {
    "Status": "Enabled",
    "Priority": 1,
    "DeleteMarkerReplication": { "Status": "Disabled" },
    "Filter" : { "Prefix": "Tax"},
    "Destination": {
      "Bucket":
        "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-
ID/accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT"
    }
  }
]
}

```

- b. Execute o comando `put-bucket-replication` a seguir para adicionar a configuração de replicação ao seu bucket do Outposts de origem. Para usar esse comando, substitua *user input placeholders* por suas informações.

```

aws s3control put-bucket-replication --account-id 123456789012 --
bucket arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-
ID/bucket/SOURCE-OUTPOSTS-BUCKET --replication-configuration file://
replication.json --profile acctA

```

- c. Para recuperar a configuração de replicação, use o comando `get-bucket-replication`. Para usar esse comando, substitua *user input placeholders* por suas informações.

```

aws s3control get-bucket-replication --account-id 123456789012 --bucket
arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/
bucket/SOURCE-OUTPOSTS-BUCKET --profile acctA

```

6. Teste a configuração no console do Amazon S3:

- Faça login no AWS Management Console e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
- No bucket de *SOURCE-OUTPOSTS-BUCKET*, crie uma pasta denominada Tax.
- Adicione exemplos de objetos à pasta Tax no bucket de *SOURCE-OUTPOSTS-BUCKET*.
- No bucket de *DESTINATION-OUTPOSTS-BUCKET*, verifique o seguinte:
 - O S3 no Outposts replicou os objetos.

Note

O tempo que o S3 no Outposts leva para 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](#).

- Na guia Propriedades, o Status de replicação está definido como Réplica (identificando-a como um objeto de réplica).

Gerenciar sua replicação

Esta seção descreve opções adicionais de configuração de replicação disponíveis no S3 no Outposts, como determinar o status da replicação e como solucionar problemas de replicação. Para obter informações sobre a configuração de replicação principal, consulte [Configuração da replicação](#).

Tópicos

- [Monitorar o andamento com métricas de replicação](#)
- [Obtenção de informações sobre o status da replicação](#)
- [Solução de problemas de replicação](#)
- [Usar o EventBridge para replicação do S3 no Outposts](#)

Monitorar o andamento com métricas de replicação

A replicação do S3 no Outposts fornece métricas detalhadas para as regras de replicação na configuração de replicação. Com métricas de replicação, você pode monitorar em intervalos de cinco minutos o andamento da replicação rastreando bytes pendentes de replicação, latência de replicação e operações pendentes. Para auxiliar na solução de problemas de configuração, você também pode configurar o Amazon EventBridge para receber notificações sobre falhas de replicação.

Quando métricas de replicação são ativadas, a replicação do S3 no Outposts publica as seguintes métricas no Amazon CloudWatch:

- Bytes pendentes de replicação: o número total de bytes de objetos com replicação pendente para determinada regra de replicação.
- Latência de replicação: o número máximo de segundos pelo qual o bucket de destino da replicação está atrás do bucket de origem para determinada regra de replicação.

- Operações pendentes de replicação: o número de operações pendentes de replicação para determinada regra de replicação. As operações incluem objetos, marcadores de exclusão e tags.

Note

As métricas de replicação do S3 no Outposts são cobradas usando a mesma taxa das métricas personalizadas do CloudWatch. Para obter mais informações, consulte [Preço do CloudWatch](#).

Obtenção de informações sobre o status da replicação

O status da replicação pode ajudar você a determinar o estado atual de um objeto que está sendo replicado pelo Amazon S3 no Outposts. O status de replicação de um objeto de origem retornará PENDINGCOMPLETED, ou FAILED. O status de replicação de uma réplica retornará REPLICATED.

Visão geral do status da replicação

Em um cenário de replicação, você tem um bucket de origem em que configura a replicação e um bucket de destino onde o S3 no Outposts replica objetos. Ao solicitar um objeto (usando `GetObject`) ou metadados de objeto (usando `HeadObject`) nesses buckets, o S3 no Outposts retornará o cabeçalho `x-amz-replication-status` na resposta da seguinte maneira:

- Ao solicitar um objeto no bucket de origem, o S3 no Outposts 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 S3 no Outposts 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 solicitações de objeto com esse prefixo de nome de chave, o S3 no Outposts retorna o cabeçalho `x-amz-replication-status` com um dos seguintes valores para o status de replicação do 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 o estado FAILED em caso de problemas como a ausência das

permissões da função de replicação ou do bucket. No caso de falhas temporárias, por exemplo, se um bucket ou uma 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 no Outposts retomará a replicação desses objetos.

- Ao solicitar um objeto no bucket de destino, se o objeto de sua solicitação for uma réplica criada pelo S3 no Outposts, o S3 no Outposts retornará o cabeçalho `x-amz-replication-status` com o 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.

Status da replicação se a sincronização de modificação de réplica do Amazon S3 no Outposts estiver ativada

Quando suas regras de replicação ativam a sincronização de modificação de réplica do S3, as réplicas podem informar um status diferente de REPLICIA. Se alterações de metadados estiverem no processo de replicação, o cabeçalho `x-amz-replication-status` da réplica retornará PENDING. Se a sincronização de modificação de réplica não replicar os metadados, o cabeçalho da réplica retornará FAILED. Se os metadados forem replicados corretamente, o cabeçalho da réplica retornará o valor REPLICIA.

Solução de problemas de replicação

Se as réplicas dos objetos não aparecerem no bucket do Amazon S3 no Outposts de destino depois de configurar a replicação, use estas dicas para identificar e corrigir os problemas.

- O tempo que o S3 no Outposts leva para replicar um objeto depende de vários fatores, incluindo a distância entre o Outposts de origem e destino e o tamanho do objeto.

Você pode conferir o status de replicação do objeto de origem. Se o status de replicação do objeto for PENDING, o S3 no Outposts 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 do recurso da Amazon (ARN) do ponto de acesso do bucket de destino está correto.

- O prefixo do nome de chave está correto. Por exemplo, se você definiu a configuração para replicar objetos com o prefixo Tax, apenas objetos com nomes de chaves como Tax/document1 ou Tax/document2 serão replicados. Um objeto com o nome de chave document3 não será replicado.
- O status é Enabled.
- Verifique se o versionamento não foi suspenso em nenhum bucket. Tanto o bucket de origem quanto o de destino devem ter o versionamento ativado.
- 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 [Conceder permissões quando os buckets do Outposts de origem e destino pertencerem a Contas da AWS diferentes](#).
- Se a réplica do objeto não aparecer no bucket de destino, os seguintes problemas podem ter impedido a replicação:
 - O S3 no Outposts não replica um objeto em um bucket de origem que seja uma réplica criada por outra configuração de replicação. Por exemplo, se você definir uma configuração de replicação do bucket A para o bucket B e para o bucket C, o S3 no Outposts não replicará réplicas de objetos no bucket B para o bucket C.

Se você quiser replicar objetos no bucket A para o bucket B e o bucket C, defina vários destinos de bucket em diferentes regras de replicação para sua configuração de replicação do bucket de origem. Por exemplo, crie duas regras de replicação no bucket A de origem, com uma regra para replicar no bucket B de destino e a outra regra para replicar no 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. Para evitar problemas de replicação, 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](#).
- Vamos supor que, na configuração da replicação, você adicione uma regra para replicar um subconjunto de objetos com uma tag específica. Nesse caso, você deve atribuir a chave da tag específica e o valor no momento de criar o objeto para o S3 no Outposts replicar o objeto. Se você primeiro criar um objeto e depois adicionar a tag ao objeto existente, o S3 no Outposts 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-outposts:GetObjectVersionForReplication",  
"s3-outposts:GetObjectVersionTagging"
```

Buckets de destino:

```
"s3-outposts:ReplicateObject",  
"s3-outposts:ReplicateDelete",  
"s3-outposts:ReplicateTags"
```

- O Amazon EventBridge pode notificar você quando os objetos não forem replicados em seus Outposts de destino. Para obter mais informações, consulte [Usar o EventBridge para replicação do S3 no Outposts](#).

Usar o EventBridge para replicação do S3 no Outposts

O Amazon S3 no Outposts é integrado ao Amazon EventBridge e usa o namespace `s3-outposts`. O EventBridge é um serviço de barramento de eventos com tecnologia sem servidor que você pode usar para conectar suas aplicações a dados de diversas origens. Para obter mais informações, consulte [O que é o Amazon EventBridge?](#) no Manual do usuário do Amazon EventBridge.

Para auxiliar na solução de problemas de configuração da replicação, você pode configurar o Amazon EventBridge para receber notificações sobre eventos de falha de replicação. O EventBridge pode notificar você em instâncias quando os objetos não forem replicados em seu Outposts de destino. Para ter mais informações sobre o estado atual de um objeto que está sendo replicado, consulte [Visão geral do status da replicação](#).

Sempre que determinados eventos acontecem no bucket do Outposts, o S3 no Outposts pode enviar eventos ao EventBridge. Ao contrário de outros destinos, não é necessário selecionar quais tipos de evento você deseja entregar. Você também pode usar regras do EventBridge para encaminhar eventos para outros destinos. Depois que o EventBridge é ativado, o S3 no Outposts envia todos os eventos a seguir ao EventBridge.

Tipo de evento	Descrição	Namespace
OperationFailedReplication	Ocorreu uma falha na replicação de um objeto em uma regra de replicação. Para ter mais informações sobre motivos de falha de replicação do S3 no Outposts, consulte Usar o EventBridge para visualizar os motivos de falha da replicação do S3 no Outposts .	s3-outposts

Usar o EventBridge para visualizar os motivos de falha da replicação do S3 no Outposts

A tabela a seguir relaciona os motivos de falha da replicação do S3 no Outposts. Você pode configurar uma regra do EventBridge para publicar e visualizar o motivo da falha por meio do Amazon Simple Queue Service (Amazon SQS), do Amazon Simple Notification Service (Amazon SNS), do AWS Lambda ou do Amazon CloudWatch Logs. Para ter mais informações sobre as permissões que são necessárias para usar esses recursos no EventBridge, consulte [Usar políticas com base em recursos para o EventBridge](#).

Motivos de falha da replicação	Descrição
AssumeRoleNotPermitted	O S3 no Outposts não pode assumir o perfil do AWS Identity and Access Management (IAM) especificado na configuração de replicação.
DstBucketNotFound	O S3 no Outposts não consegue encontrar o bucket de destino especificado na configuração de replicação.
DstBucketUnversioned	O versionamento não está ativado no bucket de destino do Outposts. Para ativar objetos com replicação do S3 no Outposts, você deve ativar o versionamento no bucket de destino.
DstDelObjNotPermitted	O S3 no Outposts não consegue replicar exclusões no bucket de destino. A permissão

Motivos de falha da replicação	Descrição
<code>DstMultipartCompleteNotPermitted</code>	<p><code>s3-outposts:ReplicateDelete</code> pode estar faltando para o bucket de destino.</p> <p>O S3 no Outposts não consegue fazer multipart upload de objetos no bucket de destino. A permissão <code>s3-outposts:ReplicateObject</code> pode estar faltando para o bucket de destino.</p>
<code>DstMultipartInitNotPermitted</code>	<p>O S3 no Outposts não consegue iniciar um multipart upload de objetos no bucket de destino. A permissão <code>s3-outposts:ReplicateObject</code> pode estar faltando para o bucket de destino.</p>
<code>DstMultipartPartUploadNotPermitted</code>	<p>O S3 no Outposts não consegue fazer upload de objetos multipart no bucket de destino. A permissão <code>s3-outposts:ReplicateObject</code> pode estar faltando para o bucket de destino.</p>
<code>DstOutOfCapacity</code>	<p>O S3 no Outposts não consegue se replicar para o Outpost de destino porque o Outposts está sem capacidade de armazenamento do S3.</p>
<code>DstPutObjNotPermitted</code>	<p>O S3 no Outposts não consegue replicar objetos no bucket de destino. A permissão <code>s3-outposts:ReplicateObject</code> pode estar faltando para o bucket de destino.</p>
<code>DstPutTaggingNotPermitted</code>	<p>O S3 no Outposts não consegue replicar tags de objetos no bucket de destino. A permissão <code>s3-outposts:ReplicateObject</code> pode estar faltando para o bucket de destino.</p>

Motivos de falha da replicação	Descrição
<code>DstVersionNotFound</code>	O S3 no Outposts não consegue encontrar a versão necessária do objeto no bucket de destino para replicar os metadados dessa versão do objeto.
<code>SrcBucketReplicationConfigMissing</code>	O S3 no Outposts não consegue encontrar uma configuração de replicação para o ponto de acesso associado ao bucket do Outposts de origem.
<code>SrcGetObjectNotPermitted</code>	O S3 no Outposts não consegue acessar o objeto no bucket de origem para replicação. A permissão <code>s3-outposts:GetObjectVersionForReplication</code> pode estar faltando para o bucket de origem.
<code>SrcGetTaggingNotPermitted</code>	O S3 no Outposts não consegue acessar as informações da tag do objeto do bucket de origem. A permissão <code>s3-outposts:GetObjectVersionTagging</code> pode estar faltando para o bucket de origem.
<code>SrcHeadObjectNotPermitted</code>	O S3 no Outposts não consegue recuperar metadados de objetos do bucket de origem. A permissão <code>s3-outposts:GetObjectVersionForReplication</code> pode estar faltando para o bucket de origem.
<code>SrcObjectNotEligible</code>	O objeto não é elegível para replicação. O objeto ou suas tags de objeto não correspondem à configuração de replicação.

Para ter mais informações sobre como solucionar erros de replicação, consulte os seguintes tópicos:

- [Criar um perfil do IAM](#)

- [Solução de problemas de replicação](#)

Monitorar o EventBridge com o CloudWatch

Para monitoramento, o Amazon EventBridge é integrado ao Amazon CloudWatch. O EventBridge envia automaticamente métricas ao CloudWatch a cada minuto. Essas métricas incluem o número de [eventos](#) que foram correspondidos por uma [regra](#) e o número de vezes que um [destino](#) é invocado por uma regra. Quando uma regra é executada no EventBridge, todos os destinos associados à regra são invocados. Você pode monitorar o comportamento do EventBridge por meio do CloudWatch das maneiras a seguir.

- Você pode monitorar as [métricas disponíveis do EventBridge](#) para suas regras do EventBridge no painel do CloudWatch. Depois, você pode usar os recursos do CloudWatch, como os alarmes do CloudWatch, para definir alarmes em determinadas métricas. Se essas métricas atingirem os valores de limite personalizados que você especificou nos alarmes, você receberá notificações e poderá agir adequadamente.
- Você pode definir o Amazon CloudWatch Logs como destino de sua regra do EventBridge. Depois, o EventBridge cria fluxos de log e o CloudWatch Logs armazena o texto dos eventos como entradas de log. Para ter mais informações, consulte [EventBridge and CloudWatch Logs](#).

Para ter mais informações sobre como depurar eventos de entrega e arquivamento de eventos do EventBridge, consulte os seguintes tópicos:

- [Política de repetição de eventos e uso de filas de mensagens não entregues](#)
- [Arquivar eventos do EventBridge](#)

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
S3 on Outposts	S3 on Outposts
Permite que o usuário crie buckets no Outposts e pontos de acesso, e adicione objetos a esses buckets.	Permite que o usuário crie buckets no Outposts e pontos de acesso, e adicione objetos a esses buckets.
Sub-redes	Outposts
Permite que o usuário use a nuvem privada virtual (VPC) e os endpoints associados à sub-rede.	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.

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.

- Use a Opção 1 ou a Opção 2 nos [pré-requisitos](#) 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.

- 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](#) (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	<i>111122223333</i>
Nome do ponto de acesso	<i>example-outpost-access-point</i>
ID do Outpost	<i>op-01ac5d28a6a232904</i>
Nome do bucket do Outpost	<i>DOC-EXAMPLE-BUCKET1</i>
ID da VPC	<i>vpc-1a2b3c4d5e6f7g8h9</i>

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 AWS111122223333. 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 XXXXXXXX
```

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

Monitoramento 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?](#)

Para obter mais informações sobre como monitorar 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](#)
- [Como receber notificações de eventos do S3 no Outposts usando o Amazon CloudWatch Events](#)
- [Monitoramento do S3 no Outposts com logs do AWS CloudTrail](#)

Como gerenciar a capacidade do S3 no Outposts com as métricas do Amazon CloudWatch

Para ajudar a gerenciar a capacidade fixa do S3 em seu Outpost, recomendamos que você crie alertas do CloudWatch que avisam quando a utilização do armazenamento excede determinado limite. Para obter mais informações sobre as métricas do CloudWatch para o S3 no Outposts, consulte [Métricas do CloudWatch](#). Se não houver espaço suficiente para armazenar um objeto no Outpost, a API retornará uma isenção de capacidade insuficiente (ICE). Para liberar espaço, você pode criar alarmes do CloudWatch que acionam a exclusão explícita de dados ou usar uma política de validade do ciclo de vida para expirar objetos. Para salvar os dados antes da exclusão, você pode usar o AWS DataSync para copiar os dados do Amazon S3 no Outposts para um bucket do S3 em uma Região da AWS. 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. As métricas relacionadas ao bucket ou à conta existem para todo o uso direto do S3. O uso indireto do S3, como o armazenamento de snapshots locais do Amazon Elastic Block Store ou de backups do Amazon Relational Database Service em um Outpost, consome a capacidade do S3, mas não é incluído nas métricas relacionadas ao bucket ou à conta. Para obter mais informações sobre snapshots locais do Amazon EBS, consulte [Snapshots locais do Amazon EBS em Outposts](#). Para ver seu relatório de custos do Amazon EBS, acesse <https://console.aws.amazon.com/billing/>.

Note

O S3 on Outposts apenas é compatível com as métricas do Amazon S3 a seguir. Como o S3 no Outposts tem um limite de capacidade fixo, recomendamos que você crie alertas do CloudWatch que avisam quando a utilização do armazenamento excede determinado limite.

Métrica	Descrição	Período	Unidades	Type (Tipo)
<code>OutpostTotalBytes</code>	A capacidade provisionada total em bytes para um Outpost.	5 minutos	Bytes	S3 on Outposts
<code>OutpostFreeBytes</code>	A contagem de bytes livres disponíveis em um Outpost para armazenar dados de clientes.	5 minutos	Bytes	S3 on Outposts
<code>BucketUsedBytes</code>	O tamanho total de todos os objetos de determinado bucket.	5 minutos	Bytes	S3 no Outposts. Somente uso direto do S3.
<code>AccountUsedBytes</code>	O tamanho total de todos os objetos da conta do Outposts especificada.	5 minutos	Bytes	S3 no Outposts. Somente uso direto do S3.

Métrica	Descrição	Período	Unidades	Type (Tipo)
BytesPer ingRepl: ation	O número total de bytes de objetos com replicação pendente para determinada regra de replicação. Para obter mais informações sobre como habilitar métricas de replicação, consulte Criar regras de replicação entre Outposts .	5 minutos	Bytes	Opcional. Para replicação do S3 no Outposts.
Operatio sPending eplicat: n	O número total de operações com replicação pendente para uma determinada regra de replicação. Para obter mais informações sobre como habilitar métricas de replicação, consulte Criar regras de replicação entre Outposts .	5 minutos	Contagem	Opcional. Para replicação do S3 no Outposts.
Replica onLatenc	O número atual de atraso em segundos pelo qual o bucket de destino da replicação está atrás do bucket de origem de determinada regra de replicação. Para obter mais informações sobre como habilitar métricas de replicação, consulte Criar regras de replicação entre Outposts .	5 minutos	Segundos	Opcional. Para replicação do S3 no Outposts.

Como receber notificações de eventos do S3 no Outposts usando o Amazon CloudWatch Events

Você pode usar o CloudWatch Events para criar qualquer evento de API do Amazon S3 no Outposts. Ao criar uma regra, você pode optar por receber notificações por meio de todos os destinos compatíveis do CloudWatch, incluindo 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 no 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"  
      ]  
    }  
  }  
}
```

Monitoramento do S3 no Outposts com logs do AWS CloudTrail

O Amazon S3 no Outposts é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações tomadas por um usuário, uma função ou um AWS service (Serviço da AWS) no S3 no Outposts. Você pode usar o AWS CloudTrail para obter informações sobre o S3 em solicitações no nível de bucket e de objeto do Outposts para auditar e registrar em log sua atividade de eventos do S3 no Outposts. Para habilitar eventos de dados do CloudTrail para todos os buckets do Outposts ou para uma lista de buckets do Outposts específicos, você deve [criar uma trilha manualmente no CloudTrail](#). Para obter mais informações sobre as entradas do arquivo de log do CloudTrail, consulte [Entradas de arquivo de log do S3 no Outposts](#).

Note

- É uma prática recomendada criar uma política de ciclo de vida para o bucket do Outposts de evento de dados do AWS CloudTrail. Configure a política de ciclo de vida para remover periodicamente arquivos de log após o período necessário para auditá-los. Fazer isso reduz a quantidade de dados que o Amazon Athena analisa para cada consulta. Para obter mais informações, consulte [Definir uma configuração de ciclo de vida em um bucket](#).
- 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 log do CloudTrail para objetos em um bucket do S3 no Outposts

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 Amazon S3 no Outposts. O CloudTrail é compatível com o registro em log de operações de API no nível do objeto do S3 no Outposts, 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. No entanto, é possível configurar trilhas para registrar em log eventos de dados para buckets do S3 no Outposts que você especifica, ou registrar em log eventos para todos os buckets do S3 no Outposts em sua Conta da AWS. Para obter mais informações, consulte [Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail](#).


O CloudTrail não preenche eventos de dados no histórico de eventos do CloudTrail. Além disso, nem todas as operações de API no nível do bucket do S3 no Outposts são preenchidas no histórico de eventos do CloudTrail. Para obter mais informações sobre como consultar os logs do CloudTrail, consulte [Usar padrões de filtro do Amazon CloudWatch Logs e o Amazon Athena para consultar logs do CloudTrail](#) no Centro de Conhecimento da AWS.

Para configurar uma trilha para registrar em log eventos de dados para um bucket do S3 no Outposts, 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 S3 no Outposts 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 no Outposts, 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 no Outposts.

 Note

A proprietária do bucket é a Conta da AWS que o criou e só ela pode configurar eventos de dados do S3 no Outposts a serem enviados ao AWS CloudTrail.

Para habilitar registro de eventos de dados do CloudTrail para um bucket do S3 no Outposts

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 nome do bucket do Outposts cujos eventos de dados você deseja registrar usando o CloudTrail.
4. Escolha Properties (Propriedades).
5. Na seção Eventos de dados do AWS CloudTrail, escolha Configurar no CloudTrail.

O console do AWS CloudTrail será aberto.

Você pode criar uma nova trilha do CloudTrail ou reutilizar uma trilha existente e configurar eventos de dados do S3 no Outposts para serem registrados em sua trilha.

6. Na página Painel do console do CloudTrail, escolha Criar trilha.
7. Na página Etapa 1: escolher atributos da trilha, forneça um nome para a trilha, escolha um bucket do S3 para armazenar logs da trilha, especifique qualquer outras configurações desejadas e escolha Próximo.
8. Na página Etapa 2: escolher eventos de log, em Tipo de evento, escolha Eventos de dados.

Em Tipo de evento de dados, escolha S3 no Outposts. Escolha Next (Próximo).

Note

- Ao criar uma trilha e configurar o registro em log de eventos de dados para o S3 no Outposts, você deve especificar o tipo de evento de dados corretamente.
- Se você usa o console do CloudTrail, escolha o S3 Outposts para Tipo de evento de dados. 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 S3 no Outposts no console do CloudTrail, consulte [Registro em log de eventos de dados para objetos do Amazon S3](#) no Guia do usuário do AWS CloudTrail.
- Se você usar a AWS Command Line Interface (AWS CLI) ou os AWS SDKs, defina o campo `resources.type` como `AWS::S3Outposts::Object`. Para obter mais informações sobre como registrar em log eventos de dados do S3 no Outposts com a AWS CLI, consulte [Registrar em log eventos do S3 no Outposts](#) no Guia do usuário do AWS CloudTrail.
- 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 no

Outposts, o console do Amazon S3 mostra que o registro está habilitado no nível do objeto para o bucket.

9. Na página Etapa 3: revisar e criar, revise os atributos da trilha e os eventos de log que você configurou. Depois, escolha Criar trilha.

Para desabilitar o registro de eventos de dados do CloudTrail para objetos em um bucket do S3 no Outposts

1. Faça login no AWS Management Console e abra o console do CloudTrail em <https://console.aws.amazon.com/cloudtrail/>.
2. No painel de navegação à esquerda, selecione Trilhas.
3. Escolha o nome da trilha que você criou para registrar em log eventos para o bucket do S3 no Outposts.
4. Na página de detalhes da trilha, escolha Parar o registro no canto superior direito.
5. Na caixa de diálogo exibida, selecione Parar o registro.

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?](#)

Os tópicos a seguir fornecem informações sobre desenvolvimento com o S3 on Outposts.

Tópicos

- [Operações de API do Amazon S3 on Outposts](#)
- [Configurar o cliente de controle do S3 para S3 on Outposts usando o SDK para Java](#)

- [Fazer solicitações ao S3 no Outposts por IPv6](#)

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](#)
- [Operações de API do Amazon S3 Control para gerenciar buckets](#)
- [Operações de API do S3 on Outposts para gerenciar Outposts](#)

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. É necessário usar pontos de acesso para acessar qualquer objeto em um bucket do Outpost. Ao usar uma operação de API de objeto com o S3 no Outposts, você fornece o nome do recurso da Amazon (ARN) do ponto de acesso do Outposts ou o alias do ponto de acesso. Para obter mais informações sobre alias de pontos de acesso, consulte [Usar um alias em estilo de bucket para seu ponto de acesso de bucket do S3 no Outposts](#).

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](#)
- [ListObjectVersions](#)
- [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](#)
- [DeleteBucketReplication](#)
- [DeleteBucketTagging](#)
- [GetAccessPoint](#)
- [GetAccessPointPolicy](#)
- [GetBucket](#)
- [GetBucketLifecycleConfiguration](#)
- [GetBucketPolicy](#)
- [GetBucketReplication](#)
- [GetBucketTagging](#)
- [GetBucketVersioning](#)

- [ListAccessPoints](#)
- [ListRegionalBuckets](#)
- [PutAccessPointPolicy](#)
- [PutBucketLifecycleConfiguration](#)
- [PutBucketPolicy](#)
- [PutBucketReplication](#)
- [PutBucketTagging](#)
- [PutBucketVersioning](#)

Operações de API do S3 on Outposts para gerenciar Outposts

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](#)
- [ListOutpostsWithS3](#)
- [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.AWSSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;

public AWSS3Control createS3ControlClient() {

    String accessKey = AWSSAccessKey;
    String secretKey = SecretAccessKey;
```

```
BasicAWSCredentials awsCreds = new BasicAWSCredentials(accessKey, secretKey);

return AWSS3ControlClient.builder().enableUseArnRegion()
    .withCredentials(new AWSStaticCredentialsProvider(awsCreds))
    .build();
}
```

Fazer solicitações ao S3 no Outposts por IPv6

O Amazon S3 no Outposts e os endpoints de pilha dupla do S3 no Outposts oferecem suporte a solicitações para buckets do S3 no Outposts com os protocolos IPv6 e IPv4. Com o suporte a IPv6 para o S3 no Outposts, você pode acessar e operar os buckets e os recursos do ambiente de gerenciamento por meio das APIs do S3 no Outposts em redes IPv6.

Note

As [ações em objetos do S3 no Outposts](#) (como PutObject ou GetObject) não são compatíveis em redes IPv6.

Não há nenhum custo adicional para acessar o S3 no Outposts em redes IPv6. Para obter mais informações sobre o S3 no Outposts, consulte [Preços de racks do AWS Outposts](#).

Tópicos


- [Conceitos básicos do IPv6](#)
- [Usar endpoints de pilha dupla para fazer solicitações em uma rede IPv6](#)
- [Como usar endereços do IPv6 em políticas do IAM](#)
- [Testar a compatibilidade com endereços IP](#)
- [Usar IPv6 com o AWS PrivateLink](#)
- [Usar endpoints de pilha dupla do S3 no Outposts](#)

Conceitos básicos do IPv6

Para fazer uma solicitação a um bucket do S3 no Outposts por IPv6, é necessário usar um endpoint de pilha dupla. A próxima seção descreve como fazer solicitações por meio do IPv6 usando endpoints de pilha dupla.

Veja a seguir algumas considerações importantes antes de tentar acessar um bucket do S3 no Outposts por 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 ter mais informações, consulte [Usar endpoints de pilha dupla do S3 no Outposts](#).
- Se você usar a filtragem de endereços IP de origem nas políticas de usuário do AWS Identity and Access Management (IAM) ou de bucket do S3 no Outposts, será necessário atualizar as políticas para incluir intervalos de endereços IPv6.

 Note

Esse requisito se aplica somente às operações de bucket do S3 no Outposts e aos recursos do ambiente de gerenciamento em redes IPv6. As [ações em objetos do Amazon S3 no Outposts](#) não são compatíveis em redes IPv6.

- Ao usar o IPv6, os arquivos de log de acesso ao servidor fornecem endereços IP em um formato do IPv6. É necessário atualizar as ferramentas, os scripts e o software existentes que você usa para analisar os arquivos de log do S3 no Outposts, para que eles possam analisar os endereços IP remotos formatados para IPv6. As ferramentas, os scripts e o software atualizados analisarão corretamente os endereços IP remotos formatados para IPv6.

Usar endpoints de pilha dupla para fazer solicitações em uma rede IPv6

Para fazer solicitações com chamadas de API do S3 no Outposts por IPv6, você pode usar endpoints de pilha dupla pela AWS CLI ou pelo AWS SDK. As [operações de API de controle do Amazon S3](#) e as [operações de API do S3 no Outposts](#) funcionam da mesma forma, independentemente de você estar acessando o S3 no Outposts por meio de um protocolo IPv6 ou IPv4. No entanto, lembre-se de que as [ações em objetos do S3 no Outposts](#) (como PutObject ou GetObject) não são compatíveis em redes IPv6.

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 S3 no Outposts no arquivo de configuração.

Você pode usar um endpoint de pilha dupla para acessar um bucket do S3 no Outposts por IPv6 de qualquer um dos seguintes:

- A AWS CLI, consulte [Usar endpoints de pilha dupla da AWS CLI](#).
- Os AWS SDKs, consulte [Usar endpoints de pilha dupla do S3 no Outposts com os AWS SDKs](#).

Como usar endereços do IPv6 em políticas do IAM

Antes de tentar acessar um bucket do S3 no Outposts usando um protocolo IPv6, garanta que os usuários do IAM ou as políticas de bucket do S3 no Outposts usadas para filtragem de endereços IP estejam atualizadas para incluir intervalos de endereços IPv6. Se as políticas de filtragem de endereços IP não estiverem atualizadas para lidar com endereços IPv6, você poderá perder o acesso a um bucket do S3 no Outposts ao tentar usar o protocolo IPv6.

As políticas do IAM que filtram endereços IP usam [operadores de condição de endereço IP](#). A política de bucket do S3 no Outposts a seguir identifica o intervalo 54.240.143.* de endereços IPv4 permitidos usando operadores de condição de endereço IP. Todos os endereços IP fora desse intervalo terão o acesso ao bucket do S3 no Outposts negado (DOC-EXAMPLE-BUCKET). Como todos os endereços do IPv6 estão fora do intervalo permitido, essa política impede que os endereços do IPv6 possam acessar o DOC-EXAMPLE-BUCKET.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3outposts:*",
      "Resource": "arn:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
bucket/DOC-EXAMPLE-BUCKET/*",
      "Condition": {
        "IpAddress": {"aws:SourceIp": "54.240.143.0/24"}
      }
    }
  ]
}
```

Você pode modificar o elemento `Condition` da política de bucket do S3 no Outposts para permitir intervalos de endereços IPv4 (54.240.143.0/24) e IPv6 (2001:DB8:1234:5678::/64), conforme mostrado no exemplo a seguir. Você pode usar o mesmo tipo de bloqueio de `Condition` mostrado no exemplo para atualizar as políticas de usuário e de bucket do IAM.

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

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

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 como editar políticas de bucket do S3 no Outposts, consulte [Adicionar ou editar uma política para um bucket do Amazon S3 on Outposts](#).

Testar a compatibilidade com endereços IP

Se você estiver usando uma instância do Linux ou do Unix ou a plataforma macOS X, poderá testar o acesso a um endpoint de pilha dupla por IPv6. Por exemplo, para testar a conexão com endpoints do Amazon S3 no Outposts via IPv6, use o comando `dig`:

```
dig s3-outposts.us-west-2.api.aws AAAA +short
```

Se o endpoint de pilha dupla em uma rede IPv6 estiver configurado corretamente, o comando `dig` retornará os endereços IPv6 conectados. Por exemplo:

```
dig s3-outposts.us-west-2.api.aws AAAA +short
```

```
2600:1f14:2588:4800:b3a9:1460:159f:ebce
```

```
2600:1f14:2588:4802:6df6:c1fd:ef8a:fc76
```

```
2600:1f14:2588:4801:d802:8ccf:4e04:817
```

Usar IPv6 com o AWS PrivateLink

O S3 no Outposts é compatível com o protocolo IPv6 para serviços e endpoints do AWS PrivateLink. Com o suporte ao AWS PrivateLink para o protocolo IPv6, você pode se conectar aos endpoints de serviço em sua VPC por meio de redes IPv6, tanto de conexões on-premises quanto de outras conexões privadas. O suporte a IPv6 para o [AWS PrivateLink para S3 no Outposts](#) também permite a integração do AWS PrivateLink com endpoints de pilha dupla. Para conferir etapas sobre como habilitar o IPv6 para o AWS PrivateLink, consulte [Expedite your IPv6 adoption with AWS PrivateLink services and endpoints](#).

Note

Para atualizar o tipo de endereço IP compatível de IPv4 para IPv6, consulte [Modify the supported IP address type](#) no Guia do usuário do AWS PrivateLink.

Usar IPv6 com o AWS PrivateLink

Se você estiver usando o AWS PrivateLink com IPv6, deverá criar um endpoint de interface da VPC IPv6 ou de pilha dupla. Para conferir etapas gerais sobre como criar um endpoint da VPC usando o AWS Management Console, consulte [Access an AWS service using an interface VPC endpoint](#) no Guia do usuário do AWS PrivateLink.

AWS Management Console

Use o procedimento a seguir para criar um endpoint de interface da VPC que se conecta ao S3 no Outposts.

1. Faça login no AWS Management Console e abra o console da VPC em <https://console.aws.amazon.com/vpc/>.
2. No painel de navegação, escolha Endpoints.
3. Escolha Criar endpoint.
4. Em Categoria do serviço, escolha Serviços do AWS.
5. Em Nome do serviço, escolha o serviço S3 no Outposts (com.amazonaws.us-east-1.s3-outposts).
6. Em VPC, escolha a VPC de onde você acessará o S3 no Outposts.
7. Em Sub-redes, selecione uma sub-rede por zona de disponibilidade da qual você acessará o S3 no Outposts. Não é possível selecionar várias sub-redes em uma mesma zona

de disponibilidade. Será criada uma interface de rede do endpoint para cada sub-rede selecionada. Por padrão, os endereços IP dos intervalos de endereços IP da sub-rede são atribuídos às interfaces de rede do endpoint. Para designar um endereço IP para uma interface de rede do endpoint, selecione Designar endereços IP e insira um endereço IPv6 do intervalo de endereços da sub-rede.


8. Em Tipo de endereço IP, escolha Dualstack. Atribua endereços IPv4 e IPv6 às interfaces de rede do endpoint. Só haverá suporte para esta opção se todas as sub-redes selecionadas tiverem intervalos de endereços IPv4 e IPv6.
9. Em Grupos de segurança, selecione os grupos de segurança que deseja associar às interfaces de rede do endpoint para o endpoint da VPC. Por padrão, o grupo de segurança padrão é associado à VPC.
10. Em Política, escolha Acesso total para permitir todas as operações de todas as entidades principais em todos os recursos no endpoint da VPC. Como alternativa, escolha Personalizado para anexar uma política de endpoint da VPC que controle as permissões das entidades principais para realizar ações em recursos pelo endpoint da VPC. Essa opção ficará disponível somente se o serviço for compatível com as políticas de endpoint da VPC. Para obter mais informações, consulte [Endpoint policies](#).
11. (Opcional) Para adicionar uma tag, escolha Adicionar nova tag e insira a chave e o valor da tag.
12. Escolha Criar endpoint.

Example : política de bucket do S3 no Outposts

Para permitir que o S3 no Outposts interaja com os endpoints da VPC, você pode atualizar sua política do S3 no Outposts da seguinte forma:

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3-outposts:*",
      "Resource": "*",
      "Principal": "*"
    }
  ]
}
```

AWS CLI

 Note

Para habilitar a rede IPv6 em um endpoint da VPC, é necessário definir IPv6 para o filtro `SupportedIpAddressType` para o S3 no Outposts.

O exemplo a seguir usa o comando `create-vpc-endpoint` para criar um endpoint de interface de pilha dupla.

```
aws ec2 create-vpc-endpoint \  
--vpc-id vpc-12345678 \  
--vpc-endpoint-type Interface \  
--service-name com.amazonaws.us-east-1.s3-outposts \  
--subnet-id subnet-12345678 \  
--security-group-id sg-12345678 \  
--ip-address-type dualstack \  
--dns-options "DnsRecordIpType=dualstack"
```

Dependendo da configuração do serviço AWS PrivateLink, as conexões de endpoint recém-criadas talvez precisem ser aceitas pelo provedor de serviços do endpoint da VPC antes de serem usadas. Para obter mais informações, consulte [Aceitar ou rejeitar solicitações de conexão](#) no Guia do usuário do AWS PrivateLink.

O exemplo a seguir usa o comando `modify-vpc-endpoint` para atualizar o endpoint da VPC somente IPv para um endpoint de pilha dupla. O endpoint de pilha dupla permite acesso às redes IPv4 e IPv6.

```
aws ec2 modify-vpc-endpoint \  
--vpc-endpoint-id vpce-12345678 \  
--add-subnet-ids subnet-12345678 \  
--remove-subnet-ids subnet-12345678 \  
--ip-address-type dualstack \  
--dns-options "DnsRecordIpType=dualstack"
```

Para obter mais informações sobre como habilitar a rede IPv6 para o AWS PrivateLink, consulte [Expedite your IPv6 adoption with AWS PrivateLink services and endpoints](#).

Usar endpoints de pilha dupla do S3 no Outposts

Os endpoints de pilha dupla do S3 no Outposts oferecem suporte a solicitações para buckets do S3 no Outposts por IPv6 e IPv4. Esta seção descreve como usar os endpoints de pilha dupla do S3 no Outposts.

Tópicos

- [Endpoints de pilha dupla do S3 no Outposts](#)
- [Usar endpoints de pilha dupla da AWS CLI](#)
- [Usar endpoints de pilha dupla do S3 no Outposts com os AWS SDKs](#)

Endpoints de pilha dupla do S3 no Outposts

Quando você faz uma solicitação para um endpoint de pilha dupla, o URL do bucket do S3 no Outposts é resolvido para um endereço IPv6 ou IPv4. Para obter mais informações sobre como acessar um bucket do S3 no Outposts por IPv6, consulte [Fazer solicitações ao S3 no Outposts por IPv6](#).

Para acessar um bucket do S3 no Outposts por meio de um endpoint de pilha dupla, use um nome de endpoint do tipo caminho. O S3 no Outposts oferece suporte apenas a nomes regionais de endpoint de pilha dupla, o que significa que você deve especificar a região como parte do nome.

Para endpoints FIPS de pilha dupla do tipo caminho, use a seguinte convenção de nomenclatura:

```
s3-outposts-fips.region.api.aws
```

Para endpoints não FIPS de pilha dupla, use a seguinte convenção de nomenclatura:

```
s3-outposts.region.api.aws
```

Note

Os nomes de endpoint do tipo hospedado virtual não são compatíveis com o S3 no Outposts.

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 [Como começar a usar a AWS CLI e o SDK para Java](#).

Defina o valor de configuração `use_dualstack_endpoint` como `true` em um perfil no 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`.

Ao usar endpoints de pilha dupla com a AWS CLI, somente o tipo de endereçamento `path` é compatível. O tipo de endereçamento, definido no arquivo de configuração, determina se o nome do bucket está no nome do host ou no URL. Para obter mais informações, consulte [s3outposts](#) no Guia de Usuário AWS CLI.

Para usar um endpoint de pilha dupla por meio da AWS CLI, use o parâmetro `--endpoint-url` com o endpoint `http://s3.dualstack.region.amazonaws.com` ou `https://s3-outposts-fips.region.api.aws` em qualquer comando `s3control` ou `s3outposts`.

Por exemplo:

```
$ aws s3control list-regional-buckets --endpoint-url https://s3-outposts.region.api.aws
```

Usar endpoints de pilha dupla do S3 no Outposts com os AWS SDKs

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

AWS SDK for Java 2.x Exemplo do endpoint de pilha dupla do

Os exemplos a seguir mostram como usar as classes `S3ControlClient` e `S3OutpostsClient` para habilitar endpoints de pilha dupla ao criar um cliente do S3 no Outposts usando o AWS SDK for Java 2.x. Para obter instruções sobre como criar e testar um exemplo funcional em Java para o Amazon S3 no Outposts, consulte [Como começar a usar a AWS CLI e o SDK para Java](#).

Example : criar uma classe **S3ControlClient** com endpoints de pilha dupla habilitados

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
```

```
import software.amazon.awssdk.services.s3control.model.ListRegionalBucketsRequest;
import software.amazon.awssdk.services.s3control.model.ListRegionalBucketsResponse;
import software.amazon.awssdk.services.s3control.model.S3ControlException;

public class DualStackEndpointsExample1 {

    public static void main(String[] args) {
        Region clientRegion = Region.of("us-east-1");
        String accountId = "111122223333";
        String navyId = "9876543210";

        try {
            // Create an S3ControlClient with dual-stack endpoints enabled.
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                .region(clientRegion)
                .dualstackEnabled(true)
                .build();

            ListRegionalBucketsRequest listRegionalBucketsRequest =
                ListRegionalBucketsRequest.builder()

                .accountId(accountId)

                .outpostId(navyId)

                .build();

            ListRegionalBucketsResponse listBuckets =
                s3ControlClient.listRegionalBuckets(listRegionalBucketsRequest);
            System.out.printf("ListRegionalBuckets Response: %s\n",
                listBuckets.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 on Outposts
            // couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
        catch (S3ControlException e) {
            // Unknown exceptions will be thrown as an instance of this type.
            e.printStackTrace();
        }
        catch (SdkClientException e) {
            // Amazon S3 on Outposts couldn't be contacted for a response, or the
            // client
            // couldn't parse the response from Amazon S3 on Outposts.
        }
    }
}
```

```
        e.printStackTrace();
    }
}
}
```

Example : criar um **S3OutpostsClient** com endpoints de pilha dupla habilitados

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3outposts.S3OutpostsClient;
import software.amazon.awssdk.services.s3outposts.model.ListEndpointsRequest;
import software.amazon.awssdk.services.s3outposts.model.ListEndpointsResponse;
import software.amazon.awssdk.services.s3outposts.model.S3OutpostsException;

public class DualStackEndpointsExample2 {

    public static void main(String[] args) {
        Region clientRegion = Region.of("us-east-1");

        try {
            // Create an S3OutpostsClient with dual-stack endpoints enabled.
            S3OutpostsClient s3OutpostsClient = S3OutpostsClient.builder()
                .region(clientRegion)
                .dualstackEnabled(true)
                .build();

            ListEndpointsRequest listEndpointsRequest =
                ListEndpointsRequest.builder().build();

            ListEndpointsResponse listEndpoints =
                s3OutpostsClient.listEndpoints(listEndpointsRequest);
            System.out.printf("ListEndpoints Response: %s\n",
                listEndpoints.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 on Outposts
            // couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
        catch (S3OutpostsException e) {
            // Unknown exceptions will be thrown as an instance of this type.
            e.printStackTrace();
        }
    }
}
```

```
        } catch (SdkClientException e) {  
            // Amazon S3 on Outposts couldn't be contacted for a response, or the  
client  
            // couldn't parse the response from Amazon S3 on Outposts.  
            e.printStackTrace();  
        }  
    }  
}
```

Se você estiver usando o AWS SDK for Java 2.x no Windows, talvez precise definir a seguinte propriedade da máquina virtual Java (JVM):

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

Exemplos de código do Amazon S3 usando SDKs da AWS

Os exemplos de código a seguir mostram como usar o Amazon S3 com um Kit de Desenvolvimento de Software (SDK) da AWS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço específicas, é possível ver as ações contextualizadas em seus devidos cenários e exemplos entre serviços.

Cenários são exemplos de código que mostram como realizar uma tarefa específica chamando várias funções dentro do mesmo serviço.

Exemplos entre serviços são 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](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Conceitos básicos

Olá, Amazon S3

O exemplo de código a seguir mostra como começar a usar o Amazon S3.

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Código para o arquivo CMakeLists.txt do CMake.

```
# Set the minimum required version of CMake for this project.
cmake_minimum_required(VERSION 3.13)

# Set the AWS service components used by this project.
set(SERVICE_COMPONENTS s3)
```

```
# Set this project's name.
project("hello_s3")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
  libraries for the AWS SDK.
  string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
    "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
  list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})

if (WINDOWS_BUILD AND AWSSDK_INSTALL_AS_SHARED_LIBS)
  # Copy relevant AWS SDK for C++ libraries into the current binary directory
  for running and debugging.

  # set(BIN_SUB_DIR "/Debug") # if you are building from the command line you
  may need to uncomment this
  # and set the proper subdirectory to the executables' location.

  AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
    ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
  hello_s3.cpp)

target_link_libraries(${PROJECT_NAME}
  ${AWSSDK_LINK_LIBRARIES})
```

Código para o arquivo de origem hello_s3.cpp.

```
#include <aws/core/Aws.h>
```

```
#include <aws/s3/S3Client.h>
#include <iostream>
#include <aws/core/auth/AWSCredentialsProviderChain.h>
using namespace Aws;
using namespace Aws::Auth;

/*
 * A "Hello S3" starter application which initializes an Amazon Simple Storage
 * Service (Amazon S3) client
 * and lists the Amazon S3 buckets in the selected region.
 *
 * main function
 *
 * Usage: 'hello_s3'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
    // options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
    Aws::InitAPI(options); // Should only be called once.
    int result = 0;
    {
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

        // You don't normally have to test that you are authenticated. But the
        // S3 service permits anonymous requests, thus the s3Client will return "success"
        // and 0 buckets even if you are unauthenticated, which can be confusing to a new
        // user.

        auto provider =
            Aws::MakeShared<DefaultAWSCredentialsProviderChain>("alloc-tag");
        auto creds = provider->GetAWSCredentials();
        if (creds.IsEmpty()) {
            std::cerr << "Failed authentication" << std::endl;
        }

        Aws::S3::S3Client s3Client(clientConfig);
        auto outcome = s3Client.ListBuckets();

        if (!outcome.IsSuccess()) {
```




```
        std::cerr << "Failed with error: " << outcome.GetError() <<
std::endl;
        result = 1;
    } else {
        std::cout << "Found " << outcome.GetResult().GetBuckets().size()
            << " buckets\n";
        for (auto &bucket: outcome.GetResult().GetBuckets()) {
            std::cout << bucket.GetName() << std::endl;
        }
    }
}

Aws::ShutdownAPI(options); // Should only be called once.
return result;
}
```

- Para obter informações da API, consulte [ListBuckets](#) na Referência da API AWS SDK for C++.

Go

SDK para Go V2

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
package main

import (
    "context"
    "fmt"

    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/s3"
)
```

```
// main uses the AWS SDK for Go V2 to create an Amazon Simple Storage Service
// (Amazon S3) client and list up to 10 buckets in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
    sdkConfig, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        fmt.Println("Couldn't load default configuration. Have you set up your AWS
account?")
        fmt.Println(err)
        return
    }
    s3Client := s3.NewFromConfig(sdkConfig)
    count := 10
    fmt.Printf("Let's list up to %v buckets for your account.\n", count)
    result, err := s3Client.ListBuckets(context.TODO(), &s3.ListBucketsInput{})
    if err != nil {
        fmt.Printf("Couldn't list buckets for your account. Here's why: %v\n", err)
        return
    }
    if len(result.Buckets) == 0 {
        fmt.Println("You don't have any buckets!")
    } else {
        if count > len(result.Buckets) {
            count = len(result.Buckets)
        }
        for _, bucket := range result.Buckets[:count] {
            fmt.Printf("\t\t%v\n", *bucket.Name)
        }
    }
}
```

- Para obter detalhes da API, consulte [ListBuckets](#) na Referência da API AWS SDK for Go.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.Bucket;
import software.amazon.awssdk.services.s3.model.ListBucketsResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class HelloS3 {
    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        listBuckets(s3);
    }

    public static void listBuckets(S3Client s3) {
        try {
            ListBucketsResponse response = s3.listBuckets();
            List<Bucket> bucketList = response.buckets();
            bucketList.forEach(bucket -> {
                System.out.println("Bucket Name: " + bucket.name());
            });
        }
    }
}
```

```
    });

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- Para obter informações da API, consulte [ListBuckets](#) na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
import { ListBucketsCommand, S3Client } from "@aws-sdk/client-s3";

// When no region or credentials are provided, the SDK will use the
// region and credentials from the local AWS config.
const client = new S3Client({});

export const helloS3 = async () => {
    const command = new ListBucketsCommand({});

    const { Buckets } = await client.send(command);
    console.log("Buckets: ");
    console.log(Buckets.map((bucket) => bucket.Name).join("\n"));
    return Buckets;
};
```

- Para obter detalhes da API, consulte [ListBuckets](#) na Referência da API AWS SDK for JavaScript.

PHP

SDK para PHP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
use Aws\S3\S3Client;

$client = new S3Client(['region' => 'us-west-2']);
$results = $client->listBuckets();
var_dump($results);
```

- Para obter informações da API, consulte [ListBuckets](#) na Referência da API AWS SDK for PHP.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
import boto3

def hello_s3():
    """
    Use the AWS SDK for Python (Boto3) to create an Amazon Simple Storage Service
    (Amazon S3) resource and list the buckets in your account.
    This example uses the default settings specified in your shared credentials
    and config files.
    """
```

```
s3_resource = boto3.resource("s3")
print("Hello, Amazon S3! Let's list your buckets:")
for bucket in s3_resource.buckets.all():
    print(f"\t{bucket.name}")

if __name__ == "__main__":
    hello_s3()
```

- Para obter informações da API, consulte [ListBuckets](#) na Referência da API AWS SDK para Python (Boto3).

Exemplos de código

- [Ações do Amazon S3 usando SDKs da AWS](#)
 - [Usar AbortMultipartUpload com um SDK da AWS ou uma ferramenta de linha de comandos](#)
 - [Usar AbortMultipartUploads com um SDK da AWS ou uma ferramenta de linha de comandos](#)
 - [Usar CompleteMultipartUpload com um AWS SDK ou ferramenta de linha de comando](#)
 - [Usar CopyObject com um SDK da AWS ou uma ferramenta de linha de comandos](#)
 - [Usar CreateBucket com um SDK da AWS ou uma ferramenta de linha de comandos](#)
 - [Usar CreateMultiRegionAccessPoint com um AWS SDK ou ferramenta de linha de comando](#)
 - [Usar CreateMultipartUpload com um AWS SDK ou ferramenta de linha de comando](#)
 - [Usar DeleteBucket com um SDK da AWS ou uma ferramenta de linha de comandos](#)
 - [Usar DeleteBucketAnalyticsConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
 - [Usar DeleteBucketCors com um SDK da AWS ou uma ferramenta de linha de comandos](#)
 - [Usar DeleteBucketEncryption com um SDK da AWS ou uma ferramenta de linha de comandos](#)
 - [Usar DeleteBucketInventoryConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
 - [Usar DeleteBucketLifecycle com um SDK da AWS ou uma ferramenta de linha de comandos](#)
 - [Usar DeleteBucketMetricsConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
 - [Usar DeleteBucketPolicy com um SDK da AWS ou uma ferramenta de linha de comandos](#)
 - [Usar DeleteBucketReplication com um SDK da AWS ou uma ferramenta de linha de comandos](#)

- [Usar DeleteBucketTagging com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar DeleteBucketWebsite com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar DeleteObject com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar DeleteObjectTagging com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar DeleteObjects com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar DeletePublicAccessBlock com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketAccelerateConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketAcl com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketAnalyticsConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketCors com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketEncryption com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketInventoryConfiguration com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetBucketLifecycleConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketLocation com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketLogging com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetBucketMetricsConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketNotification com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketPolicy com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketPolicyStatus com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketReplication com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetBucketRequestPayment com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetBucketTagging com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetBucketVersioning com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetBucketWebsite com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetObject com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetObjectAcl com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetObjectLegalHold com um SDK da AWS ou uma ferramenta de linha de comandos](#)

- [Usar GetObjectLockConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetObjectRetention com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetObjectTagging com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetPublicAccessBlock com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar HeadBucket com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar HeadObject com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar ListBucketAnalyticsConfigurations com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar ListBucketInventoryConfigurations com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar ListBuckets com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar ListMultipartUploads com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar ListObjectVersions com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar ListObjects com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar ListObjectsV2 com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketAccelerateConfiguration com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketAcl com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar PutBucketCors com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketEncryption com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketLifecycleConfiguration com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketLogging com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketNotification com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketNotificationConfiguration com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketPolicy com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketReplication com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketRequestPayment com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar PutBucketTagging com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketVersioning com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar PutBucketWebsite com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar PutObject com um AWS SDK ou ferramenta de linha de comando](#)

- [Usar PutObjectAcl com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar PutObjectLegalHold com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutObjectLockConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar PutObjectRetention com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar RestoreObject com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar SelectObjectContent com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar UploadPart com um AWS SDK ou ferramenta de linha de comando](#)
- [Cenários do Amazon S3 usando AWS SDKs](#)
 - [Criar um URL pré-assinado para o Amazon S3 usando um AWS SDK](#)
 - [Uma página da web que oferece uma lista de objetos do Amazon S3 usando um AWS SDK](#)
 - [Fazer download de todos os objetos de um bucket do Amazon Simple Storage Service \(Amazon S3\) em um diretório local](#)
 - [Obter um objeto do Amazon S3 de um ponto de acesso multirregional usando um AWS SDK](#)
 - [Obter um objeto de um bucket do Amazon S3 usando um AWS SDK e especificando um cabeçalho If-Modified-Since](#)
 - [Conceitos básicos de buckets e objetos do Amazon S3 usando um AWS SDK](#)
 - [Conceitos básicos de criptografia para objetos do Amazon S3 usando um AWS SDK](#)
 - [Conceitos básicos de etiquetas para objetos do Amazon S3 usando um AWS SDK](#)
 - [Obter a configuração de retenção legal de um objeto do Amazon S3 usando um SDK da AWS](#)
 - [Trabalhar com os recursos de bloqueio de objetos do Amazon S3 usando um SDK da AWS](#)
 - [Gerenciar listas de controle de acesso \(ACLs\) para buckets do Amazon S3 usando um AWS SDK](#)
 - [Gerenciar objetos do Amazon S3 versionados em lotes com uma função do Lambda usando um AWS SDK](#)
 - [Analisar URIs do Amazon S3 usando um AWS SDK](#)
 - [Executar uma cópia multipart de um objeto do Amazon S3 usando um AWS SDK](#)
 - [Realizar um carregamento fragmentado em um objeto do Amazon S3 usando um SDK da AWS](#)
 - [Monitorar o upload ou download de um objeto do Amazon S3 usando um SDK da AWS](#)
 - [Exemplos de abordagens para testes de unidade e integração com SDK da AWS](#)

- [Fazer upload recursivo de um diretório local em um bucket do Amazon Simple Storage Service \(Amazon S3\)](#)
- [Fazer upload ou download de arquivos grandes de e para o Amazon S3 usando um AWS SDK](#)
- [Faça upload de um fluxo de tamanho desconhecido em um objeto do Amazon S3 usando um AWS SDK](#)
- [Usar somas de verificação para trabalhar com um objeto do Amazon S3 usando um AWS SDK](#)
- [Trabalhar com objetos versionados do Amazon S3 usando um AWS SDK](#)
- [Exemplos sem servidor para o Amazon S3 usando AWS SDKs](#)
 - [Invocar uma função do Lambda em um acionador do Amazon S3](#)
- [Exemplos do Amazon S3 entre serviços usando AWS SDKs](#)
 - [Criar uma aplicação Amazon Transcribe](#)
 - [Converter texto em fala e de volta em texto usando um AWS SDK](#)
 - [Criar uma aplicação de gerenciamento de ativos de fotos que permita que os usuários gerenciem fotos usando rótulos](#)
 - [Criar uma aplicação de exploração do Amazon Textract](#)
 - [Detectar EPI em imagens com o Amazon Rekognition usando um AWS SDK](#)
 - [Detectar entidades em texto extraído de uma imagem usando um AWS SDK](#)
 - [Detectar faces em uma imagem usando um AWS SDK](#)
 - [Detectar objetos em imagens com o Amazon Rekognition usando um AWS SDK](#)
 - [Detectar pessoas e objetos em um vídeo com o Amazon Rekognition usando um AWS SDK](#)
 - [Salvar o EXIF e outras informações de imagem usando um AWS SDK](#)
 - [Como transformar dados para sua aplicação com o S3 Object Lambda](#)

Ações do Amazon S3 usando SDKs da AWS

Os exemplos de código a seguir demonstram como realizar ações específicas do Amazon S3 com SDKs da AWS. Esses trechos chamam a API do Amazon S3 e são trechos de código de programas maiores que devem ser executados no contexto. Cada exemplo inclui um link para o GitHub, em que você pode encontrar instruções para configurar e executar o código.

Os exemplos a seguir incluem apenas as ações mais utilizadas. Para obter uma lista completa, consulte a [Referência de API do Amazon Simple Storage Service \(Amazon S3\)](#).

Exemplos

- [Usar AbortMultipartUpload com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar AbortMultipartUploads com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar CompleteMultipartUpload com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar CopyObject com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar CreateBucket com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar CreateMultiRegionAccessPoint com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar CreateMultipartUpload com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar DeleteBucket com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar DeleteBucketAnalyticsConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar DeleteBucketCors com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar DeleteBucketEncryption com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar DeleteBucketInventoryConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar DeleteBucketLifecycle com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar DeleteBucketMetricsConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar DeleteBucketPolicy com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar DeleteBucketReplication com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar DeleteBucketTagging com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar DeleteBucketWebsite com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar DeleteObject com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar DeleteObjectTagging com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar DeleteObjects com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar DeletePublicAccessBlock com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketAccelerateConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketAcl com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketAnalyticsConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)

- [Usar GetBucketCors com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketEncryption com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketInventoryConfiguration com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetBucketLifecycleConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketLocation com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketLogging com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetBucketMetricsConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketNotification com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketPolicy com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketPolicyStatus com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetBucketReplication com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetBucketRequestPayment com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetBucketTagging com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetBucketVersioning com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetBucketWebsite com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetObject com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetObjectAcl com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetObjectLegalHold com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetObjectLockConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetObjectRetention com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar GetObjectTagging com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar GetPublicAccessBlock com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar HeadBucket com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar HeadObject com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar ListBucketAnalyticsConfigurations com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar ListBucketInventoryConfigurations com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar ListBuckets com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar ListMultipartUploads com um AWS SDK ou ferramenta de linha de comando](#)

- [Usar ListObjectVersions com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar ListObjects com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar ListObjectsV2 com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketAccelerateConfiguration com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketAcl com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar PutBucketCors com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketEncryption com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketLifecycleConfiguration com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketLogging com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketNotification com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketNotificationConfiguration com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketPolicy com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketReplication com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketRequestPayment com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar PutBucketTagging com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutBucketVersioning com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar PutBucketWebsite com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar PutObject com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutObjectAcl com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar PutObjectLegalHold com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar PutObjectLockConfiguration com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar PutObjectRetention com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar RestoreObject com um SDK da AWS ou uma ferramenta de linha de comandos](#)
- [Usar SelectObjectContent com um AWS SDK ou ferramenta de linha de comando](#)
- [Usar UploadPart com um AWS SDK ou ferramenta de linha de comando](#)

Usar **AbortMultipartUpload** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `AbortMultipartUpload`.

CLI

AWS CLI

Para interromper um carregamento fracionado especificado

O comando `abort-multipart-upload` a seguir interrompe um carregamento fracionado da chave `multipart/01` no bucket `my-bucket`.

```
aws s3api abort-multipart-upload \  
  --bucket my-bucket \  
  --key multipart/01 \  
  --upload-id  
dfRtDYU0WWCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZ1jF.Yxwh6XG7WfS2vC4to6HiV6Yj1x.cph0gtNBtJ8P3U
```

O ID de upload exigido por esse comando é a saída de `create-multipart-upload` e também pode ser recuperado com `list-multipart-uploads`.

- Para obter detalhes da API, consulte [AbortMultipartUpload](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando interrompe os carregamentos fracionados criados há mais de cinco dias.

```
Remove-S3MultipartUpload -BucketName test-files -DaysBefore 5
```

Exemplo 2: este comando interrompe os carregamentos fracionados criados antes de 2 de janeiro de 2014.

```
Remove-S3MultipartUpload -BucketName test-files -InitiatedDate "Thursday, January  
02, 2014"
```

Exemplo 3: este comando interrompe os carregamentos fracionados criados antes de 2 de janeiro de 2014, às 10:45:37.

```
Remove-S3MultipartUpload -BucketName test-files -InitiatedDate "2014/01/02  
10:45:37"
```

- Para obter detalhes da API, consulte [AbortMultipartUpload](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **AbortMultipartUploads** com um SDK da AWS ou uma ferramenta de linha de comandos

O código de exemplo a seguir mostra como usar `AbortMultipartUploads`.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Transfer;

/// <summary>
/// This example shows how to use the Amazon Simple Storage Service
/// (Amazon S3) to stop a multi-part upload process using the Amazon S3
/// TransferUtility.
/// </summary>
public class AbortMPU
{
    public static async Task Main()
    {
        string bucketName = "doc-example-bucket";

        // If the AWS Region defined for your default user is different
        // from the Region where your Amazon S3 bucket is located,
```

```
// pass the Region name to the S3 client object's constructor.
// For example: RegionEndpoint.USWest2.
IAmazonS3 client = new AmazonS3Client();

await AbortMPUAsync(client, bucketName);
}

/// <summary>
/// Cancels the multi-part copy process.
/// </summary>
/// <param name="client">The initialized client object used to create
/// the TransferUtility object.</param>
/// <param name="bucketName">The name of the S3 bucket where the
/// multi-part copy operation is in progress.</param>
public static async Task AbortMPUAsync(IAmazonS3 client, string
bucketName)
{
    try
    {
        var transferUtility = new TransferUtility(client);

        // Cancel all in-progress uploads initiated before the specified
date.

        await transferUtility.AbortMultipartUploadsAsync(
            bucketName, DateTime.Now.AddDays(-7));
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine($"Error: {e.Message}");
    }
}
}
```

- Para obter informações da API, consulte [AbortMultipartUploads](#) na Referência da API do AWS SDK for .NET.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **CompleteMultipartUpload** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o CompleteMultipartUpload.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- [Executar uma cópia multipart](#)
- [Realizar um carregamento fracionado](#)
- [Usar somas de verificação](#)

CLI

AWS CLI

O seguinte comando conclui um upload fracionado da chave multipart/01 no bucket my-bucket:

```
aws s3api complete-multipart-upload --multipart-upload file://
mpustruct --bucket my-bucket --key 'multipart/01' --upload-id
dfRtDYU0WWCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3U
```

O ID de upload exigido por esse comando é a saída de create-multipart-upload e também pode ser recuperado com list-multipart-uploads.

A opção de upload fracionado no comando acima usa uma estrutura JSON que descreve quais partes do upload fracionado devem ser reagrupadas no arquivo completo. Neste exemplo, o prefixo file:// é usado para carregar a estrutura JSON de um arquivo na pasta local chamada mpustruct.

mpustruct:

```
{
  "Parts": [
    {
      "ETag": "e868e0f4719e394144ef36531ee6824c",
      "PartNumber": 1
    },
    {
```

```

    "ETag": "6bb2b12753d66fe86da4998aa33ffffb0",
    "PartNumber": 2
  },
  {
    "ETag": "d0a0112e841abec9c9ec83406f0159c8",
    "PartNumber": 3
  }
]
}

```

O valor de ETag de cada parte carregada é gerado sempre que você fizer o upload de uma parte usando o comando `upload-part`. Ele também pode ser recuperado ao chamar `list-parts` ou calculado fazendo a soma de verificação MD5 de cada parte.

Saída:

```

{
  "ETag": "\"3944a9f7a4faab7f78788ff6210f63f0-3\"",
  "Bucket": "my-bucket",
  "Location": "https://my-bucket.s3.amazonaws.com/multipart%2F01",
  "Key": "multipart/01"
}

```

- Para obter detalhes da API, consulte [CompleteMultipartUpload](#) na Referência de comandos da AWS CLI.

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```

let _complete_multipart_upload_res = client
    .complete_multipart_upload()
    .bucket(&bucket_name)
    .key(&key)
    .multipart_upload(completed_multipart_upload)

```

```
.upload_id(upload_id)
.send()
.await
.unwrap();
```

- Para obter detalhes da API, consulte [CompleteMultipartUpload](#) na Referência da API do AWS SDK para Rust.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **CopyObject** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o CopyObject.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- [Conceitos básicos de buckets e objetos](#)
- [Conceitos básicos de criptografia](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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

```
public class CopyObject
{
    public static async Task Main()
    {
        // Specify the AWS Region where your buckets are located if it is
        // different from the AWS Region of the default user.
        IAmazonS3 s3Client = new AmazonS3Client();

        // Remember to change these values to refer to your Amazon S3
objects.
        string sourceBucketName = "doc-example-bucket1";
        string destinationBucketName = "doc-example-bucket2";
        string sourceObjectKey = "testfile.txt";
        string destinationObjectKey = "testfilecopy.txt";

        Console.WriteLine($"Copying {sourceObjectKey} from {sourceBucketName}
to ");
        Console.WriteLine($"{destinationBucketName} as
{destinationObjectKey}");

        var response = await CopyingObjectAsync(
            s3Client,
            sourceObjectKey,
            destinationObjectKey,
            sourceBucketName,
            destinationBucketName);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine("\nCopy complete.");
        }
    }

    /// <summary>
    /// This method calls the AWS SDK for .NET to copy an
    /// object from one Amazon S3 bucket to another.
    /// </summary>
    /// <param name="client">The Amazon S3 client object.</param>
    /// <param name="sourceKey">The name of the object to be copied.</param>
    /// <param name="destinationKey">The name under which to save the copy.</
param>
    /// <param name="sourceBucketName">The name of the Amazon S3 bucket
    /// where the file is located now.</param>
    /// <param name="destinationBucketName">The name of the Amazon S3
```

```
/// bucket where the copy should be saved.</param>
/// <returns>Returns a CopyObjectResponse object with the results from
/// the async call.</returns>
public static async Task<CopyObjectResponse> CopyingObjectAsync(
    IAmazonS3 client,
    string sourceKey,
    string destinationKey,
    string sourceBucketName,
    string destinationBucketName)
{
    var response = new CopyObjectResponse();
    try
    {
        var request = new CopyObjectRequest
        {
            SourceBucket = sourceBucketName,
            SourceKey = sourceKey,
            DestinationBucket = destinationBucketName,
            DestinationKey = destinationKey,
        };
        response = await client.CopyObjectAsync(request);
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error copying object: '{ex.Message}'");
    }

    return response;
}
}
```

- Para obter detalhes da API, consulte [CopyObject](#) em Referência da API AWS SDK for .NET.

Bash

AWS CLI com script Bash

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function copy_item_in_bucket
#
# This function creates a copy of the specified file in the same bucket.
#
# Parameters:
#     $1 - The name of the bucket to copy the file from and to.
#     $2 - The key of the source file to copy.
#     $3 - The key of the destination file.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function copy_item_in_bucket() {
    local bucket_name=$1
    local source_key=$2
    local destination_key=$3
    local response

    response=$(aws s3api copy-object \
        --bucket "$bucket_name" \
        --copy-source "$bucket_name/$source_key" \
```

```
--key "$destination_key")

# shellcheck disable=SC2181
if [[ $? -ne 0 ]]; then
    errecho "ERROR: AWS reports s3api copy-object operation failed.\n$response"
    return 1
fi
}
```

- Para obter detalhes da API, consulte [CopyObject](#) na Referência de comandos da AWS CLI.

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
bool AwsDoc::S3::CopyObject(const Aws::String &objectKey, const Aws::String
&fromBucket, const Aws::String &toBucket,
                           const Aws::Client::ClientConfiguration &clientConfig)
{
    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 Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: CopyObject: " <<
                  err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;

    }
    else {
```

```
        std::cout << "Successfully copied " << objectKey << " from " <<
fromBucket <<
            " to " << toBucket << "." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Para obter detalhes da API, consulte [CopyObject](#) em Referência da API AWS SDK for C++.

CLI

AWS CLI

O comando a seguir copia um objeto de bucket-1 para bucket-2:

```
aws s3api copy-object --copy-source bucket-1/test.txt --key test.txt --bucket
bucket-2
```

Saída:

```
{
  "CopyObjectResult": {
    "LastModified": "2015-11-10T01:07:25.000Z",
    "ETag": "\"589c8b79c230a6ecd5a7e1d040a9a030\""
  },
  "VersionId": "YdnYvTCVDqRRFA.NFJjy36p0hxifM1kA"
}
```

- Para obter detalhes da API, consulte [CopyObject](#) na Referência de comandos da AWS CLI.

Go

SDK para Go V2

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).


```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
// actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// CopyToBucket copies an object in a bucket to another bucket.
func (basics BucketBasics) CopyToBucket(sourceBucket string, destinationBucket
string, objectKey string) error {
    _, err := basics.S3Client.CopyObject(context.TODO(), &s3.CopyObjectInput{
        Bucket:      aws.String(destinationBucket),
        CopySource:  aws.String(fmt.Sprintf("%v/%v", sourceBucket, objectKey)),
        Key:         aws.String(objectKey),
    })
    if err != nil {
        log.Printf("Couldn't copy object from %v:%v to %v:%v. Here's why: %v\n",
            sourceBucket, objectKey, destinationBucket, objectKey, err)
    }
    return err
}
```

- Para obter detalhes da API, consulte [CopyObject](#) em Referência da API AWS SDK for Go.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Copie um objeto usando um [S3Client](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.CopyObjectRequest;
import software.amazon.awssdk.services.s3.model.CopyObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;

/**
 * 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 CopyObject {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <objectKey> <fromBucket> <toBucket>

            Where:
                objectKey - The name of the object (for example, book.pdf).
                fromBucket - The S3 bucket name that contains the object (for
                example, bucket1).
                toBucket - The S3 bucket to copy the object to (for example,
                bucket2).

            """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String objectKey = args[0];
        String fromBucket = args[1];
        String toBucket = args[2];
        System.out.format("Copying object %s from bucket %s to %s\n", objectKey,
            fromBucket, toBucket);
        Region region = Region.US_EAST_1;
```

```
S3Client s3 = S3Client.builder()
    .region(region)
    .build();

copyBucketObject(s3, fromBucket, objectKey, toBucket);
s3.close();
}

public static String copyBucketObject(S3Client s3, String fromBucket, String
objectKey, String toBucket) {
    CopyObjectRequest copyReq = CopyObjectRequest.builder()
        .sourceBucket(fromBucket)
        .sourceKey(objectKey)
        .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 "";
}
}
```

Use um [S3TransferManager](#) para [copiar um objeto](#) de um bucket para outro. Veja o [arquivo completo](#) e [teste](#).

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.services.s3.model.CopyObjectRequest;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedCopy;
import software.amazon.awssdk.transfer.s3.model.Copy;
import software.amazon.awssdk.transfer.s3.model.CopyRequest;

import java.util.UUID;
```

```
public String copyObject(S3TransferManager transferManager, String
bucketName,
    String key, String destinationBucket, String destinationKey) {
    CopyObjectRequest copyObjectRequest = CopyObjectRequest.builder()
        .sourceBucket(bucketName)
        .sourceKey(key)
        .destinationBucket(destinationBucket)
        .destinationKey(destinationKey)
        .build();

    CopyRequest copyRequest = CopyRequest.builder()
        .copyObjectRequest(copyObjectRequest)
        .build();

    Copy copy = transferManager.copy(copyRequest);

    CompletedCopy completedCopy = copy.completionFuture().join();
    return completedCopy.response().copyObjectResult().eTag();
}
```

- Para obter detalhes da API, consulte [CopyObject](#) em Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Copie o objeto.

```
import { S3Client, CopyObjectCommand } from "@aws-sdk/client-s3";

const client = new S3Client({});
```

```
export const main = async () => {
  const command = new CopyObjectCommand({
    CopySource: "SOURCE_BUCKET/SOURCE_OBJECT_KEY",
    Bucket: "DESTINATION_BUCKET",
    Key: "NEW_OBJECT_KEY",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Para obter detalhes da API, consulte [CopyObject](#) em Referência da API AWS SDK for JavaScript.

Kotlin

SDK for Kotlin

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Kotlin.

PHP

SDK para PHP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Cópia simples de um objeto.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

try {
    $folder = "copied-folder";
    $this->s3client->copyObject([
        'Bucket' => $this->bucketName,
        'CopySource' => "$this->bucketName/$fileName",
        'Key' => "$folder/$fileName-copy",
    ]);
    echo "Copied $fileName to $folder/$fileName-copy.\n";
} catch (Exception $exception) {
    echo "Failed to copy $fileName with error: " . $exception-
    >getMessage();
    exit("Please fix error with object copying before continuing.");
}
```

- Para obter detalhes da API, consulte [CopyObject](#) em Referência da API AWS SDK for PHP.

PowerShell

Tools for PowerShell

Exemplo 1: este comando copia o objeto “sample.txt” do bucket “test-files” para o mesmo bucket, mas com uma nova chave de “sample-copy.txt”.

```
Copy-S3Object -BucketName test-files -Key sample.txt -DestinationKey sample-copy.txt
```

Exemplo 2: este comando copia o objeto “sample.txt” do bucket “test-files” para o bucket “backup-files” com uma nova chave de “sample-copy.txt”.

```
Copy-S3Object -BucketName test-files -Key sample.txt -DestinationKey sample-copy.txt -DestinationBucket backup-files
```

Exemplo 3: este comando baixa o objeto “sample.txt” do bucket “test-files” em um arquivo local com o nome “local-sample.txt”.

```
Copy-S3Object -BucketName test-files -Key sample.txt -LocalFile local-sample.txt
```

Exemplo 4: um único objeto é baixado no arquivo especificado. O arquivo baixado encontra-se em c:\downloads\data\archive.zip.

```
Copy-S3Object -BucketName test-files -Key data/archive.zip -LocalFolder c:\downloads
```

Exemplo 5: todos os objetos que correspondem ao prefixo de chave especificado são baixados na pasta local. A hierarquia relativa de chaves será preservada como subpastas no local geral do download.

```
Copy-S3Object -BucketName test-files -KeyPrefix data -LocalFolder c:\downloads
```

- Para obter detalhes da API, consulte [CopyObject](#) na AWS Tools for PowerShell Cmdlet Reference.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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
end

```

```

# Copy the source object to the specified target bucket and rename it with the
target key.
#
# @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
object is copied.
# @param target_object_key [String] The key to give the copy of the object.
# @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
nil.
def copy_object(target_bucket, target_object_key)
  @source_object.copy_to(bucket: target_bucket.name, key: target_object_key)
  target_bucket.object(target_object_key)
rescue Aws::Errors::ServiceError => e
  puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's
why: #{e.message}"
end
end

# Example usage:
def run_demo
  source_bucket_name = "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 do 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

# Example usage:
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 da API AWS SDK for Ruby.

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
pub async fn copy_object(
    client: &Client,
    bucket_name: &str,
    object_key: &str,
    target_key: &str,
) -> Result<CopyObjectOutput, SdkError<CopyObjectError>> {
    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
}
```

- Para obter detalhes da API, consulte [CopyObject](#) em Referência da API AWS SDK para Rust.

SAP ABAP

SDK para SAP ABAP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
TRY.  
  lo_s3->copyobject(  
    iv_bucket = iv_dest_bucket  
    iv_key = iv_dest_object  
    iv_copysource = |{ iv_src_bucket }/{ iv_src_object }|  
  ).  
  MESSAGE 'Object copied to another bucket.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' TYPE 'E'.  
CATCH /aws1/cx_s3_nosuchkey.  
  MESSAGE 'Object key does not exist.' TYPE 'E'.  
ENDTRY.
```

- Para obter detalhes da API, consulte [CopyObject](#) na Referência da API AWS SDK para SAP ABAP.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Swift.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **CreateBucket** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o CreateBucket.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- [Conceitos básicos de buckets e objetos](#)
- [Trabalhar com objetos versionados](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <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;
    }
}
```

Crie um bucket com o bloqueio de objetos habilitado.

```
/// <summary>
/// Create a new Amazon S3 bucket with object lock actions.
/// </summary>
/// <param name="bucketName">The name of the bucket to create.</param>
/// <param name="enableObjectLock">True to enable object lock on the
bucket.</param>
/// <returns>True if successful.</returns>
public async Task<bool> CreateBucketWithObjectLock(string bucketName, bool
enableObjectLock)
{
    Console.WriteLine($"\\tCreating bucket {bucketName} with object lock
{enableObjectLock}.");
    try
    {
        var request = new PutBucketRequest
        {
            BucketName = bucketName,
            UseClientRegion = true,
            ObjectLockEnabledForBucket = enableObjectLock,
        };

        var response = await _amazonS3.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 AWS SDK for .NET.

Bash

AWS CLI com script Bash

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
#####  
# function iecho  
#  
# This function enables the script to display the specified text only if  
# the global variable $VERBOSE is set to true.  
#####  
function iecho() {  
    if [[ $VERBOSE == true ]]; then  
        echo "$@"  
    fi  
}  
  
#####  
# function errecho  
#  
# This function outputs everything sent to it to STDERR (standard error output).  
#####  
function errecho() {  
    printf "%s\n" "$*" 1>&2  
}  
  
#####  
# function create-bucket  
#  
# This function creates the specified bucket in the specified AWS Region, unless  
# it already exists.  
#  
# Parameters:  
#     -b bucket_name  -- The name of the bucket to create.  
#     -r region_code  -- The code for an AWS Region in which to  
#                       create the bucket.  
#
```

```
# Returns:
#     The URL of the bucket that was created.
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
function create_bucket() {
    local bucket_name region_code response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function create_bucket"
        echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
        echo "  -b bucket_name    The name of the bucket. It must be globally
unique."
        echo "  [-r region_code]  The code for an AWS Region in which the bucket is
created."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "b:r:h" option; do
        case "${option}" in
            b) bucket_name="${OPTARG}" ;;
            r) region_code="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done

    if [[ -z "$bucket_name" ]]; then
        errecho "ERROR: You must provide a bucket name with the -b parameter."
        usage
        return 1
    fi
}
```

```
local bucket_config_arg
# A location constraint for "us-east-1" returns an error.
if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then
    bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
fi

iecho "Parameters:\n"
iecho "    Bucket name:  $bucket_name"
iecho "    Region code:  $region_code"
iecho ""

# If the bucket already exists, we don't want to try to create it.
if (bucket_exists "$bucket_name"); then
    errecho "ERROR: A bucket with that name already exists. Try again."
    return 1
fi


# shellcheck disable=SC2086
response=$(aws s3api create-bucket \
    --bucket "$bucket_name" \
    $bucket_config_arg)

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
fi
}
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência de comandos da AWS CLI.

C++

SDK para C++

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
bool AwsDoc::S3::CreateBucket(const Aws::String &bucketName,
                              const Aws::Client::ClientConfiguration
&clientConfig) {
    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::CreateBucketRequest request;
    request.SetBucket(bucketName);

    //TODO(user): Change the bucket location constraint enum to your target
    Region.
    if (clientConfig.region != "us-east-1") {
        Aws::S3::Model::CreateBucketConfiguration createBucketConfig;
        createBucketConfig.SetLocationConstraint(

    Aws::S3::Model::BucketLocationConstraintMapper::GetBucketLocationConstraintForName(
        clientConfig.region));
        request.SetCreateBucketConfiguration(createBucketConfig);
    }

    Aws::S3::Model::CreateBucketOutcome outcome = client.CreateBucket(request);
    if (!outcome.IsSuccess()) {
        auto err = outcome.GetError();
        std::cerr << "Error: CreateBucket: " <<
            err.GetExceptionName() << ": " << err.GetMessage() <<
    std::endl;
    }
    else {
        std::cout << "Created bucket " << bucketName <<
            " in the specified AWS Region." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API AWS SDK for C+
+.

CLI

AWS CLI

Exemplo 1: como criar um bucket

O seguinte exemplo de `create-bucket` cria um bucket chamado `my-bucket`:

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region us-east-1
```

Saída:

```
{  
  "Location": "/my-bucket"  
}
```

Para obter mais informações, consulte [Como criar um bucket](#) no Guia do usuário do Amazon S3.

Exemplo 2: como criar um bucket com aplicação de políticas de proprietário

O exemplo de `create-bucket` a seguir cria um bucket chamado `my-bucket` que usa a configuração aplicada de proprietário de bucket para a propriedade de objetos do S3.

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region us-east-1 \  
  --object-ownership BucketOwnerEnforced
```

Saída:

```
{  
  "Location": "/my-bucket"  
}
```

Para obter mais informações, consulte [Controlar a propriedade de objetos e desabilitar ACLs](#), no Guia do usuário do Amazon S3.

Exemplo 3: como criar um bucket fora da região ``us-east-1``

O exemplo de `create-bucket` a seguir cria um bucket chamado `my-bucket` na região `eu-west-1`. Regiões fora da `us-east-1` exigem que o `LocationConstraint` adequado seja especificado para poder criar o bucket na região desejada.

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region eu-west-1 \  
  --create-bucket-configuration LocationConstraint=eu-west-1
```

Saída:


```
{  
  "Location": "http://my-bucket.s3.amazonaws.com/"  
}
```

Para obter mais informações, consulte [Como criar um bucket](#) no Guia do usuário do Amazon S3.

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência de comandos da AWS CLI.

Go

SDK para Go V2

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)  
actions  
// used in the examples.
```

```
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// CreateBucket creates a bucket with the specified name in the specified Region.
func (basics BucketBasics) CreateBucket(name string, region string) error {
    _, err := basics.S3Client.CreateBucket(context.TODO(), &s3.CreateBucketInput{
        Bucket: aws.String(name),
        CreateBucketConfiguration: &types.CreateBucketConfiguration{
            LocationConstraint: types.BucketLocationConstraint(region),
        },
    })
    if err != nil {
        log.Printf("Couldn't create bucket %v in Region %v. Here's why: %v\n",
            name, region, err)
    }
    return err
}
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API AWS SDK for Go.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Criará um bucket.

```
import software.amazon.awssdk.core.waiters.WaiterResponse;
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.CreateBucketRequest;
import software.amazon.awssdk.services.s3.model.HeadBucketRequest;
import software.amazon.awssdk.services.s3.model.HeadBucketResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.waiters.S3Waiter;
import java.net.URISyntaxException;

/**
 * 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 CreateBucket {
    public static void main(String[] args) throws URISyntaxException {
        final String usage = ""

            Usage:
                <bucketName>\s

            Where:
                bucketName - The name of the bucket to create. The bucket
name must be unique, or an error occurs.
            """;

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

        String bucketName = args[0];
        System.out.format("Creating a bucket named %s\n", bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        createBucket(s3, bucketName);
        s3.close();
    }
}
```



```
}

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);
    }
}
}
```

Crie um bucket com o bloqueio de objetos habilitado.

```
// Create a new Amazon S3 bucket with object lock options.
public void createBucketWithLockOptions(boolean enableObjectLock, String
bucketName) {
    S3Waiter s3Waiter = getClient().waiter();
    CreateBucketRequest bucketRequest = CreateBucketRequest.builder()
        .bucket(bucketName)
        .objectLockEnabledForBucket(enableObjectLock)
        .build();

    getClient().createBucket(bucketRequest);
    HeadBucketRequest bucketRequestWait = HeadBucketRequest.builder()
        .bucket(bucketName)
        .build();
}
```

```
// Wait until the bucket is created and print out the response.
s3Waiter.waitUntilBucketExists(bucketRequestWait);
System.out.println(bucketName + " is ready");
}
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Crie o bucket.

```
import { CreateBucketCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new CreateBucketCommand({
    // The name of the bucket. Bucket names are unique and have several other
    // constraints.
    // See https://docs.aws.amazon.com/AmazonS3/latest/userguide/
    // bucketnamingrules.html
    Bucket: "bucket-name",
  });

  try {
    const { Location } = await client.send(command);
    console.log(`Bucket created with location ${Location}`);
  } catch (err) {
    console.error(err);
  }
}
```

```
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).
- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API AWS SDK for JavaScript.

Kotlin

SDK for Kotlin

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Kotlin.

PHP

SDK para PHP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Criará um bucket.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

try {
    $this->s3client->createBucket([
        'Bucket' => $this->bucketName,
        'CreateBucketConfiguration' => ['LocationConstraint' => $region],
    ]);
    echo "Created bucket named: $this->bucketName \n";
} catch (Exception $exception) {
    echo "Failed to create bucket $this->bucketName with error: " .
    $exception->getMessage();
    exit("Please fix error with bucket creation before continuing.");
}
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API AWS SDK for PHP.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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 AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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
end

# Example usage:
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 AWS SDK for Ruby.

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```

pub async fn create_bucket(
  client: &Client,
  bucket_name: &str,
  region: &str,
) -> Result<CreateBucketOutput, SdkError<CreateBucketError>> {
  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
}
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API AWS SDK para Rust.

SAP ABAP

SDK para SAP ABAP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
TRY.
  lo_s3->createbucket(
    iv_bucket = iv_bucket_name
  ).
  MESSAGE 'S3 bucket created.' TYPE 'I'.
CATCH /aws1/cx_s3_bucketalrddyexists.
  MESSAGE 'Bucket name already exists.' TYPE 'E'.
CATCH /aws1/cx_s3_bktalrddyownedbyyou.
  MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
ENDTRY.
```

- Para obter detalhes da API, consulte [CreateBucket](#) na Referência da API AWS SDK para SAP ABAP.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Swift.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **CreateMultiRegionAccessPoint** com um AWS SDK ou ferramenta de linha de comando

O código de exemplo a seguir mostra como usar `CreateMultiRegionAccessPoint`.

Kotlin

SDK for Kotlin

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Configure o cliente de controle do S3 para enviar a solicitação à região us-west-2.

```
suspend fun createS3ControlClient(): S3ControlClient {
    // Configure your S3ControlClient to send requests to US West
    (Oregon).
    val s3Control = S3ControlClient.fromEnvironment {
        region = "us-west-2"
    }
    return s3Control
}
```

Crie o ponto de acesso multirregional.

```
suspend fun createMrap(s3Control: S3ControlClient, accountIdParam: String,
    bucketName1: String, bucketName2: String, mrapName: String): String {
    println("Creating MRAP ...")
    val createMrapResponse: CreateMultiRegionAccessPointResponse =
    s3Control.createMultiRegionAccessPoint {
        accountId = accountIdParam
        clientToken = UUID.randomUUID().toString()
        details {
            name = mrapName

            regions = listOf(
                Region {
                    bucket = bucketName1
                },
                Region {
                    bucket = bucketName2
                }
            )
        }
    }
}
```

```

    }
  }
  val requestToken: String? = createMrapResponse.requestTokenArn

  // Use the request token to check for the status of the
  CreateMultiRegionAccessPoint operation.
  if (requestToken != null) {
    waitForSucceededStatus(s3Control, requestToken, accountIdParam)
    println("MRAP created")
  }

  val getMrapResponse = s3Control.getMultiRegionAccessPoint(
    input = GetMultiRegionAccessPointRequest {
      accountId = accountIdParam
      name = mrapName
    }
  )
  val mrapAlias = getMrapResponse.accessPoint?.alias
  return "arn:aws:s3:::$accountIdParam:accesspoint/$mrpAlias"
}

```

Aguarde até que o ponto de acesso multirregional fique disponível.

```

suspend fun waitForSucceededStatus(s3Control: S3ControlClient,
requestToken: String, accountIdParam: String, timeBetweenChecks: Duration =
1.minutes) {
  var describeResponse: DescribeMultiRegionAccessPointOperationResponse
  describeResponse = s3Control.describeMultiRegionAccessPointOperation(
    input = DescribeMultiRegionAccessPointOperationRequest {
      accountId = accountIdParam
      requestTokenArn = requestToken
    }
  )

  var status: String? = describeResponse.asyncOperation?.requestStatus
  while (status != "SUCCEEDED") {
    delay(timeBetweenChecks)
    describeResponse =
s3Control.describeMultiRegionAccessPointOperation(
      input = DescribeMultiRegionAccessPointOperationRequest {
        accountId = accountIdParam
        requestTokenArn = requestToken

```

```
        }
    )
    status = describeResponse.asyncOperation?.requestStatus
    println(status)
}
}
```

- Para ter mais informações, consulte o [Guia do desenvolvedor do AWS SDK para Kotlin](#).
- Para obter detalhes da API, consulte [CreateMultiRegionAccessPoint](#) na Referência da API do AWS SDK para Kotlin.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **CreateMultipartUpload** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `CreateMultipartUpload`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- [Executar uma cópia multipart](#)
- [Realizar um carregamento fracionado](#)
- [Usar somas de verificação](#)

CLI

AWS CLI

O seguinte comando cria um upload fracionado no bucket `my-bucket` com a chave `multipart/01`:

```
aws s3api create-multipart-upload --bucket my-bucket --key 'multipart/01'
```

Saída:

```
{
  "Bucket": "my-bucket",
  "UploadId":
  "dfRtDYU0WCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZ1jF.Yxwh6XG7WfS2vC4to6HiV6Yj1x.cph0gtNBtJ8P3
  "Key": "multipart/01"
}
```

O arquivo concluído terá o nome 01 e estará em uma pasta chamada multipart no bucket my-bucket. Salve o ID de upload, a chave e o nome do bucket para usar com o comando upload-part.

- Para obter detalhes da API, consulte [CreateMultipartUpload](#) na Referência de comandos da AWS CLI.

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
let multipart_upload_res: CreateMultipartUploadOutput = client
    .create_multipart_upload()
    .bucket(&bucket_name)
    .key(&key)
    .send()
    .await
    .unwrap();
```

- Para obter detalhes da API, consulte [CreateMultipartUpload](#) na Referência da API do AWS SDK para Rust.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **DeleteBucket** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o DeleteBucket.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Conceitos básicos de buckets e objetos](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <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 AWS SDK for .NET.

Bash

AWS CLI com script Bash

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function delete_bucket
#
# This function deletes the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_bucket() {
    local bucket_name=$1
    local response

    response=$(aws s3api delete-bucket \
```

```
--bucket "$bucket_name")

# shellcheck disable=SC2181
if [[ $? -ne 0 ]]; then
    errecho "ERROR: AWS reports s3api delete-bucket failed.\n$response"
    return 1
fi
}
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência de comandos da AWS CLI.

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
bool AwsDoc::S3::DeleteBucket(const Aws::String &bucketName,
                              const Aws::Client::ClientConfiguration
&clientConfig) {

    Aws::S3::S3Client client(clientConfig);

    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;
    }
}
```

```
    else {
        std::cout << "The bucket was deleted" << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

O seguinte comando exclui o bucket `my-bucket`:

```
aws s3api delete-bucket --bucket my-bucket --region us-east-1
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência de comandos da AWS CLI.

Go

SDK para Go V2

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
// actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
type BucketBasics struct {
```

```
S3Client *s3.Client
}

// DeleteBucket deletes a bucket. The bucket must be empty or an error is
// returned.
func (basics BucketBasics) DeleteBucket(bucketName string) error {
    _, err := basics.S3Client.DeleteBucket(context.TODO(), &s3.DeleteBucketInput{
        Bucket: aws.String(bucketName)})
    if err != nil {
        log.Printf("Couldn't delete bucket %v. Here's why: %v\n", bucketName, err)
    }
    return err
}
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API AWS SDK for Go.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
DeleteBucketRequest deleteBucketRequest = DeleteBucketRequest.builder()
    .bucket(bucket)
    .build();

s3.deleteBucket(deleteBucketRequest);
s3.close();
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Excluir o bucket.

```
import { DeleteBucketCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// Delete a bucket.
export const main = async () => {
  const command = new DeleteBucketCommand({
    Bucket: "test-bucket",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).
- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API AWS SDK for JavaScript.

PHP

SDK para PHP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Exclua um bucket vazio.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

try {
    $this->s3client->deleteBucket([
        'Bucket' => $this->bucketName,
    ]);
    echo "Deleted bucket $this->bucketName.\n";
} catch (Exception $exception) {
    echo "Failed to delete $this->bucketName with error: " . $exception-
    >getMessage();
    exit("Please fix error with bucket deletion before continuing.");
}
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API AWS SDK for PHP.

PowerShell

Tools for PowerShell

Exemplo 1: este comando remove todos os objetos e versões de objetos do bucket “test-files” e, em seguida, exclui o bucket. O comando solicitará a confirmação antes de continuar. Adicione a opção `-Force` para ignorar a confirmação. Observe que os buckets que não estão vazios não podem ser excluídos.

```
Remove-S3Bucket -BucketName test-files -DeleteBucketContent
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na AWS Tools for PowerShell Cmdlet Reference.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
# 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 AWS SDK for Ruby.

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Rust.

SAP ABAP

SDK para SAP ABAP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
TRY.

    lo_s3->deletebucket(
        iv_bucket = iv_bucket_name
    ).
    MESSAGE 'Deleted S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
```

```
ENDTRY.
```

- Para obter detalhes da API, consulte [DeleteBucket](#) na Referência da API AWS SDK para SAP ABAP.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Swift.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar `DeleteBucketAnalyticsConfiguration` com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `DeleteBucketAnalyticsConfiguration`.

CLI

AWS CLI

Para excluir uma configuração de análise de um bucket

O exemplo `delete-bucket-analytics-configuration` a seguir recupera a configuração referente ao bucket e ID especificados.

```
aws s3api delete-bucket-analytics-configuration \  
  --bucket my-bucket \  
  --id 1
```

Este comando não produz saída.

- Para obter detalhes da API, consulte [DeleteBucketAnalyticsConfiguration](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: o comando remove o filtro de análise com o nome “testfilter” no bucket do S3 em questão.

```
Remove-S3BucketAnalyticsConfiguration -BucketName 's3testbucket' -AnalyticsId  
'testfilter'
```

- Para obter detalhes da API, consulte [DeleteBucketAnalyticsConfiguration](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **DeleteBucketCors** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o DeleteBucketCors.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <summary>
/// Deletes a CORS configuration from an Amazon S3 bucket.
/// </summary>
/// <param name="client">The initialized Amazon S3 client object used
/// to delete the CORS configuration from the bucket.</param>
private static async Task DeleteCORSConfigurationAsync(AmazonS3Client
client)
{
    DeleteCORSConfigurationRequest request = new
DeleteCORSConfigurationRequest()
    {
        BucketName = BucketName,
    };
    await client.DeleteCORSConfigurationAsync(request);
}
```

- Para obter detalhes da API, consulte [DeleteBucketCors](#) na Referência da API AWS SDK for .NET.

CLI

AWS CLI

O seguinte comando exclui a configuração de compartilhamento de recursos de origem cruzada do bucket `my-bucket`:

```
aws s3api delete-bucket-cors --bucket my-bucket
```

- Para obter detalhes da API, consulte [DeleteBucketCors](#) na Referência de comandos da AWS CLI.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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
  end
end
```

```
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 AWS SDK for Ruby.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **DeleteBucketEncryption** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o DeleteBucketEncryption.

CLI

AWS CLI

Para excluir a configuração de criptografia do lado do servidor de um bucket

O exemplo `delete-bucket-encryption` a seguir exclui a configuração de criptografia do lado do servidor do bucket especificado.

```
aws s3api delete-bucket-encryption \
  --bucket my-bucket
```

Este comando não produz saída.

- Para obter detalhes da API, consulte [DeleteBucketEncryption](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: isso desabilita a criptografia habilitada para o bucket do S3 fornecido.

```
Remove-S3BucketEncryption -BucketName 's3casetestbucket'
```

Saída:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-S3BucketEncryption (DeleteBucketEncryption)" on
target "s3casetestbucket".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"): Y
```

- Para obter detalhes da API, consulte [DeleteBucketEncryption](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **DeleteBucketInventoryConfiguration** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `DeleteBucketInventoryConfiguration`.

CLI

AWS CLI

Para excluir a configuração de inventário de um bucket

O exemplo `delete-bucket-inventory-configuration` a seguir exclui a configuração do inventário com ID 1 do bucket especificado.

```
aws s3api delete-bucket-inventory-configuration \
```



```
--bucket my-bucket \  
--id 1
```

Este comando não produz saída.

- Para obter detalhes da API, consulte [DeleteBucketInventoryConfiguration](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando remove o inventário denominado “testInventoryName” correspondente ao bucket do S3 em questão.

```
Remove-S3BucketInventoryConfiguration -BucketName 's3testbucket' -InventoryId  
'testInventoryName'
```

Saída:

```
Confirm  
Are you sure you want to perform this action?  
Performing the operation "Remove-S3BucketInventoryConfiguration  
(DeleteBucketInventoryConfiguration)" on target "s3testbucket".  
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is  
"Y"): Y
```

- Para obter detalhes da API, consulte [DeleteBucketInventoryConfiguration](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **DeleteBucketLifecycle** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `DeleteBucketLifecycle`.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <summary>
/// This method removes the Lifecycle configuration from the named
/// S3 bucket.
/// </summary>
/// <param name="client">The S3 client object used to call
/// the RemoveLifecycleConfigAsync method.</param>
/// <param name="bucketName">A string representing the name of the
/// S3 bucket from which the configuration will be removed.</param>
public static async Task RemoveLifecycleConfigAsync(IAmazonS3 client,
string bucketName)
{
    var request = new DeleteLifecycleConfigurationRequest()
    {
        BucketName = bucketName,
    };
    await client.DeleteLifecycleConfigurationAsync(request);
}
```

- Para obter detalhes da API, consulte [DeleteBucketLifecycle](#) na Referência da API AWS SDK for .NET.

CLI

AWS CLI

O seguinte comando exclui a configuração de ciclo de vida do bucket my-bucket:

```
aws s3api delete-bucket-lifecycle --bucket my-bucket
```

- Para obter detalhes da API, consulte [DeleteBucketLifecycle](#) na Referência de comandos da AWS CLI.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **DeleteBucketMetricsConfiguration** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o DeleteBucketMetricsConfiguration.

CLI

AWS CLI

Para excluir uma configuração de métricas referente a um bucket

O exemplo delete-bucket-metrics-configuration a seguir remove a configuração referente ao bucket e ID especificados.

```
aws s3api delete-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123
```

Este comando não produz saída.

- Para obter detalhes da API, consulte [DeleteBucketMetricsConfiguration](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: o comando remove o filtro de métricas com o nome “testmetrics” no bucket do S3 em questão.

```
Remove-S3BucketMetricsConfiguration -BucketName 's3testbucket' -MetricsId
'testmetrics'
```

- Para obter detalhes da API, consulte [DeleteBucketMetricsConfiguration](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **DeleteBucketPolicy** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o DeleteBucketPolicy.

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
bool AwsDoc::S3::DeleteBucketPolicy(const Aws::String &bucketName,
                                    const Aws::Client::ClientConfiguration
&clientConfig) {
    Aws::S3::S3Client client(clientConfig);

    Aws::S3::Model::DeleteBucketPolicyRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::DeleteBucketPolicyOutcome outcome =
    client.DeleteBucketPolicy(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: DeleteBucketPolicy: " <<
```

```
        err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    }
    else {
        std::cout << "Policy was deleted from the bucket." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Para obter detalhes da API, consulte [DeleteBucketPolicy](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

O seguinte comando exclui uma política de bucket do bucket my-bucket:

```
aws s3api delete-bucket-policy --bucket my-bucket
```

- Para obter detalhes da API, consulte [DeleteBucketPolicy](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.DeleteBucketPolicyRequest;
```

```
/**
 * 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 DeleteBucketPolicy {
    public static void main(String[] args) {

        final String usage = ""

            Usage:
                <bucketName>

            Where:
                bucketName - The Amazon S3 bucket to delete the policy from
(for example, bucket1).""";

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

        String bucketName = args[0];
        System.out.format("Deleting policy from bucket: \"%s\"\n\n", bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        deleteS3BucketPolicy(s3, bucketName);
        s3.close();
    }

    // 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 AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Exclua a política de bucket.

```
import { DeleteBucketPolicyCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// This will remove the policy from the bucket.
export const main = async () => {
    const command = new DeleteBucketPolicyCommand({
        Bucket: "test-bucket",
    });

    try {
        const response = await client.send(command);
        console.log(response);
    } catch (err) {
        console.error(err);
    }
}
```



```
}  
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).
- Para obter detalhes da API, consulte [DeleteBucketPolicy](#) na Referência da API AWS SDK for JavaScript.

Kotlin

SDK for Kotlin

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Kotlin.

PowerShell

Tools for PowerShell

Exemplo 1: o comando remove a política de bucket associada ao bucket do S3 em questão.

```
Remove-S3BucketPolicy -BucketName 's3testbucket'
```

- Para obter detalhes da API, consulte [DeleteBucketPolicy](#) na AWS Tools for PowerShell Cmdlet Reference.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
# 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

end
```

- Para obter detalhes da API, consulte [DeleteBucketPolicy](#) na Referência da API AWS SDK for Ruby.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **DeleteBucketReplication** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o DeleteBucketReplication.

CLI

AWS CLI

O seguinte comando exclui uma configuração de replicação de um bucket denominado my-bucket:

```
aws s3api delete-bucket-replication --bucket my-bucket
```

- Para obter detalhes da API, consulte [DeleteBucketReplication](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: exclui a configuração de replicação associada ao bucket denominado “mybucket”. Observe que essa operação exige permissão para a ação s3:DeleteReplicationConfiguration. Será solicitada uma confirmação antes que a operação continue. Para ignorar a confirmação, use a opção -Force.

```
Remove-S3BucketReplication -BucketName mybucket
```

- Para obter detalhes da API, consulte [DeleteBucketReplication](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **DeleteBucketTagging** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o DeleteBucketTagging.

CLI

AWS CLI

O seguinte comando exclui uma configuração de marcação de um bucket denominado my-bucket:

```
aws s3api delete-bucket-tagging --bucket my-bucket
```

- Para obter detalhes da API, consulte [DeleteBucketTagging](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando remove todas as tags associadas ao bucket do S3 em questão.

```
Remove-S3BucketTagging -BucketName 's3testbucket'
```

Saída:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-S3BucketTagging (DeleteBucketTagging)" on target
"s3testbucket".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"): Y
```

- Para obter detalhes da API, consulte [DeleteBucketTagging](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **DeleteBucketWebsite** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `DeleteBucketWebsite`.

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
bool AwsDoc::S3::DeleteBucketWebsite(const Aws::String &bucketName,
                                     const Aws::Client::ClientConfiguration
                                     &clientConfig) {
    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::cerr << "Error: DeleteBucketWebsite: " <<
            err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    }
    else {
        std::cout << "Website configuration was removed." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Para obter detalhes da API, consulte [DeleteBucketWebsite](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

O seguinte comando exclui uma configuração de site do bucket my-bucket:

```
aws s3api delete-bucket-website --bucket my-bucket
```

- Para obter detalhes da API, consulte [DeleteBucketWebsite](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.DeleteBucketWebsiteRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;

/**
 * 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 DeleteWebsiteConfiguration {
    public static void main(String[] args) {
        final String usage = ""

            Usage:      <bucketName>

            Where:
                bucketName - The Amazon S3 bucket to delete the website
configuration from.
            """;

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

        String bucketName = args[0];
        System.out.format("Deleting website configuration for Amazon S3 bucket:
%s\n", bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        deleteBucketWebsiteConfig(s3, bucketName);
        System.out.println("Done!");
        s3.close();
    }

    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 AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Exclua a configuração de site do bucket.

```
import { DeleteBucketWebsiteCommand, S3Client } from "@aws-sdk/client-s3";  
  
const client = new S3Client({});  
  
// Disable static website hosting on the bucket.  
export const main = async () => {  
  const command = new DeleteBucketWebsiteCommand({  
    Bucket: "test-bucket",  
  });  
  
  try {  
    const response = await client.send(command);  
    console.log(response);  
  } catch (err) {  
    console.error(err);  
  }  
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).

- Para obter detalhes da API, consulte [DeleteBucketWebsite](#) na Referência da API AWS SDK for JavaScript.

PowerShell

Tools for PowerShell

Exemplo 1: este comando desabilita a propriedade de hospedagem de site estático do bucket do S3 em questão.

```
Remove-S3BucketWebsite -BucketName 's3testbucket'
```

Saída:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-S3BucketWebsite (DeleteBucketWebsite)" on target
"s3testbucket".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"): Y
```

- Para obter detalhes da API, consulte [DeleteBucketWebsite](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **DeleteObject** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `DeleteObject`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Trabalhar com objetos versionados](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Exclua um objeto de um bucket do S3 sem versionamento.

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

/// <summary>
/// This example shows how to delete an object from a non-versioned Amazon
/// Simple Storage Service (Amazon S3) bucket.
/// </summary>
public class DeleteObject
{
    /// <summary>
    /// The Main method initializes the necessary variables and then calls
    /// the DeleteObjectNonVersionedBucketAsync method to delete the object
    /// named by the keyName parameter.
    /// </summary>
    public static async Task Main()
    {
        const string bucketName = "doc-example-bucket";
        const string keyName = "testfile.txt";

        // If the Amazon S3 bucket is located in an AWS Region other than the
        // Region of the default account, define the AWS Region for the
        // Amazon S3 bucket in your call to the AmazonS3Client constructor.
        // For example RegionEndpoint.USWest2.
        IAmazonS3 client = new AmazonS3Client();
        await DeleteObjectNonVersionedBucketAsync(client, bucketName,
keyName);
    }

    /// <summary>
```

```
/// The DeleteObjectNonVersionedBucketAsync takes care of deleting the
/// desired object from the named bucket.
/// </summary>
/// <param name="client">An initialized Amazon S3 client used to delete
/// an object from an Amazon S3 bucket.</param>
/// <param name="bucketName">The name of the bucket from which the
/// object will be deleted.</param>
/// <param name="keyName">The name of the object to delete.</param>
public static async Task DeleteObjectNonVersionedBucketAsync(IAmazonS3
client, string bucketName, string keyName)
{
    try
    {
        var deleteObjectRequest = new DeleteObjectRequest
        {
            BucketName = bucketName,
            Key = keyName,
        };

        Console.WriteLine($"Deleting object: {keyName}");
        await client.DeleteObjectAsync(deleteObjectRequest);
        Console.WriteLine($"Object: {keyName} deleted from
{bucketName}.");
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error encountered on server.
Message: '{ex.Message}' when deleting an object.");
    }
}
}
```

Exclua um objeto de um bucket do S3 com versionamento.

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

/// <summary>
/// This example creates an object in an Amazon Simple Storage Service
```

```
/// (Amazon S3) bucket and then deletes the object version that was
/// created.
/// </summary>
public class DeleteObjectVersion
{
    public static async Task Main()
    {
        string bucketName = "doc-example-bucket";
        string keyName = "verstioned-object.txt";

        // If the AWS Region of the default user is different from the AWS
        // Region of the Amazon S3 bucket, pass the AWS Region of the
        // bucket region to the Amazon S3 client object's constructor.
        // Define it like this:
        //     RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        IAmazonS3 client = new AmazonS3Client();

        await CreateAndDeleteObjectVersionAsync(client, bucketName, keyName);
    }

    /// <summary>
    /// This method creates and then deletes a versioned object.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
    /// create and delete the object.</param>
    /// <param name="bucketName">The name of the Amazon S3 bucket where the
    /// object will be created and deleted.</param>
    /// <param name="keyName">The key name of the object to create.</param>
    public static async Task CreateAndDeleteObjectVersionAsync(IAmazonS3
client, string bucketName, string keyName)
    {
        try
        {
            // Add a sample object.
            string versionID = await PutAnObject(client, bucketName,
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 ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}

/// <summary>
/// This method is used to create the temporary Amazon S3 object.
/// </summary>
/// <param name="client">The initialized Amazon S3 object which will be
used
/// to create the temporary Amazon S3 object.</param>
/// <param name="bucketName">The name of the Amazon S3 bucket where the
object
/// will be created.</param>
/// <param name="objectKey">The name of the Amazon S3 object co create.</
param>
/// <returns>The Version ID of the created object.</returns>
public static async Task<string> PutAnObject(IAmazonS3 client, string
bucketName, 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;
}
}
```

- Para obter detalhes da API, consulte [DeleteObject](#) na Referência da API AWS SDK for .NET.

Bash

AWS CLI com script Bash

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function delete_item_in_bucket
#
# This function deletes the specified file from the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.
#     $2 - The key (file name) in the bucket to delete.

# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_item_in_bucket() {
    local bucket_name=$1
    local key=$2
    local response

    response=$(aws s3api delete-object \
        --bucket "$bucket_name" \
        --key "$key")

# shellcheck disable=SC2181
```

```
if [[ $? -ne 0 ]]; then
    errecho "ERROR: AWS reports s3api delete-object operation failed.\n
$response"
    return 1
fi
}
```

- Para obter detalhes da API, consulte [DeleteObject](#) na Referência de comandos da AWS CLI.

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
bool AwsDoc::S3::DeleteObject(const Aws::String &objectKey,
                              const Aws::String &fromBucket,
                              const Aws::Client::ClientConfiguration
&clientConfig) {
    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::cerr << "Error: DeleteObject: " <<
            err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    }
    else {
```



```
        std::cout << "Successfully deleted the object." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Para obter detalhes da API, consulte [DeleteObject](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

O seguinte comando exclui um objeto `test.txt` do bucket `my-bucket`:

```
aws s3api delete-object --bucket my-bucket --key test.txt
```

Se o versionamento de bucket estiver habilitado, a saída conterá o ID de versão do marcador excluído:

```
{
  "VersionId": "9_gKg5vG56F.TTEUdwkxGpJ3tND1WlGq",
  "DeleteMarker": true
}
```

Para obter mais informações sobre como excluir objetos, consulte Excluir objetos no Guia do desenvolvedor do Amazon S3.

- Para obter detalhes da API, consulte [DeleteObject](#) na Referência de comandos da AWS CLI.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Exclua um objeto.

```
import { DeleteObjectCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new DeleteObjectCommand({
    Bucket: "test-bucket",
    Key: "test-key.txt",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Para obter detalhes da API, consulte [DeleteObject](#) na Referência da API AWS SDK for JavaScript.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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
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(
obj_version_id
            "Object %s, version %s is a delete marker.", obj_key,
)
        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:
```

```
retried.

        # Mark request timeout as a temporary failure so it will be
        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 AWS SDK para Python (Boto3).

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Rust.

SAP ABAP

SDK para SAP ABAP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

TRY.

```
lo_s3->deleteobject(  
    iv_bucket = iv_bucket_name  
    iv_key = iv_object_key  
).  
MESSAGE 'Object deleted from S3 bucket.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- Para obter detalhes da API, consulte [DeleteObject](#) na Referência da API AWS SDK para SAP ABAP.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Swift.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **DeleteObjectTagging** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o DeleteObjectTagging.

CLI

AWS CLI

Para excluir os conjuntos de tags de um objeto

O exemplo `delete-object-tagging` a seguir exclui a tag com a chave especificada do objeto `doc1.rtf`.

```
aws s3api delete-object-tagging \  
  --bucket my-bucket \  
  --key doc1.rtf
```

Este comando não produz saída.

- Para obter detalhes da API, consulte [DeleteObjectTagging](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando remove todas as tags associadas ao objeto com a chave “`testfile.txt`” no bucket do S3 em questão.

```
Remove-S3ObjectTagSet -Key 'testfile.txt' -BucketName 's3testbucket' -Select
'^Key'
```

Saída:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-S3ObjectTagSet (DeleteObjectTagging)" on target
"testfile.txt".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"): Y
testfile.txt
```

- Para obter detalhes da API, consulte [DeleteObjectTagging](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **DeleteObjects** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `DeleteObjects`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Conceitos básicos de buckets e objetos](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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
    {
        ListObjectsV2Response response;

        do
        {
            response = await client.ListObjectsV2Async(request);
            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;
    }
}
```

```
    }
}
```

Exclua vários objetos de um bucket do S3 sem versionamento.

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to delete multiple objects from an Amazon Simple
/// Storage Service (Amazon S3) bucket.
/// </summary>
public class DeleteMultipleObjects
{
    /// <summary>
    /// The Main method initializes the Amazon S3 client and the name of
    /// the bucket and then passes those values to MultiObjectDeleteAsync.
    /// </summary>
    public static async Task Main()
    {
        const string bucketName = "doc-example-bucket";

        // If the Amazon S3 bucket from which you wish to delete objects is
        not
        // located in the same AWS Region as the default user, define the
        // AWS Region for the Amazon S3 bucket as a parameter to the client
        // constructor.
        IAmazonS3 s3Client = new AmazonS3Client();

        await MultiObjectDeleteAsync(s3Client, bucketName);
    }

    /// <summary>
    /// This method uses the passed Amazon S3 client to first create and then
    /// delete three files from the named bucket.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
    call
```

```
    /// Amazon S3 methods.</param>
    /// <param name="bucketName">The name of the Amazon S3 bucket where
objects
    /// will be created and then deleted.</param>
    public static async Task MultiObjectDeleteAsync(IAmazonS3 client, string
bucketName)
    {
        // Create three sample objects which we will then delete.
        var keysAndVersions = await PutObjectsAsync(client, 3, bucketName);

        // Now perform the multi-object delete, passing the key names and
        // version IDs. Since we are working with a non-versioned bucket,
        // the object keys collection includes null version IDs.
        DeleteObjectsRequest multiObjectDeleteRequest = new
DeleteObjectsRequest
        {
            BucketName = bucketName,
            Objects = keysAndVersions,
        };

        // You can add a specific object key to the delete request using the
        // AddKey method of the multiObjectDeleteRequest.
        try
        {
            DeleteObjectsResponse response = await
client.DeleteObjectsAsync(multiObjectDeleteRequest);
            Console.WriteLine("Successfully deleted all the {0} items",
response.DeletedObjects.Count);
        }
        catch (DeleteObjectsException e)
        {
            PrintDeletionErrorStatus(e);
        }
    }

    /// <summary>
    /// Prints the list of errors raised by the call to DeleteObjectsAsync.
    /// </summary>
    /// <param name="ex">A collection of exceptions returned by the call to
    /// DeleteObjectsAsync.</param>
    public static void PrintDeletionErrorStatus(DeleteObjectsException ex)
    {
        DeleteObjectsResponse errorResponse = ex.Response;
        Console.WriteLine("x {0}", errorResponse.DeletedObjects.Count);
    }
}
```

```
        Console.WriteLine($"Successfully deleted
{errorResponse.DeletedObjects.Count}.");
        Console.WriteLine($"No. of objects failed to delete =
{errorResponse.DeleteErrors.Count}");

        Console.WriteLine("Printing error data...");
        foreach (DeleteError deleteError in errorResponse.DeleteErrors)
        {
            Console.WriteLine($"Object Key:
{deleteError.Key}\t{deleteError.Code}\t{deleteError.Message}");
        }
    }

    /// <summary>
    /// This method creates simple text file objects that can be used in
    /// the delete method.
    /// </summary>
    /// <param name="client">The Amazon S3 client used to call
PutObjectAsync.</param>
    /// <param name="number">The number of objects to create.</param>
    /// <param name="bucketName">The name of the bucket where the objects
    /// will be created.</param>
    /// <returns>A list of keys (object keys) and versions that the calling
    /// method will use to delete the newly created files.</returns>
    public static async Task<List<KeyVersion>> PutObjectsAsync(IAmazonS3
client, int number, string bucketName)
    {
        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
client.PutObjectAsync(request);

            // For non-versioned bucket operations, we only need the
            // object key.
```



```
        KeyVersion keyVersion = new KeyVersion
        {
            Key = key,
        };
        keys.Add(keyVersion);
    }

    return keys;
}
}
```

Exclua vários objetos de um bucket do S3 com versionamento.

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to delete objects in a version-enabled Amazon
/// Simple StorageService (Amazon S3) bucket.
/// </summary>
public class DeleteMultipleObjects
{
    public static async Task Main()
    {
        string bucketName = "doc-example-bucket";

        // If the AWS Region for your Amazon S3 bucket is different from
        // the AWS Region of the default user, define the AWS Region for
        // the Amazon S3 bucket and pass it to the client constructor
        // like this:
        // RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        IAmazonS3 s3Client;

        s3Client = new AmazonS3Client();
        await DeleteMultipleObjectsFromVersionedBucketAsync(s3Client,
bucketName);
    }
}
```

```

    /// <summary>
    /// This method removes multiple versions and objects from a
    /// version-enabled Amazon S3 bucket.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// DeleteObjectVersionsAsync, DeleteObjectsAsync, and
    /// RemoveDeleteMarkersAsync.</param>
    /// <param name="bucketName">The name of the bucket from which to delete
    /// objects.</param>
    public static async Task
DeleteMultipleObjectsFromVersionedBucketAsync(IAmazonS3 client, string
bucketName)
    {
        // Delete objects (specifying object version in the request).
        await DeleteObjectVersionsAsync(client, bucketName);

        // Delete objects (without specifying object version in the request).
        var deletedObjects = await DeleteObjectsAsync(client, bucketName);

        // Additional exercise - remove the delete markers Amazon S3 returned
from
        // 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(client, bucketName, deletedObjects);
    }

    /// <summary>
    /// Creates and then deletes non-versioned Amazon S3 objects and then
deletes
    /// them again. The method returns a list of the Amazon S3 objects
    /// deleted.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// PubObjectsAsync and NonVersionedDeleteAsync.</param>
    /// <param name="bucketName">The name of the bucket where the objects
    /// will be created and then deleted.</param>
    /// <returns>A list of DeletedObjects.</returns>
    public static async Task<List<DeletedObject>>
DeleteObjectsAsync(IAmazonS3 client, string bucketName)
    {
        // Upload the sample objects.

```

```
        var keysAndVersions2 = await PutObjectsAsync(client, bucketName, 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(client, bucketName, keysAndVersions2);
        return deletedObjects;
    }

    /// <summary>
    /// This method creates several temporary objects and then deletes them.
    /// </summary>
    /// <param name="client">The S3 client.</param>
    /// <param name="bucketName">Name of the bucket.</param>
    /// <returns>Async task.</returns>
    public static async Task DeleteObjectVersionsAsync(IAmazonS3 client,
string bucketName)
    {
        // Upload the sample objects.
        var keysAndVersions1 = await PutObjectsAsync(client, bucketName, 3);

        // Delete the specific object versions.
        await VersionedDeleteAsync(client, bucketName, keysAndVersions1);
    }

    /// <summary>
    /// Displays the list of information about deleted files to the console.
    /// </summary>
    /// <param name="e">Error information from the delete process.</param>
    private static void DisplayDeletionErrors(DeleteObjectsException e)
    {
        var errorResponse = e.Response;
        Console.WriteLine($"No. of objects successfully deleted =
{errorResponse.DeletedObjects.Count}");
        Console.WriteLine($"No. of objects failed to delete =
{errorResponse.DeleteErrors.Count}");
        Console.WriteLine("Printing error data...");
        foreach (var deleteError in errorResponse.DeleteErrors)
        {
            Console.WriteLine($"Object Key:
{deleteError.Key}\t{deleteError.Code}\t{deleteError.Message}");
        }
    }
}
```

```

    /// <summary>
    /// Delete multiple objects from a version-enabled bucket.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// DeleteObjectVersionsAsync, DeleteObjectsAsync, and
    /// RemoveDeleteMarkersAsync.</param>
    /// <param name="bucketName">The name of the bucket from which to delete
    /// objects.</param>
    /// <param name="keys">A list of key names for the objects to delete.</
param>
    private static async Task VersionedDeleteAsync(IAmazonS3 client, string
bucketName, List<KeyVersion> keys)
    {
        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
client.DeleteObjectsAsync(multiObjectDeleteRequest);
            Console.WriteLine($"Successfully deleted all the
{response.DeletedObjects.Count} items");
        }
        catch (DeleteObjectsException ex)
        {
            DisplayDeletionErrors(ex);
        }
    }

    /// <summary>
    /// Deletes multiple objects from a non-versioned Amazon S3 bucket.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// DeleteObjectVersionsAsync, DeleteObjectsAsync, and
    /// RemoveDeleteMarkersAsync.</param>
    /// <param name="bucketName">The name of the bucket from which to delete

```

```
    /// objects.</param>
    /// <param name="keys">A list of key names for the objects to delete.</
param>
    /// <returns>A list of the deleted objects.</returns>
    private static async Task<List<DeletedObject>>
NonVersionedDeleteAsync(IAmazonS3 client, string bucketName, 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 DeleteObjectsAsync.
        // The DeleteObjectsAsync method adds a delete marker for each
        // object deleted. You can verify that the objects were removed
        // using the Amazon S3 console.
        DeleteObjectsResponse response;
        try
        {
            Console.WriteLine("Executing NonVersionedDelete...");
            response = await
client.DeleteObjectsAsync(multiObjectDeleteRequest);
            Console.WriteLine("Successfully deleted all the {0} items",
response.DeletedObjects.Count);
        }
        catch (DeleteObjectsException ex)
        {
            DisplayDeletionErrors(ex);
            throw; // Some deletions failed. Investigate before continuing.
        }

        // This response contains the DeletedObjects list which we use to
delete the delete markers.
        return response.DeletedObjects;
    }

    /// <summary>
    /// Deletes the markers left after deleting the temporary objects.
```

```
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// DeleteObjectVersionsAsync, DeleteObjectsAsync, and
    /// RemoveDeleteMarkersAsync.</param>
    /// <param name="bucketName">The name of the bucket from which to delete
    /// objects.</param>
    /// <param name="deletedObjects">A list of the objects that were
deleted.</param>
    private static async Task RemoveDeleteMarkersAsync(IAmazonS3 client,
string bucketName, 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
client.DeleteObjectsAsync(multiObjectDeleteRequest);
            Console.WriteLine($"Successfully deleted the
{deleteObjectResponse.DeletedObjects.Count} delete markers");
        }
        catch (DeleteObjectsException ex)
        {
            DisplayDeletionErrors(ex);
        }
    }
}
```

```
    }
  }

  /// <summary>
  /// Create temporary Amazon S3 objects to show how object deletion works
in an
  /// Amazon S3 bucket with versioning enabled.
  /// </summary>
  /// <param name="client">The initialized Amazon S3 client object used to
call
  /// PutObjectAsync to create temporary objects for the example.</param>
  /// <param name="bucketName">A string representing the name of the S3
  /// bucket where we will create the temporary objects.</param>
  /// <param name="number">The number of temporary objects to create.</
param>
  /// <returns>A list of the KeyVersion objects.</returns>
  private static async Task<List<KeyVersion>> PutObjectsAsync(IAmazonS3
client, string bucketName, 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 client.PutObjectAsync(request);
      KeyVersion keyVersion = new KeyVersion
      {
        Key = key,
        VersionId = response.VersionId,
      };

      keys.Add(keyVersion);
    }

    return keys;
  }
}
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API AWS SDK for .NET.

Bash

AWS CLI com script Bash

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function delete_items_in_bucket
#
# This function deletes the specified list of keys from the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.
#     $2 - A list of keys in the bucket to delete.

# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_items_in_bucket() {
    local bucket_name=$1
    local keys=$2
```



```

local response

# Create the JSON for the items to delete.
local delete_items
delete_items="{\"Objects\":["
for key in $keys; do
    delete_items="$delete_items{\"Key\": \"$key\"},"
done
delete_items=${delete_items%?} # Remove the final comma.
delete_items="$delete_items]}"

response=$(aws s3api delete-objects \
    --bucket "$bucket_name" \
    --delete "$delete_items")

# shellcheck disable=SC2181
if [[ $? -ne 0 ]]; then
    errecho "ERROR: AWS reports s3api delete-object operation failed.\n
$response"
    return 1
fi
}

```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência de comandos da AWS CLI.

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```

bool AwsDoc::S3::DeleteObjects(const std::vector<Aws::String> &objectKeys,
                               const Aws::String &fromBucket,
                               const Aws::Client::ClientConfiguration
                               &clientConfig) {

```

```
Aws::S3::S3Client client(clientConfig);
Aws::S3::Model::DeleteObjectsRequest request;

Aws::S3::Model::Delete deleteObject;
for (const Aws::String& objectKey : objectKeys)
{
deleteObject.AddObjects(Aws::S3::Model::ObjectIdentifier().WithKey(objectKey));
}

request.SetDelete(deleteObject);
request.SetBucket(fromBucket);

Aws::S3::Model::DeleteObjectsOutcome outcome =
    client.DeleteObjects(request);

if (!outcome.IsSuccess()) {
    auto err = outcome.GetError();
    std::cerr << "Error deleting objects. " <<
        err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
}
else {
    std::cout << "Successfully deleted the objects.";
    for (size_t i = 0; i < objectKeys.size(); ++i)
    {
        std::cout << objectKeys[i];
        if (i < objectKeys.size() - 1)
        {
            std::cout << ", ";
        }
    }

    std::cout << " from bucket " << fromBucket << "." << std::endl;
}

return outcome.IsSuccess();
}
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

O seguinte comando exclui um objeto do bucket my-bucket:

```
aws s3api delete-objects --bucket my-bucket --delete file://delete.json
```

delete.json é um documento JSON no diretório atual que especifica o objeto a ser excluído:

```
{
  "Objects": [
    {
      "Key": "test1.txt"
    }
  ],
  "Quiet": false
}
```

Saída:

```
{
  "Deleted": [
    {
      "DeleteMarkerVersionId": "mYAT5Mc6F7aeUL8SS7FAAqUP01koHwzU",
      "Key": "test1.txt",
      "DeleteMarker": true
    }
  ]
}
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência de comandos da AWS CLI.

Go

SDK para Go V2

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
// actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// DeleteObjects deletes a list of objects from a bucket.
func (basics BucketBasics) DeleteObjects(bucketName string, objectKeys []string)
error {
    var objectIds []types.ObjectIdentifier
    for _, key := range objectKeys {
        objectIds = append(objectIds, types.ObjectIdentifier{Key: aws.String(key)})
    }
    output, err := basics.S3Client.DeleteObjects(context.TODO(),
    &s3.DeleteObjectsInput{
        Bucket: aws.String(bucketName),
        Delete: &types.Delete{Objects: objectIds},
    })
    if err != nil {
        log.Printf("Couldn't delete objects from bucket %v. Here's why: %v\n",
        bucketName, err)
    } else {
        log.Printf("Deleted %v objects.\n", len(output.Deleted))
    }
    return err
}
```

```
}
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API AWS SDK for Go.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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.PutObjectRequest;
import software.amazon.awssdk.services.s3.model.ObjectIdentifier;
import software.amazon.awssdk.services.s3.model.Delete;
import software.amazon.awssdk.services.s3.model.DeleteObjectsRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.util.ArrayList;

/**
 * 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 DeleteMultiObjects {
    public static void main(String[] args) {
        final String usage = ""

                Usage:    <bucketName>
```

```
        Where:
            bucketName - the Amazon S3 bucket name.
        """;

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

    String bucketName = args[0];
    Region region = Region.US_EAST_1;
    S3Client s3 = S3Client.builder()
        .region(region)
        .build();

    deleteBucketObjects(s3, bucketName);
    s3.close();
}

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 AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Exclua vários objetos.

```
import { DeleteObjectsCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
```

```
const command = new DeleteObjectsCommand({
  Bucket: "test-bucket",
  Delete: {
    Objects: [{ Key: "object1.txt" }, { Key: "object2.txt" }],
  },
});

try {
  const { Deleted } = await client.send(command);
  console.log(
    `Successfully deleted ${Deleted.length} objects from S3 bucket. Deleted
objects:`,
  );
  console.log(Deleted.map((d) => ` • ${d.Key}`).join("\n"));
} catch (err) {
  console.error(err);
}
};
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API AWS SDK for JavaScript.

Kotlin

SDK for Kotlin

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Kotlin.

PHP

SDK para PHP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Exclua um conjunto de objetos de uma lista de chaves.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

try {
    $objects = [];
    foreach ($contents['Contents'] as $content) {
        $objects[] = [
            'Key' => $content['Key'],
        ];
    }
    $this->s3client->deleteObjects([
        'Bucket' => $this->bucketName,
        'Delete' => [
            'Objects' => $objects,
        ],
    ],
```

```
]);
$check = $this->s3client->listObjectsV2([
    'Bucket' => $this->bucketName,
]);
if (count($check) <= 0) {
    throw new Exception("Bucket wasn't empty.");
}
echo "Deleted all objects and folders from $this->bucketName.\n";
} catch (Exception $exception) {
    echo "Failed to delete $fileName from $this->bucketName with error:
" . $exception->getMessage();
    exit("Please fix error with object deletion before continuing.");
}
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API AWS SDK for PHP.

PowerShell

Tools for PowerShell

Exemplo 1: este comando remove o objeto “sample.txt” do bucket “test-files”. Será solicitada uma confirmação antes que o comando seja executado. Para ignorar a solicitação, use a opção -Force.

```
Remove-S3Object -BucketName test-files -Key sample.txt
```

Exemplo 2: este comando remove a versão especificada do objeto “sample.txt” do bucket “test-files”, supondo que o bucket tenha sido configurado para habilitar versões de objetos.

```
Remove-S3Object -BucketName test-files -Key sample.txt -VersionId
HLbxnx6V9omT6AQYVpks8mmFKQcejpqt
```

Exemplo 3: este comando remove objetos “sample1.txt”, “sample2.txt” e “sample3.txt” do bucket “test-files” como uma operação em lote única. A resposta do serviço listará todas as chaves processadas, independentemente do status de êxito ou erro da exclusão. Para obter somente erros de chaves que não puderam ser processadas pelo serviço, adicione o parâmetro -ReportErrorsOnly (esse parâmetro também pode ser especificado com o alias -Quiet).

```
Remove-S3Object -BucketName test-files -KeyCollection @( "sample1.txt",  
"sample2.txt", "sample3.txt" )
```

Exemplo 4: este exemplo usa uma expressão em linha com o parâmetro `-KeyCollection` para obter as chaves dos objetos a serem excluídos. `Get-S3Object` retorna um conjunto de instâncias `Amazon.S3.Model.S3Object`, cada uma com um membro de chave do tipo `string` identificando o objeto.

```
Remove-S3Object -bucketname "test-files" -KeyCollection (Get-S3Object "test-  
files" -KeyPrefix "prefix/subprefix" | select -ExpandProperty Key)
```

Exemplo 5: este exemplo obtém todos os objetos que têm um prefixo de chave “prefix/subprefix” no bucket e os exclui. Observe que os objetos de entrada são processados um de cada vez. Para conjuntos grandes, considere a possibilidade de passar o conjunto para o parâmetro `-InputObject` (alias `-S3ObjectCollection`) do cmdlet para permitir que a exclusão ocorra em lote com uma única chamada ao serviço.

```
Get-S3Object -BucketName "test-files" -KeyPrefix "prefix/subprefix" | Remove-  
S3Object -Force
```

Exemplo 6: este exemplo envia um conjunto de instâncias `Amazon.S3.Model.S3ObjectVersion` que representam marcadores de exclusão do cmdlet para exclusão. Observe que os objetos de entrada são processados um de cada vez. Para conjuntos grandes, considere a possibilidade de passar o conjunto para o parâmetro `-InputObject` (alias `-S3ObjectCollection`) do cmdlet para permitir que a exclusão ocorra em lote com uma única chamada ao serviço.

```
(Get-S3Version -BucketName "test-files").Versions | Where {$_.IsDeleteMarker -eq  
"True"} | Remove-S3Object -Force
```

Exemplo 7: este script mostra como realizar uma exclusão em lote de um conjunto de objetos (nesse caso, marcadores de exclusão) construindo uma matriz de objetos a serem usados com o parâmetro `-KeyAndVersionCollection`.

```
$keyVersions = @()  
$markers = (Get-S3Version -BucketName $BucketName).Versions | Where  
{$_ .IsDeleteMarker -eq "True"}  
foreach ($marker in $markers) { $keyVersions += @{ Key = $marker.Key; VersionId =  
$marker.VersionId } }
```

```
Remove-S3Object -BucketName $BucketName -KeyAndVersionCollection $keyVersions -
Force
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na AWS Tools for PowerShell Cmdlet Reference.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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'.",
                [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 AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
# 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 AWS SDK for Ruby.

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
pub async fn delete_objects(client: &Client, bucket_name: &str) ->
Result<Vec<String>, Error> {
    let objects = client.list_objects_v2().bucket(bucket_name).send().await?;

    let mut delete_objects: Vec<ObjectIdentifier> = vec![];
    for obj in objects.contents() {
        let obj_id = ObjectIdentifier::builder()
            .set_key(Some(obj.key().unwrap().to_string()))
            .build()
            .map_err(Error::from)?;
        delete_objects.push(obj_id);
    }

    let return_keys = delete_objects.iter().map(|o| o.key.clone()).collect();

    if !delete_objects.is_empty() {
        client
            .delete_objects()
            .bucket(bucket_name)
            .delete(
                Delete::builder()
                    .set_objects(Some(delete_objects))
                    .build()
                    .map_err(Error::from)?,
            )
            .send()
            .await?;
    }

    let objects: ListObjectsV2Output =
client.list_objects_v2().bucket(bucket_name).send().await?;

    eprintln!("{objects:?}");

    match objects.key_count {
        Some(0) => Ok(return_keys),
        _ => Err(Error::unhandled(
            "There were still objects left in the bucket.",
        )),
    }
}
```


- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API 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.

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 {
        let output = try await client.deleteObjects(input: input)

        // As of the last update to this example, any errors are returned
        // in the `output` object's `errors` property. If there are any
        // errors in this array, throw an exception. Once the error
        // handling is finalized in later updates to the AWS SDK for
        // Swift, this example will be updated to handle errors better.

        guard let errors = output.errors else {
            return // No errors.
        }
    }
}
```

```
        if errors.count != 0 {
            throw ServiceHandlerError.deleteObjectsError
        }
    } catch {
        throw error
    }
}
```

- Para obter detalhes da API, consulte [DeleteObjects](#) na Referência da API AWS SDK para Swift.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **DeletePublicAccessBlock** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o DeletePublicAccessBlock.

CLI

AWS CLI

Para excluir a configuração do Bloqueio de Acesso Público para um bucket

O exemplo delete-public-access-block a seguir remove a configuração do Bloqueio de Acesso Público no bucket especificado.

```
aws s3api delete-public-access-block \
    --bucket my-bucket
```

Este comando não produz saída.

- Para obter detalhes da API, consulte [DeletePublicAccessBlock](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando desativa a configuração do Bloqueio de Acesso Público do bucket em questão.

```
Remove-S3PublicAccessBlock -BucketName 's3testbucket' -Force -Select  
'^BucketName'
```

Saída:

```
s3testbucket
```

- Para obter detalhes da API, consulte [DeletePublicAccessBlock](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketAccelerateConfiguration** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetBucketAccelerateConfiguration`.

CLI

AWS CLI

Para recuperar a configuração de aceleração de um bucket

O exemplo `get-bucket-accelerate-configuration` a seguir recupera a configuração de aceleração do bucket especificado.

```
aws s3api get-bucket-accelerate-configuration \  
  --bucket my-bucket
```

Saída:

```
{
  "Status": "Enabled"
}
```

- Para obter detalhes da API, consulte [GetBucketAccelerateConfiguration](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: se as configurações de aceleração de transferência estiverem habilitadas para o bucket especificado, este comando retornará o valor `Habilitado`.

```
Get-S3BucketAccelerateConfiguration -BucketName 's3testbucket'
```

Saída:

```
Value
-----
Enabled
```

- Para obter detalhes da API, consulte [GetBucketAccelerateConfiguration](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketAcl** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetBucketAcl`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Gerenciar listas de controle de acesso \(ACLs\)](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).


```
    /// <summary>
    /// Get the access control list (ACL) for the new bucket.
    /// </summary>
    /// <param name="client">The initialized client object used to get the
    /// access control list (ACL) of the bucket.</param>
    /// <param name="newBucketName">The name of the newly created bucket.</
param>
    /// <returns>An S3AccessControlList.</returns>
    public static async Task<S3AccessControlList>
    GetACLForBucketAsync(IAmazonS3 client, string newBucketName)
    {
        // Retrieve bucket ACL to show that the ACL was properly applied to
        // the new bucket.
        GetACLResponse getACLResponse = await client.GetACLAsync(new
    GetACLRequest
    {
        BucketName = newBucketName,
    });

        return getACLResponse.AccessControlList;
    }
```

- Para obter detalhes da API, consulte [GetBucketAcl](#) na Referência da API AWS SDK for .NET.

C++

SDK para C++

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
bool AwsDoc::S3::GetBucketAcl(const Aws::String &bucketName,
                             const Aws::Client::ClientConfiguration
&clientConfig) {
    Aws::S3::S3Client s3_client(clientConfig);

    Aws::S3::Model::GetBucketAclRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::GetBucketAclOutcome outcome =
        s3_client.GetBucketAcl(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: GetBucketAcl: "
            << err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    }
    else {
        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(grant.GetPermission()) <<
              std::endl << std::endl;
}
}

return outcome.IsSuccess();
}

//! Routine which converts a built-in type enumeration to a human-readable
string.
/*!
 \sa GetGranteeTypeString()
 \param type Type enumeration.
 */

Aws::String GetGranteeTypeString(const Aws::S3::Model::Type &type) {
    switch (type) {
        case Aws::S3::Model::Type::AmazonCustomerByEmail:
            return "Email address of an AWS account";
        case Aws::S3::Model::Type::CanonicalUser:
            return "Canonical user ID of an AWS account";
    }
}

```

```
        case Aws::S3::Model::Type::Group:
            return "Predefined Amazon S3 group";
        case Aws::S3::Model::Type::NOT_SET:
            return "Not set";
        default:
            return "Type unknown";
    }
}

//! Routine which converts a built-in type enumeration to a human-readable
string.
/*!
 \sa GetPermissionString()
 \param permission Permission enumeration.
 */

Aws::String GetPermissionString(const Aws::S3::Model::Permission &permission) {
    switch (permission) {
        case Aws::S3::Model::Permission::FULL_CONTROL:
            return "Can list objects in this bucket, create/overwrite/delete "
                "objects in this bucket, and read/write this "
                "bucket's permissions";
        case Aws::S3::Model::Permission::NOT_SET:
            return "Permission not set";
        case Aws::S3::Model::Permission::READ:
            return "Can list objects in this bucket";
        case Aws::S3::Model::Permission::READ_ACP:
            return "Can read this bucket's permissions";
        case Aws::S3::Model::Permission::WRITE:
            return "Can create, overwrite, and delete objects in this bucket";
        case Aws::S3::Model::Permission::WRITE_ACP:
            return "Can write this bucket's permissions";
        default:
            return "Permission unknown";
    }

    return "Permission unknown";
}
```

- Para obter detalhes da API, consulte [GetBucketAcl](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

O seguinte comando recupera a lista de controle de acesso do bucket `my-bucket`:

```
aws s3api get-bucket-acl --bucket my-bucket
```

Saída:

```
{
  "Owner": {
    "DisplayName": "my-username",
    "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
  },
  "Grants": [
    {
      "Grantee": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
      },
      "Permission": "FULL_CONTROL"
    }
  ]
}
```

- Para obter detalhes da API, consulte [GetBucketAcl](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
import software.amazon.awssdk.services.s3.model.S3Exception;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectAclRequest;
import software.amazon.awssdk.services.s3.model.GetObjectAclResponse;
import software.amazon.awssdk.services.s3.model.Grant;
import java.util.List;

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

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

            Usage:
                <bucketName> <objectKey>

            Where:
                bucketName - The Amazon S3 bucket to get the access control
list (ACL) for.
                objectKey - The object to get the ACL for.\s
            """;

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

        String bucketName = args[0];
        String objectKey = args[1];
        System.out.println("Retrieving ACL for object: " + objectKey);
        System.out.println("in bucket: " + bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();
```

```
        getBucketACL(s3, objectKey, bucketName);
        s3.close();
        System.out.println("Done!");
    }

    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 AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Obtenha as permissões de ACL.

```
import { GetBucketAclCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new GetBucketAclCommand({
    Bucket: "test-bucket",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).
- Para obter detalhes da API, consulte [GetBucketAcl](#) na Referência da API AWS SDK for JavaScript.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketAnalyticsConfiguration** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetBucketAnalyticsConfiguration`.

CLI

AWS CLI

Para recuperar a configuração de análise de um bucket com um ID específico

O exemplo `get-bucket-analytics-configuration` a seguir recupera a configuração de análise referente ao bucket e ID especificados.

```
aws s3api get-bucket-analytics-configuration \  
  --bucket my-bucket \  
  --id 1
```

Saída:

```
{  
  "AnalyticsConfiguration": {  
    "StorageClassAnalysis": {},  
    "Id": "1"  
  }  
}
```

- Para obter detalhes da API, consulte [GetBucketAnalyticsConfiguration](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando retorna os detalhes do filtro de análise com o nome “testfilter” no bucket do S3 em questão.

```
Get-S3BucketAnalyticsConfiguration -BucketName 's3testbucket' -AnalyticsId  
'testfilter'
```

- Para obter os detalhes da API, consulte [GetBucketAnalyticsConfiguration](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketCors** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetBucketCors`.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <summary>  
/// Retrieve the CORS configuration applied to the Amazon S3 bucket.  
/// </summary>  
/// <param name="client">The initialized Amazon S3 client object used  
/// to retrieve the CORS configuration.</param>  
/// <returns>The created CORS configuration object.</returns>
```

```
private static async Task<CORSConfiguration>
RetrieveCORSConfigurationAsync(AmazonS3Client client)
{
    GetCORSConfigurationRequest request = new
GetCORSConfigurationRequest()
    {
        BucketName = BucketName,
    };
    var response = await client.GetCORSConfigurationAsync(request);
    var configuration = response.Configuration;
    PrintCORSRules(configuration);
    return configuration;
}
```

- Para obter detalhes da API, consulte [GetBucketCors](#) na Referência da API AWS SDK for .NET.

CLI

AWS CLI

O seguinte comando recupera a configuração de compartilhamento de recursos de origem cruzada para o bucket `my-bucket`:

```
aws s3api get-bucket-cors --bucket my-bucket
```

Saída:

```
{
  "CORSRules": [
    {
      "AllowedHeaders": [
        "*"
      ],
      "ExposeHeaders": [
        "x-amz-server-side-encryption"
      ],
      "AllowedMethods": [
        "PUT",
        "POST",
```



```
        "DELETE"
      ],
      "MaxAgeSeconds": 3000,
      "AllowedOrigins": [
        "http://www.example.com"
      ]
    },
    {
      "AllowedHeaders": [
        "Authorization"
      ],
      "MaxAgeSeconds": 3000,
      "AllowedMethods": [
        "GET"
      ],
      "AllowedOrigins": [
        "*"
      ]
    }
  ]
}
```

- Para obter detalhes da API, consulte [GetBucketCors](#) na Referência de comandos da AWS CLI.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Obtenha a política de CORS para o bucket.

```
import { GetBucketCorsCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});
```

```
export const main = async () => {
  const command = new GetBucketCorsCommand({
    Bucket: "test-bucket",
  });

  try {
    const { CORSRules } = await client.send(command);
    CORSRules.forEach((cr, i) => {
      console.log(
        `\\nCORSRule ${i + 1}`,
        `\\n${"-".repeat(10)}`,
        `\\nAllowedHeaders: ${cr.AllowedHeaders.join(" ")}`,
        `\\nAllowedMethods: ${cr.AllowedMethods.join(" ")}`,
        `\\nAllowedOrigins: ${cr.AllowedOrigins.join(" ")}`,
        `\\nExposeHeaders: ${cr.ExposeHeaders.join(" ")}`,
        `\\nMaxAgeSeconds: ${cr.MaxAgeSeconds}`,
      );
    });
  } catch (err) {
    console.error(err);
  }
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).
- Para obter detalhes da API, consulte [GetBucketCors](#) na Referência da API AWS SDK for JavaScript.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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

end
```

- Para obter detalhes da API, consulte [GetBucketCors](#) na Referência da API AWS SDK for Ruby.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketEncryption** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetBucketEncryption`.

CLI

AWS CLI

Para recuperar a configuração de criptografia do lado do servidor para um bucket

O exemplo `get-bucket-encryption` a seguir recupera a configuração de criptografia do lado do servidor do bucket `my-bucket`.

```
aws s3api get-bucket-encryption \  
  --bucket my-bucket
```

Saída:

```
{  
  "ServerSideEncryptionConfiguration": {  
    "Rules": [  
      {  
        "ApplyServerSideEncryptionByDefault": {  
          "SSEAlgorithm": "AES256"  
        }  
      }  
    ]  
  }  
}
```

- Para obter detalhes da API, consulte [GetBucketEncryption](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando retorna todas as regras de criptografia do lado do servidor associadas ao bucket em questão.

```
Get-S3BucketEncryption -BucketName 's3casetestbucket'
```

- Para obter detalhes da API, consulte [GetBucketEncryption](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketInventoryConfiguration** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `GetBucketInventoryConfiguration`.

CLI

AWS CLI

Para recuperar a configuração de inventário de um bucket

O exemplo `get-bucket-inventory-configuration` a seguir recupera a configuração de inventário do bucket especificado com ID 1.

```
aws s3api get-bucket-inventory-configuration \  
  --bucket my-bucket \  
  --id 1
```

Saída:

```
{  
  "InventoryConfiguration": {  
    "IsEnabled": true,  
  }  
}
```

```
    "Destination": {
      "S3BucketDestination": {
        "Format": "ORC",
        "Bucket": "arn:aws:s3:::my-bucket",
        "AccountId": "123456789012"
      }
    },
    "IncludedObjectVersions": "Current",
    "Id": "1",
    "Schedule": {
      "Frequency": "Weekly"
    }
  }
}
```

- Para obter detalhes da API, consulte [GetBucketInventoryConfiguration](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando retorna os detalhes do inventário denominado “testinventory” para o bucket do S3 em questão.

```
Get-S3BucketInventoryConfiguration -BucketName 's3testbucket' -InventoryId
'testinventory'
```

- Para obter detalhes da API, consulte [GetBucketInventoryConfiguration](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketLifecycleConfiguration** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetBucketLifecycleConfiguration`.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <summary>
/// Returns a configuration object for the supplied bucket name.
/// </summary>
/// <param name="client">The S3 client object used to call
/// the GetLifecycleConfigurationAsync method.</param>
/// <param name="bucketName">The name of the S3 bucket for which a
/// configuration will be created.</param>
/// <returns>Returns a new LifecycleConfiguration object.</returns>
public static async Task<LifecycleConfiguration>
RetrieveLifecycleConfigAsync(IAmazonS3 client, string bucketName)
{
    var request = new GetLifecycleConfigurationRequest()
    {
        BucketName = bucketName,
    };
    var response = await client.GetLifecycleConfigurationAsync(request);
    var configuration = response.Configuration;
    return configuration;
}
```

- Para obter detalhes da API, consulte [GetBucketLifecycleConfiguration](#) na Referência da API AWS SDK for .NET.

CLI

AWS CLI

O seguinte comando recupera a configuração do ciclo de vida do bucket my-bucket:


```
aws s3api get-bucket-lifecycle-configuration --bucket my-bucket
```

Saída:

```
{
  "Rules": [
    {
      "ID": "Move rotated logs to Glacier",
      "Prefix": "rotated/",
      "Status": "Enabled",
      "Transitions": [
        {
          "Date": "2015-11-10T00:00:00.000Z",
          "StorageClass": "GLACIER"
        }
      ]
    },
    {
      "Status": "Enabled",
      "Prefix": "",
      "NoncurrentVersionTransitions": [
        {
          "NoncurrentDays": 0,
          "StorageClass": "GLACIER"
        }
      ],
      "ID": "Move old versions to Glacier"
    }
  ]
}
```

- Para obter detalhes da API, consulte [GetBucketLifecycleConfiguration](#) na Referência de comandos da AWS CLI.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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:
            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 AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketLocation** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetBucketLocation`.

CLI

AWS CLI

O seguinte comando recupera a restrição de localização do bucket `my-bucket`, se houver uma restrição:

```
aws s3api get-bucket-location --bucket my-bucket
```

Saída:

```
{
  "LocationConstraint": "us-west-2"
}
```

- Para obter detalhes da API, consulte [GetBucketLocation](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: se houver uma restrição, este comando retornará a restrição de localização do bucket `"s3testbucket"`.

```
Get-S3BucketLocation -BucketName 's3testbucket'
```

Saída:

```
Value
-----
ap-south-1
```

- Para obter detalhes da API, consulte [GetBucketLocation](#) na AWS Tools for PowerShell Cmdlet Reference.

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
async fn show_buckets(strict: bool, client: &Client, region: &str) -> Result<(),
Error> {
    let resp = client.list_buckets().send().await?;
    let buckets = resp.buckets();
    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 AWS SDK para Rust.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketLogging** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `GetBucketLogging`.

CLI

AWS CLI

Para recuperar o status de registro em log de um bucket

O exemplo `get-bucket-logging` a seguir recupera o status de registro em log do bucket especificado.

```
aws s3api get-bucket-logging \
```

```
--bucket my-bucket
```

Saída:

```
{
  "LoggingEnabled": {
    "TargetPrefix": "",
    "TargetBucket": "my-bucket-logs"
  }
}
```

- Para obter detalhes da API, consulte [GetBucketLogging](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando retorna o status de registro em log do bucket especificado.

```
Get-S3BucketLogging -BucketName 's3testbucket'
```

Saída:

TargetBucketName	Grants	TargetPrefix
testbucket1	{}	testprefix

- Para obter detalhes da API, consulte [GetBucketLogging](#), na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketMetricsConfiguration** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetBucketMetricsConfiguration`.

CLI

AWS CLI

Para recuperar a configuração de métricas de um bucket com um ID específico

O exemplo `get-bucket-metrics-configuration` a seguir exibe a configuração de métricas referente ao bucket e ID especificados.

```
aws s3api get-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123
```

Saída:

```
{  
  "MetricsConfiguration": {  
    "Filter": {  
      "Prefix": "logs"  
    },  
    "Id": "123"  
  }  
}
```

- Para obter detalhes da API, consulte [GetBucketMetricsConfiguration](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando retorna detalhes sobre o filtro de métricas denominado “testfilter” para o bucket do S3 em questão.

```
Get-S3BucketMetricsConfiguration -BucketName 's3testbucket' -MetricsId  
'testfilter'
```

- Para obter detalhes da API, consulte [GetBucketMetricsConfiguration](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketNotification** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetBucketNotification`.

CLI

AWS CLI

O seguinte comando recupera a configuração de notificação do bucket denominado `my-bucket`:

```
aws s3api get-bucket-notification --bucket my-bucket
```

Saída:

```
{
  "TopicConfiguration": {
    "Topic": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
    "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MCM0ZjgyLWZDBiZWw1",
    "Event": "s3:ObjectCreated:*",
    "Events": [
      "s3:ObjectCreated:*"
    ]
  }
}
```

- Para obter detalhes da API, consulte [GetBucketNotification](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este exemplo recupera a configuração de notificação do bucket em questão.


```
Get-S3BucketNotification -BucketName kt-tools | select -ExpandProperty
TopicConfigurations
```

Saída:

```
Id      Topic
--      -
mimo    arn:aws:sns:eu-west-1:123456789012:topic-1
```

- Para obter detalhes da API, consulte [GetBucketNotification](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketPolicy** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetBucketPolicy`.

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
bool AwsDoc::S3::GetBucketPolicy(const Aws::String &bucketName,
                                const Aws::Client::ClientConfiguration
                                &clientConfig) {
    Aws::S3::S3Client s3_client(clientConfig);

    Aws::S3::Model::GetBucketPolicyRequest request;
    request.SetBucket(bucketName);
```

```

    Aws::S3::Model::GetBucketPolicyOutcome outcome =
        s3_client.GetBucketPolicy(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: GetBucketPolicy: "
            << err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    }
    else {
        Aws::StringStream policy_stream;
        Aws::String line;

        outcome.GetResult().GetPolicy() >> line;
        policy_stream << line;

        std::cout << "Retrieve the policy for bucket '" << bucketName << "':\n\n"
<<
            policy_stream.str() << std::endl;
    }

    return outcome.IsSuccess();
}

```

- Para obter detalhes da API, consulte [GetBucketPolicy](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

O seguinte comando recupera política de bucket do bucket my-bucket:

```
aws s3api get-bucket-policy --bucket my-bucket
```

Saída:

```
{
  "Policy": "{\"Version\":\"2008-10-17\",\"Statement\": [{\"Sid\":\"\", \"Effect\": \"Allow\", \"Principal\": \"*\", \"Action\": \"s3:GetObject\", \"Resource\": \"arn:aws:s3:::my-bucket/*\"}, {\"Sid\":\"\", \"Effect\": \"Deny\", \"Principal\": \"\"} ]}"
}
```

```
\"*\", \"Action\": \"s3:GetObject\", \"Resource\": \"arn:aws:s3::my-bucket/secret/*\n\"]]]\"\n}
```

Obter e colocar uma política de bucket em vigor O exemplo a seguir mostra como fazer download de uma política do Amazon S3, fazer alterações no arquivo e usar `put-bucket-policy` para aplicar a política de bucket modificada. Para fazer o download de uma política de bucket em um arquivo, você pode executar:

```
aws s3api get-bucket-policy --bucket mybucket --query Policy --output text >\n  policy.json
```

Em seguida, modifique o arquivo `policy.json` conforme necessário. Por fim, aplique essa política modificada ao bucket do S3 executando:

o arquivo `policy.json` conforme necessário. Por fim, aplique essa política modificada ao bucket do S3 executando:

conforme necessário. Por fim, aplique essa política modificada ao bucket do S3 executando:

```
aws s3api put-bucket-policy --bucket mybucket --policy file://policy.json
```

- Para obter detalhes da API, consulte [GetBucketPolicy](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
import software.amazon.awssdk.services.s3.model.S3Exception;\nimport software.amazon.awssdk.regions.Region;\nimport software.amazon.awssdk.services.s3.S3Client;\nimport software.amazon.awssdk.services.s3.model.GetBucketPolicyRequest;\nimport software.amazon.awssdk.services.s3.model.GetBucketPolicyResponse;
```

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

            Usage:
                <bucketName>

            Where:
                bucketName - The Amazon S3 bucket to get the policy from.
            """;

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

        String bucketName = args[0];
        System.out.format("Getting policy for bucket: \"%s\"\n\n", bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        String polText = getPolicy(s3, bucketName);
        System.out.println("Policy Text: " + polText);
        s3.close();
    }

    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 AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Obtenha a política de bucket.

```
import { GetBucketPolicyCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
    const command = new GetBucketPolicyCommand({
        Bucket: "test-bucket",
    });
};
```

```
try {
  const { Policy } = await client.send(command);
  console.log(JSON.parse(Policy));
} catch (err) {
  console.error(err);
}
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).
- Para obter detalhes da API, consulte [GetBucketPolicy](#) na Referência da API AWS SDK for JavaScript.

Kotlin

SDK for Kotlin

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Kotlin.

PowerShell

Tools for PowerShell

Exemplo 1: este comando gera a política de bucket associada ao bucket do S3 em questão.

```
Get-S3BucketPolicy -BucketName 's3testbucket'
```

- Para obter detalhes da API, consulte [GetBucketPolicy](#) na AWS Tools for PowerShell Cmdlet Reference.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
# 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 AWS SDK for Ruby.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketPolicyStatus** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetBucketPolicyStatus`.

CLI

AWS CLI

Para recuperar o status da política de um bucket que indica se o bucket é público

O exemplo `get-bucket-policy-status` a seguir recupera o status da política do bucket `my-bucket`.

```
aws s3api get-bucket-policy-status \
  --bucket my-bucket
```

Saída:

```
{
  "PolicyStatus": {
    "IsPublic": false
  }
}
```

```
}
```

- Para obter detalhes da API, consulte [GetBucketPolicyStatus](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando retorna o status da política do bucket do S3 em questão, indicando se o bucket é público.

```
Get-S3BucketPolicyStatus -BucketName 's3casetestbucket'
```

- Para obter detalhes da API, consulte [GetBucketPolicyStatus](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketReplication** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `GetBucketReplication`.

CLI

AWS CLI

O seguinte comando recupera a configuração de replicação de um bucket denominado `my-bucket`:

```
aws s3api get-bucket-replication --bucket my-bucket
```

Saída:

```
{
```

```
"ReplicationConfiguration": {
  "Rules": [
    {
      "Status": "Enabled",
      "Prefix": "",
      "Destination": {
        "Bucket": "arn:aws:s3:::my-bucket-backup",
        "StorageClass": "STANDARD"
      },
      "ID": "ZmUwNzE4ZmQ4tMjVhOS00MTlkLOGI4NDkzZTIWJjNTUtYTA1"
    }
  ],
  "Role": "arn:aws:iam::123456789012:role/s3-replication-role"
}
```

- Para obter detalhes da API, consulte [GetBucketReplication](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: retorna as informações da configuração de replicação definida no bucket denominado “mybucket”.

```
Get-S3BucketReplication -BucketName mybucket
```

- Para obter detalhes da API, consulte [GetBucketReplication](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketRequestPayment** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `GetBucketRequestPayment`.

CLI

AWS CLI

Para recuperar a configuração de pagamento de solicitação de um bucket

O exemplo `get-bucket-request-payment` a seguir recupera a configuração de pagamento de solicitação do bucket especificado.

```
aws s3api get-bucket-request-payment \
  --bucket my-bucket
```

Saída:

```
{
  "Payer": "BucketOwner"
}
```

- Para obter detalhes da API, consulte [GetBucketRequestPayment](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: retorna as informações da configuração de pagamento de solicitação do bucket denominado “mybucket”. Por padrão, o proprietário do bucket paga pelos downloads feitos no bucket.

```
Get-S3BucketRequestPayment -BucketName mybucket
```

- Para obter detalhes da API, consulte [GetBucketRequestPayment](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketTagging** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `GetBucketTagging`.

CLI

AWS CLI

O seguinte comando recupera a configuração de marcação de um bucket denominado `my-bucket`:

```
aws s3api get-bucket-tagging --bucket my-bucket
```

Saída:

```
{
  "TagSet": [
    {
      "Value": "marketing",
      "Key": "organization"
    }
  ]
}
```

- Para obter detalhes da API, consulte [GetBucketTagging](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando retorna todas as tags associadas ao bucket em questão.

```
Get-S3BucketTagging -BucketName 's3casetestbucket'
```

- Para obter detalhes da API, consulte [GetBucketTagging](#), na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketVersioning** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `GetBucketVersioning`.

CLI

AWS CLI

O seguinte comando recupera a configuração de versionamento de um bucket denominado `my-bucket`:

```
aws s3api get-bucket-versioning --bucket my-bucket
```

Saída:

```
{
  "Status": "Enabled"
}
```

- Para obter detalhes da API, consulte [GetBucketVersioning](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando retorna o status de versionamento referente ao bucket em questão.

```
Get-S3BucketVersioning -BucketName 's3testbucket'
```

- Para obter detalhes da API, consulte [GetBucketVersioning](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetBucketWebsite** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetBucketWebsite`.

.NET

AWS SDK for .NET

Note


Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
// Get the website configuration.
GetBucketWebsiteRequest getRequest = new
GetBucketWebsiteRequest()
{
    BucketName = bucketName,
};
GetBucketWebsiteResponse getResponse = await
client.GetBucketWebsiteAsync(getRequest);
Console.WriteLine($"Index document:
{getResponse.WebsiteConfiguration.IndexDocumentSuffix}");
Console.WriteLine($"Error document:
{getResponse.WebsiteConfiguration.ErrorDocument}");
```

- Para obter detalhes da API, consulte [GetBucketWebsite](#) na Referência da API AWS SDK for .NET.

C++

SDK para C++

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
bool AwsDoc::S3::GetWebsiteConfig(const Aws::String &bucketName,
                                  const Aws::Client::ClientConfiguration
&clientConfig) {
    Aws::S3::S3Client s3_client(clientConfig);

    Aws::S3::Model::GetBucketWebsiteRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::GetBucketWebsiteOutcome outcome =
        s3_client.GetBucketWebsite(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();

        std::cerr << "Error: GetBucketWebsite: "
                  << err.GetMessage() << std::endl;
    }
    else {
        Aws::S3::Model::GetBucketWebsiteResult websiteResult =
outcome.GetResult();

        std::cout << "Success: GetBucketWebsite: "
                  << std::endl << std::endl
                  << "For bucket '" << bucketName << "':"
                  << std::endl
                  << "Index page : "
                  << websiteResult.GetIndexDocument().GetSuffix()
                  << std::endl
                  << "Error page: "
                  << websiteResult.GetErrorDocument().GetKey()
                  << std::endl;
    }
}
```



```
    return outcome.IsSuccess();  
}
```

- Para obter detalhes da API, consulte [GetBucketWebsite](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

O seguinte comando recupera a configuração estática do site do bucket my-bucket:

```
aws s3api get-bucket-website --bucket my-bucket
```

Saída:

```
{  
  "IndexDocument": {  
    "Suffix": "index.html"  
  },  
  "ErrorDocument": {  
    "Key": "error.html"  
  }  
}
```

- Para obter detalhes da API, consulte [GetBucketWebsite](#) na Referência de comandos da AWS CLI.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Obtenha a configuração do site.

```
import { GetBucketWebsiteCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new GetBucketWebsiteCommand({
    Bucket: "test-bucket",
  });

  try {
    const { ErrorDocument, IndexDocument } = await client.send(command);
    console.log(
      `Your bucket is set up to host a website. It has an error document:`,
      `${ErrorDocument.Key}, and an index document: ${IndexDocument.Suffix}.`,
    );
  } catch (err) {
    console.error(err);
  }
};
```

- Para obter detalhes da API, consulte [GetBucketWebsite](#) na Referência da API AWS SDK for JavaScript.

PowerShell

Tools for PowerShell

Exemplo 1: este comando retorna os detalhes das configurações do site estático do bucket do S3 em questão.

```
Get-S3BucketWebsite -BucketName 's3testbucket'
```

- Para obter detalhes da API, consulte [GetBucketWebsite](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetObject** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `GetObject`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- [Obter um objeto de um bucket que foi modificado](#)
- [Obter um objeto de um ponto de acesso multirregional](#)
- [Conceitos básicos de buckets e objetos](#)
- [Conceitos básicos de criptografia](#)
- [Monitorar uploads e downloads](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <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($"{filePath}\
\{objectName}", true, CancellationToken.None);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error saving {objectName}: {ex.Message}");
        return false;
    }
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API AWS SDK for .NET.

Bash

AWS CLI com script Bash

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function download_object_from_bucket
#
# This function downloads an object in a bucket to a file.
#
# Parameters:
#     $1 - The name of the bucket to download the object from.
#     $2 - The path and file name to store the downloaded bucket.
#     $3 - The key (name) of the object in the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function download_object_from_bucket() {
    local bucket_name=$1
    local destination_file_name=$2
    local object_name=$3
    local response

    response=$(aws s3api get-object \
        --bucket "$bucket_name" \
        --key "$object_name" \
```

```
"$destination_file_name")

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
fi
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência de comandos da AWS CLI.

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
bool AwsDoc::S3::GetObject(const Aws::String &objectKey,
                           const Aws::String &fromBucket,
                           const Aws::Client::ClientConfiguration &clientConfig)
{
    Aws::S3::S3Client client(clientConfig);

    Aws::S3::Model::GetObjectRequest request;
    request.SetBucket(fromBucket);
    request.SetKey(objectKey);

    Aws::S3::Model::GetObjectOutcome outcome =
        client.GetObject(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: GetObject: " <<
            err.GetExceptionName() << ": " << err.GetMessage() <<
        std::endl;
    }
    else {
```

```
        std::cout << "Successfully retrieved '" << objectKey << "' from '"  
                << fromBucket << "'." << std::endl;  
    }  
  
    return outcome.IsSuccess();  
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

O seguinte exemplo usa o comando `get-object` para fazer download de um objeto do Amazon S3:

```
aws s3api get-object --bucket text-content --key dir/my_images.tar.bz2  
my_images.tar.bz2
```

Observe que o parâmetro `outfile` é especificado sem um nome de opção, como `--outfile`. O nome do arquivo de saída deve ser o último parâmetro no comando.

O exemplo abaixo demonstra o uso de `--range` para fazer download de um intervalo de bytes específico de um objeto. Observe que os intervalos de bytes precisam ser prefixados com `bytes=`:


```
aws s3api get-object --bucket text-content --key dir/my_data --range  
bytes=8888-9999 my_data_range
```

Para obter mais informações sobre como recuperar objetos, consulte [Obter um objeto](#) no Guia do desenvolvedor do Amazon S3.

- Para obter detalhes da API, consulte [GetObject](#) na Referência de comandos da AWS CLI.

Go

SDK para Go V2

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
// actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// DownloadFile gets an object from a bucket and stores it in a local file.
func (basics BucketBasics) DownloadFile(bucketName string, objectKey string,
    fileName string) error {
    result, err := basics.S3Client.GetObject(context.TODO(), &s3.GetObjectInput{
        Bucket: aws.String(bucketName),
        Key:    aws.String(objectKey),
    })
    if err != nil {
        log.Printf("Couldn't get object %v:%v. Here's why: %v\n", bucketName,
            objectKey, err)
        return err
    }
    defer result.Body.Close()
    file, err := os.Create(fileName)
    if err != nil {
        log.Printf("Couldn't create file %v. Here's why: %v\n", fileName, err)
        return err
    }
    defer file.Close()
```



```
body, err := io.ReadAll(result.Body)
if err != nil {
    log.Printf("Couldn't read object body from %v. Here's why: %v\n", objectKey,
err)
}
_, err = file.Write(body)
return err
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API AWS SDK for Go.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Leia dados como uma matriz de bytes usando um [S3Client](#).

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 */
```

```
* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
*/

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

            Usage:
                <bucketName> <keyName> <path>

            Where:
                bucketName - The Amazon S3 bucket name.\s
                keyName - The key name.\s
                path - The path where the file is written to.\s
            """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String keyName = args[1];
        String path = args[2];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        getObjectBytes(s3, bucketName, keyName, path);
    }

    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);
}
}
```

Use [S3TransferManager](#) para [fazer download de um objeto](#) em um bucket do S3 para um arquivo local. Veja o [arquivo completo](#) e [teste](#).

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedFileDownload;
import software.amazon.awssdk.transfer.s3.model.DownloadFileRequest;
import software.amazon.awssdk.transfer.s3.model.FileDownload;
import software.amazon.awssdk.transfer.s3.progress.LoggingTransferListener;

import java.io.IOException;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.UUID;

public Long downloadFile(S3TransferManager transferManager, String
bucketName,

String key, String downloadedFilePath) {
```

```

        DownloadFileRequest downloadFileRequest = DownloadFileRequest.builder()
            .getObjectRequest(b -> b.bucket(bucketName).key(key))
            .destination(Paths.get(downloadedFilePath))
            .build();

        FileDownload downloadFile =
transferManager.downloadFile(downloadFileRequest);

        CompletedFileDownload downloadResult =
downloadFile.completionFuture().join();
        logger.info("Content length [{}]",
downloadResult.response().contentType());
        return downloadResult.response().contentType();
    }

```

Leia etiquetas que pertencem a um objeto usando um [S3Client](#).

```

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectTaggingRequest;
import software.amazon.awssdk.services.s3.model.GetObjectTaggingResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.Tag;
import java.util.List;

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

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

            Usage:
                <bucketName> <keyName>\s

```

```
        Where:
            bucketName - The Amazon S3 bucket name.\s
            keyName - A key name that represents the object.\s
        """;

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

    String bucketName = args[0];
    String keyName = args[1];
    Region region = Region.US_EAST_1;
    S3Client s3 = S3Client.builder()
        .region(region)
        .build();

    listTags(s3, bucketName, keyName);
    s3.close();
}

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 para um objeto usando um [S3Client](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetUrlRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.net.URL;

/**
 * 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 GetObjectUrl {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <bucketName> <keyName>\s

            Where:
                bucketName - The Amazon S3 bucket name.
                keyName - A key name that represents the object.\s
            """;

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

        String bucketName = args[0];
        String keyName = args[1];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();
```

```
        getURL(s3, bucketName, keyName);
        s3.close();
    }

    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 usando um [S3Client](#).

```
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.time.Duration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import
    software.amazon.awssdk.services.s3.presigner.model.GetObjectPresignRequest;
import
    software.amazon.awssdk.services.s3.presigner.model.PresignedGetObjectRequest;
import software.amazon.awssdk.services.s3.presigner.S3Presigner;
import software.amazon.awssdk.utils.IoUtils;

/**
 * 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 GetObjectPresignedUrl {
    public static void main(String[] args) {
        final String USAGE = ""

            Usage:
                <bucketName> <keyName>\s

            Where:
                bucketName - The Amazon S3 bucket name.\s
                keyName - A key name that represents a text file.\s
            """;

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

        String bucketName = args[0];
        String keyName = args[1];
        Region region = Region.US_EAST_1;
        S3Presigner presigner = S3Presigner.builder()
            .region(region)
            .build();

        getPresignedUrl(presigner, bucketName, keyName);
        presigner.close();
    }

    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();
}
}
```

Obtenha um objeto usando um objeto ResponseTransformer e o [S3Client](#).

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.core.sync.ResponseTransformer;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;

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

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

            Usage:
                <bucketName> <keyName> <path>

            Where:
                bucketName - The Amazon S3 bucket name.\s
                keyName - The key name.\s
                path - The path where the file is written to.\s
            """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String keyName = args[1];
```

```
String path = args[2];
Region region = Region.US_EAST_1;
S3Client s3 = S3Client.builder()
    .region(region)
    .build();

getObjectBytes(s3, bucketName, keyName, path);
s3.close();
}

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.getObject(objectRequest, ResponseTransformer.toBytes());
        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);
    }
}
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Baixe o objeto.

```
import { GetObjectCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new GetObjectCommand({
    Bucket: "test-bucket",
    Key: "hello-s3.txt",
  });

  try {
    const response = await client.send(command);
    // The Body object also has 'transformToByteArray' and 'transformToWebStream'
    methods.
    const str = await response.Body.transformToString();
    console.log(str);
  } catch (err) {
    console.error(err);
  }
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).
- Para obter detalhes da API, consulte [GetObject](#) na Referência da API AWS SDK for JavaScript.

Kotlin

SDK for Kotlin

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK for Kotlin.

PHP

SDK para PHP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Obtenha um objeto.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

try {
    $file = $this->s3client->getObject([
        'Bucket' => $this->bucketName,
        'Key' => $fileName,
    ]);
    $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 $fileName from $this->bucketName with error:
" . $exception->getMessage();
    exit("Please fix error with file downloading before continuing.");
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API AWS SDK for PHP.

PowerShell

Tools for PowerShell

Exemplo 1: este comando recupera o item “sample.txt” do bucket “test-files” e o salva em um arquivo chamado “local-sample.txt” no local atual. O arquivo “local-sample.txt” não precisa existir antes de esse comando ser chamado.

```
Read-S3Object -BucketName test-files -Key sample.txt -File local-sample.txt
```

Exemplo 2: este comando recupera o diretório virtual “DIR” do bucket “test-files” e o salva em uma pasta chamada “Local-DIR” no local atual. O arquivo “Local-DIR” não precisa existir antes de esse comando ser chamado.

```
Read-S3Object -BucketName test-files -KeyPrefix DIR -Folder Local-DIR
```

Exemplo 3: baixa todos os objetos com chaves terminadas em “.json” de buckets com “config” no respectivo nome em arquivos na pasta especificada. As chaves do objeto são usadas para definir o nome dos arquivos.

```
Get-S3Bucket | ? { $_.BucketName -like '*config*' } | Get-S3Object | ? { $_.Key -like '*.json' } | Read-S3Object -Folder C:\ConfigObjects
```

- Para obter detalhes da API, consulte [GetObject](#) na AWS Tools for PowerShell Cmdlet Reference.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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

# Example usage:
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 do 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

# Example usage:
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 AWS SDK for Ruby.

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```

async fn get_object(client: Client, opt: Opt) -> Result<usize, anyhow::Error> {
  trace!("bucket:      {}", opt.bucket);
  trace!("object:       {}", opt.object);

```

```
trace!("destination: {}", opt.destination.display());

let mut file = File::create(opt.destination.clone())?;

let mut object = client
    .get_object()
    .bucket(opt.bucket)
    .key(opt.object)
    .send()
    .await?;

let mut byte_count = 0_usize;
while let Some(bytes) = object.body.try_next().await? {
    let bytes_len = bytes.len();
    file.write_all(&bytes)?;
    trace!("Intermediate write of {bytes_len}");
    byte_count += bytes_len;
}

Ok(byte_count)
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API AWS SDK para Rust.

SAP ABAP

SDK para SAP ABAP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
TRY.
    oo_result = lo_s3->getobject(           " oo_result is returned for
testing purposes. "
        iv_bucket = iv_bucket_name
        iv_key = iv_object_key
    ).
    DATA(lv_object_data) = oo_result->get_body( ).
```

```
MESSAGE 'Object retrieved from S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
MESSAGE 'Bucket does not exist.' TYPE 'E'.
CATCH /aws1/cx_s3_nosuchkey.
MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API AWS SDK para SAP ABAP.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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,
```

```
        let data = try await body.readData() else {
            return
        }
        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,
        let data = try await body.readData() else {
        return "".data(using: .utf8)!
    }

    return data
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API do AWS SDK para Swift.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetObjectAcl** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetObjectAcl`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Gerenciar listas de controle de acesso \(ACLs\)](#)

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
bool AwsDoc::S3::GetObjectAcl(const Aws::String &bucketName,
                              const Aws::String &objectKey,
                              const Aws::Client::ClientConfiguration
&clientConfig) {
    Aws::S3::S3Client s3_client(clientConfig);

    Aws::S3::Model::GetObjectAclRequest request;
    request.SetBucket(bucketName);
    request.SetKey(objectKey);

    Aws::S3::Model::GetObjectAclOutcome outcome =
        s3_client.GetObjectAcl(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: GetObjectAcl: "
            << err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    }
    else {
        Aws::Vector<Aws::S3::Model::Grant> grants =
            outcome.GetResult().GetGrants();

        for (auto it = grants.begin(); it != grants.end(); it++) {
            std::cout << "For object " << objectKey << ": "
                << std::endl << std::endl;

            Aws::S3::Model::Grant grant = *it;
            Aws::S3::Model::Grantee grantee = grant.GetGrantee();
```

```

        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(grant.GetPermission()) <<
            std::endl << std::endl;
    }
}

return outcome.IsSuccess();
}

//! Routine which converts a built-in type enumeration to a human-readable
string.
/*!
 \fn GetGranteeTypeString()
 \param type Type enumeration.
 */

Aws::String GetGranteeTypeString(const Aws::S3::Model::Type &type) {
    switch (type) {

```

```

        case Aws::S3::Model::Type::AmazonCustomerByEmail:
            return "Email address of an AWS account";
        case Aws::S3::Model::Type::CanonicalUser:
            return "Canonical user ID of an AWS account";
        case Aws::S3::Model::Type::Group:
            return "Predefined Amazon S3 group";
        case Aws::S3::Model::Type::NOT_SET:
            return "Not set";
        default:
            return "Type unknown";
    }
}

//! Routine which converts a built-in type enumeration to a human-readable
string.
/*!
 \fn GetPermissionString()
 \param permission Permission enumeration.
 */

Aws::String GetPermissionString(const Aws::S3::Model::Permission &permission) {
    switch (permission) {
        case Aws::S3::Model::Permission::FULL_CONTROL:
            return "Can read this object's data and its metadata, "
                "and read/write this object's permissions";
        case Aws::S3::Model::Permission::NOT_SET:
            return "Permission not set";
        case Aws::S3::Model::Permission::READ:
            return "Can read this object's data and its metadata";
        case Aws::S3::Model::Permission::READ_ACP:
            return "Can read this object's permissions";
            // case Aws::S3::Model::Permission::WRITE // Not applicable.
        case Aws::S3::Model::Permission::WRITE_ACP:
            return "Can write this object's permissions";
        default:
            return "Permission unknown";
    }
}
}

```

- Para obter detalhes da API, consulte [GetObjectAcl](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

O seguinte comando recupera a lista de controle de acesso de um objeto em um bucket chamado `my-bucket`:

```
aws s3api get-object-acl --bucket my-bucket --key index.html
```

Saída:

```
{
  "Owner": {
    "DisplayName": "my-username",
    "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
  },
  "Grants": [
    {
      "Grantee": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
      },
      "Permission": "FULL_CONTROL"
    },
    {
      "Grantee": {
        "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
      },
      "Permission": "READ"
    }
  ]
}
```

- Para obter detalhes da API, consulte [GetObjectAcl](#) na Referência de comandos da AWS CLI.

Kotlin

SDK for Kotlin

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Kotlin.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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)
        raise
    else:
        return acl
```

- Para obter detalhes da API, consulte [GetObjectAcl](#) na Referência da API AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar `GetObjectLegalHold` com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetObjectLegalHold`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- [Obter a configuração de retenção legal de um objeto](#)
- [Bloquear objetos do Amazon S3](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <summary>
/// Get the legal hold details for an S3 object.
/// </summary>
/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The object key.</param>
/// <returns>The object legal hold details.</returns>
public async Task<ObjectLockLegalHold> GetObjectLegalHold(string bucketName,
    string objectKey)
{
    try
    {
        var request = new GetObjectLegalHoldRequest()
        {
            BucketName = bucketName,
            Key = objectKey
        };

        var response = await _amazonS3.GetObjectLegalHoldAsync(request);
```

```
        Console.WriteLine($"{\tObject legal hold for {objectKey} in
{bucketName}: " +
                        $"\n\tStatus: {response.LegalHold.Status}");
        return response.LegalHold;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"{\tUnable to fetch legal hold: '{ex.Message}'");
        return new ObjectLockLegalHold();
    }
}
```

- Consulte detalhes da API em [GetObjectLegalHold](#) na Referência de API do AWS SDK for .NET.

CLI

AWS CLI

Recupera o status de “Retenção jurídica” de um objeto

O exemplo de `get-object-legal-hold` a seguir recupera o status de “Retenção jurídica” do objeto especificado.

```
aws s3api get-object-legal-hold \
  --bucket my-bucket-with-object-lock \
  --key doc1.rtf
```


Saída:

```
{
  "LegalHold": {
    "Status": "ON"
  }
}
```

- Consulte detalhes da API em [GetObjectLegalHold](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
// Get the legal hold details for an S3 object.
public ObjectLockLegalHold getObjectLegalHold(String bucketName, String
objectKey) {
    try {
        GetObjectLegalHoldRequest legalHoldRequest =
GetObjectLegalHoldRequest.builder()
            .bucket(bucketName)
            .key(objectKey)
            .build();

        GetObjectLegalHoldResponse response =
getClient().getObjectLegalHold(legalHoldRequest);
        System.out.println("Object legal hold for " + objectKey + " in " +
bucketName +
            ":\n\tStatus: " + response.legalHold().status());
        return response.legalHold();

    } catch (S3Exception ex) {
        System.out.println("\tUnable to fetch legal hold: '" +
ex.getMessage() + "'");
    }

    return null;
}
```

- Consulte detalhes da API em [GetObjectLegalHold](#) na Referência de API do AWS SDK for Java 2.x.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar `GetObjectLockConfiguration` com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetObjectLockConfiguration`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Bloquear objetos do Amazon S3](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <summary>
/// Get the object lock configuration details for an S3 bucket.
/// </summary>
/// <param name="bucketName">The bucket to get details.</param>
/// <returns>The bucket's object lock configuration details.</returns>
public async Task<ObjectLockConfiguration>
GetBucketObjectLockConfiguration(string bucketName)
{
    try
    {
        var request = new GetObjectLockConfigurationRequest()
        {
            BucketName = bucketName
        };
    }
}
```

```
        var response = await
        _amazonS3.GetObjectLockConfigurationAsync(request);
        Console.WriteLine($"{\tBucket object lock config for {bucketName} in
{bucketName}: " +
            $"{\n\tEnabled:
{response.ObjectLockConfiguration.ObjectLockEnabled}" +
            $"{\n\tRule:
{response.ObjectLockConfiguration.Rule?.DefaultRetention}");

        return response.ObjectLockConfiguration;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"{\tUnable to fetch object lock config:
'{ex.Message}'");
        return new ObjectLockConfiguration();
    }
}
```

- Consulte detalhes da API em [GetObjectLockConfiguration](#) na Referência de API do AWS SDK for .NET.

CLI

AWS CLI

Como recuperar uma configuração de bloqueio de objetos de um bucket

O exemplo de `get-object-lock-configuration` a seguir recupera a configuração de bloqueio de objetos do bucket especificado.

```
aws s3api get-object-lock-configuration \
    --bucket my-bucket-with-object-lock
```

Saída:

```
{
  "ObjectLockConfiguration": {
    "ObjectLockEnabled": "Enabled",
    "Rule": {
```



```
        "DefaultRetention": {
            "Mode": "COMPLIANCE",
            "Days": 50
        }
    }
}
```

- Consulte detalhes da API em [GetObjectLockConfiguration](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
// Get the object lock configuration details for an S3 bucket.
public void getBucketObjectLockConfiguration(String bucketName) {
    GetObjectLockConfigurationRequest objectLockConfigurationRequest =
GetObjectLockConfigurationRequest.builder()
        .bucket(bucketName)
        .build();

    GetObjectLockConfigurationResponse response =
getClient().getObjectLockConfiguration(objectLockConfigurationRequest);
    System.out.println("Bucket object lock config for "+bucketName+": ");
    System.out.println("\tEnabled:
"+response.getObjectLockConfiguration().objectLockEnabled());
    System.out.println("\tRule: "+
response.getObjectLockConfiguration().rule().defaultRetention());
}
```

- Consulte detalhes da API em [GetObjectLockConfiguration](#) na Referência de API do AWS SDK for Java 2.x.

PowerShell

Tools for PowerShell

Exemplo 1: se a configuração do Bloqueio de Objetos estiver habilitada para o bucket do S3 em questão, este comando retornará o valor “Habilitado”.

```
Get-S3ObjectLockConfiguration -BucketName 's3buckettesting' -Select  
ObjectLockConfiguration.ObjectLockEnabled
```

Saída:

```
Value  
-----  
Enabled
```

- Para obter detalhes da API, consulte [GetObjectLockConfiguration](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetObjectRetention** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `GetObjectRetention`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Bloquear objetos do Amazon S3](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <summary>
/// Get the retention period for an S3 object.
/// </summary>
/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The object key.</param>
/// <returns>The object retention details.</returns>
public async Task<ObjectLockRetention> GetObjectRetention(string bucketName,
    string objectKey)
{
    try
    {
        var request = new GetObjectRetentionRequest()
        {
            BucketName = bucketName,
            Key = objectKey
        };

        var response = await _amazonS3.GetObjectRetentionAsync(request);
        Console.WriteLine($"\\tObject retention for {objectKey} in
{bucketName}: " +
            $"\\n\\t{response.Retention.Mode} until
{response.Retention.RetainUntilDate:d}.");
        return response.Retention;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"\\tUnable to fetch object lock retention:
'{ex.Message}'");
        return new ObjectLockRetention();
    }
}
```

- Consulte detalhes da API em [GetObjectRetention](#) na Referência de API do AWS SDK for .NET.

CLI

AWS CLI

Como recuperar a configuração de retenção de objetos de um objeto

O exemplo de `get-object-retention` a seguir recupera a configuração de retenção de objetos do objeto especificado.

```
aws s3api get-object-retention \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf
```

Saída:

```
{  
  "Retention": {  
    "Mode": "GOVERNANCE",  
    "RetainUntilDate": "2025-01-01T00:00:00.000Z"  
  }  
}
```

- Consulte detalhes da API em [GetObjectRetention](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
// Get the retention period for an S3 object.  
public ObjectLockRetention getObjectRetention(String bucketName, String key){
```

```
try {
    GetObjectRetentionRequest retentionRequest =
GetObjectRetentionRequest.builder()
        .bucket(bucketName)
        .key(key)
        .build();

    GetObjectRetentionResponse response =
getClient().getObjectRetention(retentionRequest);
    System.out.println("tObject retention for "+key +"
in "+ bucketName +": " + response.retention().mode() +" until "+
response.retention().retainUntilDate() +".");
    return response.retention();

} catch (S3Exception e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    return null;
}
}
```

- Consulte detalhes da API em [GetObjectRetention](#) na Referência de API do AWS SDK for Java 2.x.

PowerShell

Tools for PowerShell

Exemplo 1: o comando retorna o modo e a data até a qual o objeto ficará retido.

```
Get-S3ObjectRetention -BucketName 's3buckettesting' -Key 'testfile.txt'
```

- Para obter detalhes da API, consulte [GetObjectRetention](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetObjectTagging** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetObjectTagging`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Conceitos básicos de etiquetas](#)

CLI

AWS CLI

Para recuperar as tags anexadas a um objeto

O exemplo `get-object-tagging` a seguir recupera os valores da chave especificada do objeto especificado.

```
aws s3api get-object-tagging \  
  --bucket my-bucket \  
  --key doc1.rtf
```

Saída:

```
{  
  "TagSet": [  
    {  
      "Value": "confidential",  
      "Key": "designation"  
    }  
  ]  
}
```

O exemplo `get-object-tagging` a seguir tenta recuperar os conjuntos de tags do objeto `doc2.rtf`, o qual não tem tags.

```
aws s3api get-object-tagging \  
  --bucket my-bucket \  
  --key doc2.rtf
```

Saída:

```
{
  "TagSet": []
}
```

O exemplo `get-object-tagging` a seguir recupera os conjuntos de tags do objeto `doc3.rtf`, o qual não tem tags.

```
aws s3api get-object-tagging \
  --bucket my-bucket \
  --key doc3.rtf
```

Saída:

```
{
  "TagSet": [
    {
      "Value": "confidential",
      "Key": "designation"
    },
    {
      "Value": "finance",
      "Key": "department"
    },
    {
      "Value": "payroll",
      "Key": "team"
    }
  ]
}
```

- Para obter detalhes da API, consulte [GetObjectTagging](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: o exemplo retorna as tags associadas ao objeto presente no bucket do S3 em questão.

```
Get-S3ObjectTagSet -Key 'testfile.txt' -BucketName 'testbucket123'
```

Saída:

```
Key Value
--- -----
test value
```

- Para obter detalhes da API, consulte [GetObjectTagging](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **GetPublicAccessBlock** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `GetPublicAccessBlock`.

CLI

AWS CLI

Para definir ou modificar a configuração do Bloqueio de Acesso Público de um bucket

O exemplo `get-public-access-block` a seguir exibe a configuração do Bloqueio de Acesso Público do bucket especificado.

```
aws s3api get-public-access-block \  
  --bucket my-bucket
```

Saída:

```
{  
  "PublicAccessBlockConfiguration": {  
    "IgnorePublicAcls": true,  
    "BlockPublicPolicy": true,  
    "BlockPublicAcls": true,
```



```
    "RestrictPublicBuckets": true
  }
}
```

- Para obter detalhes da API, consulte [GetPublicAccessBlock](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: o comando retorna a configuração do Bloqueio de Acesso Público do bucket do S3 em questão.

```
Get-S3PublicAccessBlock -BucketName 's3testbucket'
```

- Para obter detalhes da API, consulte [GetPublicAccessBlock](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **HeadBucket** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o HeadBucket.

Bash

AWS CLI com script Bash

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
#####  
# function bucket_exists
```

```

#
# This function checks to see if the specified bucket already exists.
#
# Parameters:
#     $1 - The name of the bucket to check.
#
# Returns:
#     0 - If the bucket already exists.
#     1 - If the bucket doesn't exist.
#####
function bucket_exists() {
    local bucket_name
    bucket_name=$1

    # Check whether the bucket already exists.
    # We suppress all output - we're interested only in the return code.

    if aws s3api head-bucket \
        --bucket "$bucket_name" \
        >/dev/null 2>&1; then
        return 0 # 0 in Bash script means true.
    else
        return 1 # 1 in Bash script means false.
    fi
}

```

- Para obter detalhes da API, consulte [HeadBucket](#) na Referência de comandos da AWS CLI.

CLI

AWS CLI

O seguinte comando verifica o acesso ao bucket `my-bucket`:

```
aws s3api head-bucket --bucket my-bucket
```

Se o bucket existir e você tiver acesso a ele, nenhuma saída será retornada. Caso contrário, uma mensagem de erro será exibida. Por exemplo:

```
A client error (404) occurred when calling the HeadBucket operation: Not Found
```

- Para obter detalhes da API, consulte [HeadBucket](#) na Referência de comandos da AWS CLI.

Go

SDK para Go V2

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
// actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// BucketExists checks whether a bucket exists in the current account.
func (basics BucketBasics) BucketExists(bucketName string) (bool, error) {
    _, err := basics.S3Client.HeadBucket(context.TODO(), &s3.HeadBucketInput{
        Bucket: aws.String(bucketName),
    })
    exists := true
    if err != nil {
        var apiError smithy.APIError
        if errors.As(err, &apiError) {
            switch apiError.(type) {
            case *types.NotFound:
                log.Printf("Bucket %v is available.\n", bucketName)
                exists = false
                err = nil
            default:
                log.Printf("Either you don't have access to bucket %v or another error
occurred. "+
```

```
        "Here's what happened: %v\n", bucketName, err)
    }
} else {
    log.Printf("Bucket %v exists and you already own it.", bucketName)
}

return exists, err
}
```

- Para obter detalhes da API, consulte [HeadBucket](#) na Referência da API AWS SDK for Go.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **HeadObject** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o HeadObject.

CLI

AWS CLI

O seguinte comando recupera metadados de um objeto no bucket my-bucket:

```
aws s3api head-object --bucket my-bucket --key index.html
```

Saída:

```
{
  "AcceptRanges": "bytes",
  "ContentType": "text/html",
```

```
"LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
"ContentLength": 77,
"VersionId": "null",
"ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
"Metadata": {}
}
```

- Para obter detalhes da API, consulte [HeadObject](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Determine o tipo de conteúdo de um objeto.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.HeadObjectRequest;
import software.amazon.awssdk.services.s3.model.HeadObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;

/**
 * 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 GetObjectContentType {
    public static void main(String[] args) {
        final String usage = ""

                Usage:
                <bucketName> <keyName>>
```

```
        Where:
            bucketName - The Amazon S3 bucket name.\s
            keyName - The key name.\s
        """;

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

    String bucketName = args[0];
    String keyName = args[1];
    Region region = Region.US_EAST_1;
    S3Client s3 = S3Client.builder()
        .region(region)
        .build();

    getContentType(s3, bucketName, keyName);
    s3.close();
}

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);
    }
}
}
```

Obtenha o status de restauração de um objeto.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.HeadObjectRequest;
import software.amazon.awssdk.services.s3.model.HeadObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;

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

            Usage:
                <bucketName> <keyName>\s

            Where:
                bucketName - The Amazon S3 bucket name.\s
                keyName - A key name that represents the object.\s
            """;

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

        String bucketName = args[0];
        String keyName = args[1];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        checkStatus(s3, bucketName, keyName);
        s3.close();
    }

    public static void checkStatus(S3Client s3, String bucketName, String
keyName) {
        try {
            HeadObjectRequest headObjectRequest = HeadObjectRequest.builder()
                .bucket(bucketName)
                .key(keyName)
                .build();

            HeadObjectResponse response = s3.headObject(headObjectRequest);
```



```
        System.out.println("The Amazon S3 object restoration status is " +
response.restore());

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- Para obter detalhes da API, consulte [HeadObject](#) em Referência da API AWS SDK for Java 2.x.

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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
end

# Example usage:
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 AWS SDK for Ruby.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **ListBucketAnalyticsConfigurations** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `ListBucketAnalyticsConfigurations`.

CLI

AWS CLI

Como recuperar uma lista de configurações de análise de um bucket

O exemplo `list-bucket-analytics-configurations` a seguir recupera uma lista de configurações de análise do bucket especificado.

```
aws s3api list-bucket-analytics-configurations \
```

```
--bucket my-bucket
```

Saída:

```
{
  "AnalyticsConfigurationList": [
    {
      "StorageClassAnalysis": {},
      "Id": "1"
    }
  ],
  "IsTruncated": false
}
```

- Para obter detalhes da API, consulte [ListBucketAnalyticsConfigurations](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando retorna as cem primeiras análises do bucket do S3 em questão.

```
Get-S3BucketAnalyticsConfigurationList -BucketName 's3casetestbucket'
```

- Para obter detalhes da API, consulte [ListBucketAnalyticsConfigurations](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **ListBucketInventoryConfigurations** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `ListBucketInventoryConfigurations`.

CLI

AWS CLI

Como recuperar uma lista de configurações de inventário de um bucket

O exemplo `list-bucket-inventory-configurations` a seguir recupera as configurações de inventário do bucket especificado.

```
aws s3api list-bucket-inventory-configurations \  
  --bucket my-bucket
```

Saída:

```
{  
  "InventoryConfigurationList": [  
    {  
      "IsEnabled": true,  
      "Destination": {  
        "S3BucketDestination": {  
          "Format": "ORC",  
          "Bucket": "arn:aws:s3:::my-bucket",  
          "AccountId": "123456789012"  
        }  
      },  
      "IncludedObjectVersions": "Current",  
      "Id": "1",  
      "Schedule": {  
        "Frequency": "Weekly"  
      }  
    },  
    {  
      "IsEnabled": true,  
      "Destination": {  
        "S3BucketDestination": {  
          "Format": "CSV",  
          "Bucket": "arn:aws:s3:::my-bucket",  
          "AccountId": "123456789012"  
        }  
      },  
      "IncludedObjectVersions": "Current",  
      "Id": "2",  
      "Schedule": {
```

```
        "Frequency": "Daily"
      }
    }
  ],
  "IsTruncated": false
}
```

- Para obter detalhes da API, consulte [ListBucketInventoryConfigurations](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando retorna as cem primeiras configurações de inventário do bucket do S3 em questão.

```
Get-S3BucketInventoryConfigurationList -BucketName 's3testbucket'
```

- Para obter detalhes da API, consulte [ListBucketInventoryConfigurations](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **ListBuckets** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o ListBuckets.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
namespace ListBucketsExample
{
    using System;
    using System.Collections.Generic;
    using System.Threading.Tasks;
    using Amazon.S3;
    using Amazon.S3.Model;

    /// <summary>
    /// This example uses the AWS SDK for .NET to list the Amazon Simple Storage
    /// Service (Amazon S3) buckets belonging to the default account.
    /// </summary>
    public class ListBuckets
    {
        private static IAmazonS3 _s3Client;

        /// <summary>
        /// Get a list of the buckets owned by the default user.
        /// </summary>
        /// <param name="client">An initialized Amazon S3 client object.</param>
        /// <returns>The response from the ListingBuckets call that contains a
        /// list of the buckets owned by the default user.</returns>
        public static async Task<ListBucketsResponse> GetBuckets(IAmazonS3
client)
        {
            return await client.ListBucketsAsync();
        }

        /// <summary>
        /// This method lists the name and creation date for the buckets in
        /// the passed List of S3 buckets.
        /// </summary>
        /// <param name="bucketList">A List of S3 bucket objects.</param>
        public static void DisplayBucketList(List<S3Bucket> bucketList)
        {
            bucketList
                .ForEach(b => Console.WriteLine($"Bucket name: {b.BucketName},
created on: {b.CreationDate}"));
        }

        public static async Task Main()
        {
            // The client uses the AWS Region of the default user.
        }
    }
}
```

```
        // If the Region where the buckets were created is different,
        // pass the Region to the client constructor. For example:
        // _s3Client = new AmazonS3Client(RegionEndpoint.USEast1);
        _s3Client = new AmazonS3Client();
        var response = await GetBuckets(_s3Client);
        DisplayBucketList(response.Buckets);
    }
}
```

- Para obter detalhes da API, consulte [ListBuckets](#) na Referência da API AWS SDK for .NET.

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
bool AwsDoc::S3::ListBuckets(const Aws::Client::ClientConfiguration
&clientConfig) {
    Aws::S3::S3Client client(clientConfig);

    auto outcome = client.ListBuckets();

    bool result = true;
    if (!outcome.IsSuccess()) {
        std::cerr << "Failed with error: " << outcome.GetError() << std::endl;
        result = false;
    }
    else {
        std::cout << "Found " << outcome.GetResult().GetBuckets().size() << "
buckets\n";
        for (auto &&b: outcome.GetResult().GetBuckets()) {
            std::cout << b.GetName() << std::endl;
        }
    }
}
```

```
    return result;
}
```

- Para obter detalhes da API, consulte [ListBuckets](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

O seguinte comando usa o comando `list-buckets` para exibir os nomes de todos os buckets do Amazon S3 (em todas as regiões):

```
aws s3api list-buckets --query "Buckets[].Name"
```

A opção de consulta filtra a saída de `list-buckets` até sobrar apenas os nomes dos buckets.

Para obter mais informações sobre buckets, consulte [Working with Amazon S3 Buckets](#) no Guia do desenvolvedor do Amazon S3.

- Para obter detalhes da API, consulte [ListBuckets](#) na Referência de comandos da AWS CLI.

Go

SDK para Go V2

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
// actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
```



```
type BucketBasics struct {
    S3Client *s3.Client
}

// ListBuckets lists the buckets in the current account.
func (basics BucketBasics) ListBuckets() ([]types.Bucket, error) {
    result, err := basics.S3Client.ListBuckets(context.TODO(),
        &s3.ListBucketsInput{})
    var buckets []types.Bucket
    if err != nil {
        log.Printf("Couldn't list buckets for your account. Here's why: %v\n", err)
    } else {
        buckets = result.Buckets
    }
    return buckets, err
}
```

- Para obter detalhes da API, consulte [ListBuckets](#) na Referência da API AWS SDK for Go.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.Bucket;
import software.amazon.awssdk.services.s3.model.ListBucketsResponse;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
```

```
*
* For more information, see the following documentation topic:
*
* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
*/
public class ListBuckets {
    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        listAllBuckets(s3);
    }
    public static void listAllBuckets(S3Client s3) {
        ListBucketsResponse response = s3.listBuckets();
        List<Bucket> bucketList = response.buckets();
        for (Bucket bucket: bucketList) {
            System.out.println("Bucket name "+bucket.name());
        }
    }
}
```

- Para obter informações da API, consulte [ListBuckets](#) na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Liste os buckets.

```
import { ListBucketsCommand, S3Client } from "@aws-sdk/client-s3";
```

```
const client = new S3Client({});

export const main = async () => {
  const command = new ListBucketsCommand({});

  try {
    const { Owner, Buckets } = await client.send(command);
    console.log(
      `${Owner.DisplayName} owns ${Buckets.length} bucket${
        Buckets.length === 1 ? "" : "s"
      }:`
    );
    console.log(`${Buckets.map((b) => ` • ${b.Name}`).join("\n")}`);
  } catch (err) {
    console.error(err);
  }
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).
- Para obter detalhes da API, consulte [ListBuckets](#) na Referência da API AWS SDK for JavaScript.

PowerShell

Tools for PowerShell

Exemplo 1: este comando retorna todos os buckets do S3.

```
Get-S3Bucket
```

Exemplo 2: este comando retorna um bucket denominado “test-files”.

```
Get-S3Bucket -BucketName test-files
```

- Para obter informações da API, consulte [ListBuckets](#) na AWS Tools for PowerShell Cmdlet Reference.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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}"
  end
end
```

```
    false
  end
end

# Example usage:
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 AWS SDK for Ruby.

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
async fn show_buckets(strict: bool, client: &Client, region: &str) -> Result<(),
Error> {
  let resp = client.list_buckets().send().await?;
  let buckets = resp.buckets();
  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 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.

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// Return an array containing information about every available bucket.
///
/// - Returns: An array of ``S3ClientTypes.Bucket`` objects describing
/// each bucket.
public func getAllBuckets() async throws -> [S3ClientTypes.Bucket] {
    let output = try await client.listBuckets(input: ListBucketsInput())

    guard let buckets = output.buckets else {
        return []
    }
    return buckets
}
```

- Para obter detalhes da API, consulte [ListBuckets](#) na Referência de API do AWS SDK para Swift.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **ListMultipartUploads** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `ListMultipartUploads`.

CLI

AWS CLI

O seguinte comando lista todos os uploads fracionados ativos do bucket `my-bucket`:

```
aws s3api list-multipart-uploads --bucket my-bucket
```

Saída:

```
{
  "Uploads": [
    {
      "Initiator": {
        "DisplayName": "username",
```



```

        "ID": "arn:aws:iam::0123456789012:user/username"
    },
    "Initiated": "2015-06-02T18:01:30.000Z",
    "UploadId":
"dfRtDYU0WwCCcH43C3WfbkR0NycyCpTJJvxu2i5GYkZ1jF.Yxwh6XG7WfS2vC4to6HiV6Yj1x.cph0gtNBtJ8P3
    "StorageClass": "STANDARD",
    "Key": "multipart/01",
    "Owner": {
        "DisplayName": "aws-account-name",
        "ID":
"100719349fc3b6dcd7c820a124bf7aec408092c3d7b51b38494939801fc248b"
    }
    },
    "CommonPrefixes": []
}

```

Os uploads fracionados em andamento incorrem em custos de armazenamento no Amazon S3. Conclua ou cancele um upload fracionado ativo para remover suas partes da conta.

- Para obter detalhes da API, consulte [ListMultipartUploads](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.ListMultipartUploadsRequest;
import software.amazon.awssdk.services.s3.model.ListMultipartUploadsResponse;
import software.amazon.awssdk.services.s3.model.MultipartUpload;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.util.List;

/**

```

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

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

            Usage:
                <bucketName>\s

            Where:
                bucketName - The name of the Amazon S3 bucket where an in-
progress multipart upload is occurring.
            """;

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

        String bucketName = args[0];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();
        listUploads(s3, bucketName);
        s3.close();
    }

    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 AWS SDK for Java 2.x.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **ListObjectVersions** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `ListObjectVersions`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Trabalhar com objetos versionados](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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

/// <summary>
/// This example lists the versions of the objects in a version enabled
/// Amazon Simple Storage Service (Amazon S3) bucket.
/// </summary>
public class ListObjectVersions
{
    public static async Task Main()
    {
        string bucketName = "doc-example-bucket";

        // If the AWS Region where your bucket is defined is different from
        // the AWS Region where the Amazon S3 bucket is defined, pass the
constant
        // for the AWS Region to the client constructor like this:
        //     var client = new AmazonS3Client(RegionEndpoint.USWest2);
        IAmazonS3 client = new AmazonS3Client();
        await GetObjectListWithAllVersionsAsync(client, bucketName);
    }

    /// <summary>
    /// This method lists all versions of the objects within an Amazon S3
    /// version enabled bucket.
    /// </summary>
    /// <param name="client">The initialized client object used to call
    /// ListVersionsAsync.</param>
    /// <param name="bucketName">The name of the version enabled Amazon S3
bucket
    /// for which you want to list the versions of the contained objects.</
param>
    public static async Task GetObjectListWithAllVersionsAsync(IAmazonS3
client, string bucketName)
    {
        try
        {
            // When you instantiate the ListVersionRequest, you can
            // optionally specify a key name prefix in the request
            // if you want a list of object versions of a specific object.
```

```
// For this example we set a small limit in MaxKeys to return
// a small list of versions.
ListVersionsRequest request = new ListVersionsRequest()
{
    BucketName = bucketName,
    MaxKeys = 2,
};

do
{
    ListVersionsResponse response = await
client.ListVersionsAsync(request);

    // Process response.
    foreach (S3ObjectVersion entry in response.Versions)
    {
        Console.WriteLine($"key: {entry.Key} size:
{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 ex)
{
    Console.WriteLine($"Error: '{ex.Message}'");
}
}
```

- Para obter detalhes da API, consulte [ListObjectVersions](#) na Referência da API AWS SDK for .NET.

CLI

AWS CLI

O seguinte comando recupera as informações da versão de um objeto do bucket my-bucket:

```
aws s3api list-object-versions --bucket my-bucket --prefix index.html
```

Saída:

```
{
  "DeleteMarkers": [
    {
      "Owner": {
        "DisplayName": "my-username",
        "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
      },
      "IsLatest": true,
      "VersionId": "B2VsEK5saUNNHKc0AJj7hIE86RozToyq",
      "Key": "index.html",
      "LastModified": "2015-11-10T00:57:03.000Z"
    },
    {
      "Owner": {
        "DisplayName": "my-username",
        "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
      },
      "IsLatest": false,
      "VersionId": ".FLQEZscLIcfxSq.jsFJ.szUkmng2Yw6",
      "Key": "index.html",
      "LastModified": "2015-11-09T23:32:20.000Z"
    }
  ],
  "Versions": [
    {
      "LastModified": "2015-11-10T00:20:11.000Z",
      "VersionId": "Rb_l2T8UHDkFEwCgJjhlgPOZC0qJ.vpD",
```

```

    "ETag": "\"0622528de826c0df5db1258a23b80be5\"",
    "StorageClass": "STANDARD",
    "Key": "index.html",
    "Owner": {
      "DisplayName": "my-username",
      "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
    },
    "IsLatest": false,
    "Size": 38
  },
  {
    "LastModified": "2015-11-09T23:26:41.000Z",
    "VersionId": "rasWWGpgk9E4s0LyTJgusGeRQKLVIAff",
    "ETag": "\"06225825b8028de826c0df5db1a23be5\"",
    "StorageClass": "STANDARD",
    "Key": "index.html",
    "Owner": {
      "DisplayName": "my-username",
      "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
    },
    "IsLatest": false,
    "Size": 38
  },
  {
    "LastModified": "2015-11-09T22:50:50.000Z",
    "VersionId": "null",
    "ETag": "\"d1f45267a863c8392e07d24dd592f1b9\"",
    "StorageClass": "STANDARD",
    "Key": "index.html",
    "Owner": {
      "DisplayName": "my-username",
      "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
    },
    "IsLatest": false,
    "Size": 533823
  }
]
}

```

- Para obter detalhes da API, consulte [ListObjectVersions](#) na Referência de comandos da AWS CLI.

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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() {
        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 AWS SDK para Rust.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **ListObjects** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o ListObjects.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Criar uma página da web que oferece uma lista de objetos do Amazon S3](#)

CLI

AWS CLI

O seguinte exemplo usa o comando `list-objects` para exibir os nomes de todos os objetos presentes no bucket especificado:

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size: Size}'
```

O exemplo usa o argumento `--query` para filtrar a saída de `list-objects` até encontrar o valor e o tamanho da chave de cada objeto

Para obter mais informações sobre objetos, consulte [Working with Amazon S3 Buckets](#) no Guia do desenvolvedor do Amazon S3.

- Para obter detalhes da API, consulte [ListObjects](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando recupera as informações sobre todos os itens no bucket “test-files”.

```
Get-S3Object -BucketName test-files
```

Exemplo 2: este comando recupera as informações sobre o item “sample.txt” do bucket “test-files”.

```
Get-S3Object -BucketName test-files -Key sample.txt
```

Exemplo 3: este comando recupera as informações sobre todos os itens com prefixo “sample” do bucket “test-files”.

```
Get-S3Object -BucketName test-files -KeyPrefix sample
```

- Consulte detalhes da API em [ListObjects](#) em AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **ListObjectsV2** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o ListObjectsV2.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Conceitos básicos de buckets e objetos](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <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("-----");

        ListObjectsV2Response response;

        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;
    }
}
```

Liste objetos com um paginador.

```
using System;
```

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

/// <summary>
/// The following example lists objects in an Amazon Simple Storage
/// Service (Amazon S3) bucket.
/// </summary>
public class ListObjectsPaginator
{
    private const string BucketName = "doc-example-bucket";

    public static async Task Main()
    {
        IAmazonS3 s3Client = new AmazonS3Client();

        Console.WriteLine($"Listing the objects contained in {BucketName}:
\n");
        await ListingObjectsAsync(s3Client, BucketName);
    }

    /// <summary>
    /// This method uses a paginator to retrieve the list of objects in an
    /// an Amazon S3 bucket.
    /// </summary>
    /// <param name="client">An Amazon S3 client object.</param>
    /// <param name="bucketName">The name of the S3 bucket whose objects
    /// you want to list.</param>
    public static async Task ListingObjectsAsync(IAmazonS3 client, string
bucketName)
    {
        var listObjectsV2Paginator = client.Paginators.ListObjectsV2(new
ListObjectsV2Request
        {
            BucketName = bucketName,
        });

        await foreach (var response in listObjectsV2Paginator.Responses)
        {
            Console.WriteLine($"HttpStatusCode: {response.HttpStatusCode}");
            Console.WriteLine($"Number of Keys: {response.KeyCount}");
            foreach (var entry in response.S3Objects)
            {
                Console.WriteLine($"Key = {entry.Key} Size = {entry.Size}");
            }
        }
    }
}
```

```

    }
  }
}

```

- Para obter detalhes da API, consulte [ListObjectsV2](#) na Referência da API AWS SDK for .NET.

Bash

AWS CLI com script Bash

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and
# size fields from the Contents collection.
#
# Parameters:
#     $1 - The name of the bucket.
#
# Returns:
#     The list of files in text format.

```

```

# And:
# 0 - If successful.
# 1 - If it fails.
#####
function list_items_in_bucket() {
    local bucket_name=$1
    local response

    response=$(aws s3api list-objects \
        --bucket "$bucket_name" \
        --output text \
        --query 'Contents[].{Key: Key, Size: Size}')

    # shellcheck disable=SC2181
    if [[ ${?} -eq 0 ]]; then
        echo "$response"
    else
        errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
        return 1
    fi
}

```

- Para obter detalhes da API, consulte [ListObjectsV2](#) na Referência de comandos da AWS CLI.

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```

bool AwsDoc::S3::ListObjects(const Aws::String &bucketName,
                             const Aws::Client::ClientConfiguration
&clientConfig) {
    Aws::S3::S3Client s3_client(clientConfig);

```

```
Aws::S3::Model::ListObjectsV2Request request;
request.WithBucket(bucketName);

Aws::String continuationToken; // Used for pagination.
Aws::Vector<Aws::S3::Model::Object> allObjects;

do {
    if (!continuationToken.empty()) {
        request.SetContinuationToken(continuationToken);
    }

    auto outcome = s3_client.ListObjectsV2(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error: ListObjects: " <<
            outcome.GetError().GetMessage() << std::endl;
        return false;
    }
    else {
        Aws::Vector<Aws::S3::Model::Object> objects =
            outcome.GetResult().GetContents();

        allObjects.insert(allObjects.end(), objects.begin(), objects.end());
        continuationToken = outcome.GetResult().GetNextContinuationToken();
    }
} while (!continuationToken.empty());

std::cout << allObjects.size() << " object(s) found:" << std::endl;

for (const auto &object: allObjects) {
    std::cout << " " << object.GetKey() << std::endl;
}

return true;
}
```

- Para obter detalhes da API, consulte [ListObjectsV2](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

Como obter uma lista dos objetos em um bucket

O exemplo de `list-objects-v2` a seguir lista os objetos no bucket especificado.

```
aws s3api list-objects-v2 \  
  --bucket my-bucket
```

Saída:

```
{  
  "Contents": [  
    {  
      "LastModified": "2019-11-05T23:11:50.000Z",  
      "ETag": "\"621503c373607d548b37cff8778d992c\"",  
      "StorageClass": "STANDARD",  
      "Key": "doc1.rtf",  
      "Size": 391  
    },  
    {  
      "LastModified": "2019-11-05T23:11:50.000Z",  
      "ETag": "\"a2cecc36ab7c7fe3a71a273b9d45b1b5\"",  
      "StorageClass": "STANDARD",  
      "Key": "doc2.rtf",  
      "Size": 373  
    },  
    {  
      "LastModified": "2019-11-05T23:11:50.000Z",  
      "ETag": "\"08210852f65a2e9cb999972539a64d68\"",  
      "StorageClass": "STANDARD",  
      "Key": "doc3.rtf",  
      "Size": 399  
    },  
    {  
      "LastModified": "2019-11-05T23:11:50.000Z",  
      "ETag": "\"d1852dd683f404306569471af106988e\"",  
      "StorageClass": "STANDARD",  
      "Key": "doc4.rtf",  
      "Size": 6225  
    }  
  ]  
}
```




```
]
}
```

- Para obter detalhes da API, consulte [ListObjectsV2](#) na Referência de comandos da AWS CLI.

Go

SDK para Go V2

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
// actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// ListObjects lists the objects in a bucket.
func (basics BucketBasics) ListObjects(bucketName string) ([]types.Object, error)
{
    result, err := basics.S3Client.ListObjectsV2(context.TODO(),
    &s3.ListObjectsV2Input{
        Bucket: aws.String(bucketName),
    })
    var contents []types.Object
    if err != nil {
        log.Printf("Couldn't list objects in bucket %v. Here's why: %v\n", bucketName,
        err)
    } else {
```

```
    contents = result.Contents
  }
  return contents, err
}
```

- Para obter detalhes da API, consulte [ListObjectsV2](#) na Referência da API AWS SDK for Go.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.ListObjectsRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.S3Object;
import java.util.List;

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

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

        Usage:
```

```
        <bucketName>\s

        Where:
            bucketName - The Amazon S3 bucket from which objects are
read.\s
            """;

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

        String bucketName = args[0];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        listBucketObjects(s3, bucketName);
        s3.close();
    }

    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 " + calcKb(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;
}
}
```

Liste objetos usando paginação.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.paginators.ListObjectsV2Iterable;

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

                Usage:
                <bucketName>\s

                Where:
                bucketName - The Amazon S3 bucket from which objects are
read.\s

                """;

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

        String bucketName = args[0];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        listBucketObjects(s3, bucketName);
        s3.close();
    }

    public static void listBucketObjects(S3Client s3, String bucketName) {
```

```
try {
    ListObjectsV2Request listReq = ListObjectsV2Request.builder()
        .bucket(bucketName)
        .maxKeys(1)
        .build();

    ListObjectsV2Iterable listRes = s3.listObjectsV2Paginator(listReq);
    listRes.stream()
        .flatMap(r -> r.contents().stream())
        .forEach(content -> System.out.println(" Key: " +
content.key() + " size = " + content.size()));

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Para obter detalhes da API, consulte [ListObjectsV2](#) na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Liste todos os objetos no bucket. Se houver mais de um objeto, `IsTruncated` e `NextContinuationToken` serão usados para iterar a lista completa.

```
import {
    S3Client,
    // This command supersedes the ListObjectsCommand and is the recommended way to
    list objects.
    ListObjectsV2Command,
```

```
} from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new ListObjectsV2Command({
    Bucket: "my-bucket",
    // The default and maximum number of keys returned is 1000. This limits it to
    // one for demonstration purposes.
    MaxKeys: 1,
  });

  try {
    let isTruncated = true;

    console.log("Your bucket contains the following objects:\n");
    let contents = "";

    while (isTruncated) {
      const { Contents, IsTruncated, NextContinuationToken } =
        await client.send(command);
      const contentsList = Contents.map((c) => ` • ${c.Key}`).join("\n");
      contents += contentsList + "\n";
      isTruncated = IsTruncated;
      command.input.ContinuationToken = NextContinuationToken;
    }
    console.log(contents);
  } catch (err) {
    console.error(err);
  }
};
```

- Para obter detalhes da API, consulte [ListObjectsV2](#) na Referência da API AWS SDK for JavaScript.

Kotlin

SDK for Kotlin

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 ${myObject.size?.let { calKb(it) }} KBs")
            println("The owner is ${myObject.owner}")
        }
    }
}

private fun calKb(intValue: Long): Long {
    return intValue / 1024
}
```

- Para obter detalhes da API, consulte [ListObjectsV2](#) na Referência da API AWS SDK para Kotlin.

PHP

SDK para PHP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Liste objetos em um bucket.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

try {
    $contents = $this->s3client->listObjectsV2([
        'Bucket' => $this->bucketName,
    ]);
    echo "The contents of your bucket are: \n";
    foreach ($contents['Contents'] as $content) {
        echo $content['Key'] . "\n";
    }
} catch (Exception $exception) {
    echo "Failed to list objects in $this->bucketName with error: " .
    $exception->getMessage();
    exit("Please fix error with listing objects before continuing.");
}
```

- Para obter detalhes da API, consulte [ListObjectsV2](#) na Referência da API AWS SDK for PHP.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).


```
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 [ListObjectsV2](#) na Referência da API AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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

# Example usage:
```

```
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 [ListObjectsV2](#) na Referência da API AWS SDK for Ruby.

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
pub async fn list_objects(client: &Client, bucket: &str) -> Result<(), Error> {
  let mut response = client
    .list_objects_v2()
    .bucket(bucket.to_owned())
    .max_keys(10) // In this example, go 10 at a time.
    .into_paginator()
    .send();

  while let Some(result) = response.next().await {
    match result {
      Ok(output) => {
        for object in output.contents() {
          println!(" - {}", object.key().unwrap_or("Unknown"));
        }
      }
      Err(err) => {
        eprintln!("{err:?}")
      }
    }
  }
}
```

```
    }  
  }  
}  
  
Ok(())  
}
```

- Para obter detalhes da API, consulte [ListObjectsV2](#) na Referência da API AWS SDK para Rust.

SAP ABAP

SDK para SAP ABAP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
TRY.  
    oo_result = lo_s3->listobjectsv2(           " oo_result is returned for  
testing purposes. "  
    iv_bucket = iv_bucket_name  
    ).  
    MESSAGE 'Retrieved list of objects in S3 bucket.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- Para obter detalhes da API, consulte [ListObjectsV2](#) na Referência da API AWS SDK para SAP ABAP.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 [ListObjectsV2](#) na Referência da API do AWS SDK para Swift.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutBucketAccelerateConfiguration** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `PutBucketAccelerateConfiguration`.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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

/// <summary>
/// Amazon Simple Storage Service (Amazon S3) Transfer Acceleration is a
/// bucket-level feature that enables you to perform faster data transfers
/// to Amazon S3. This example shows how to configure Transfer
/// Acceleration.
/// </summary>
public class TransferAcceleration
{
    /// <summary>
    /// The main method initializes the client object and sets the
    /// Amazon Simple Storage Service (Amazon S3) bucket name before
    /// calling EnableAccelerationAsync.
    /// </summary>
    public static async Task Main()
    {
        var s3Client = new AmazonS3Client();
        const string bucketName = "doc-example-bucket";
```

```
        await EnableAccelerationAsync(s3Client, bucketName);
    }

    /// <summary>
    /// This method sets the configuration to enable transfer acceleration
    /// for the bucket referred to in the bucketName parameter.
    /// </summary>
    /// <param name="client">An Amazon S3 client used to enable the
    /// acceleration on an Amazon S3 bucket.</param>
    /// <param name="bucketName">The name of the Amazon S3 bucket for which
the
    /// method will be enabling acceleration.</param>
    private static async Task EnableAccelerationAsync(AmazonS3Client client,
string bucketName)
    {
        try
        {
            var putRequest = new PutBucketAccelerateConfigurationRequest
            {
                BucketName = bucketName,
                AccelerateConfiguration = new AccelerateConfiguration
                {
                    Status = BucketAccelerateStatus.Enabled,
                },
            };
            await client.PutBucketAccelerateConfigurationAsync(putRequest);

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

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

- Para obter detalhes da API, consulte [PutBucketAccelerateConfiguration](#) na Referência da API AWS SDK for .NET.

CLI

AWS CLI

Para definir a configuração de aceleração de um bucket

O exemplo `put-bucket-accelerate-configuration` a seguir habilita a configuração de aceleração do bucket especificado.

```
aws s3api put-bucket-accelerate-configuration \  
  --bucket my-bucket \  
  --accelerate-configuration Status=Enabled
```

Este comando não produz saída.

- Para obter detalhes da API, consulte [PutBucketAccelerateConfiguration](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando habilita a aceleração de transferência do bucket do S3 em questão.

```
$statusVal = New-Object Amazon.S3.BucketAccelerateStatus('Enabled')  
Write-S3BucketAccelerateConfiguration -BucketName 's3testbucket' -  
AccelerateConfiguration_Status $statusVal
```

- Para obter detalhes da API, consulte [PutBucketAccelerateConfiguration](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutBucketAcl** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `PutBucketAcl`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Gerenciar listas de controle de acesso \(ACLs\)](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <summary>
/// Creates an Amazon S3 bucket with an ACL to control access to the
/// bucket and the objects stored in it.
/// </summary>
/// <param name="client">The initialized client object used to create
/// an Amazon S3 bucket, with an ACL applied to the bucket.
/// </param>
/// <param name="region">The AWS Region where the bucket will be
created.</param>
/// <param name="newBucketName">The name of the bucket to create.</param>
/// <returns>A boolean value indicating success or failure.</returns>
public static async Task<bool> CreateBucketUseCannedACLAsync(IAmazonS3
client, S3Region region, string newBucketName)
{
    try
    {
        // Create a new Amazon S3 bucket with Canned ACL.
        var putBucketRequest = new PutBucketRequest()
        {
            BucketName = newBucketName,
```

```
        BucketRegion = region,
        CannedACL = S3CannedACL.LogDeliveryWrite,
    };

    PutBucketResponse putBucketResponse = await
client.PutBucketAsync(putBucketRequest);

    return putBucketResponse.HttpStatusCode ==
System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Amazon S3 error: {ex.Message}");
    }

    return false;
}
```

- Para obter detalhes da API, consulte [PutBucketAcl](#) na Referência da API AWS SDK for .NET.

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
bool AwsDoc::S3::PutBucketAcl(const Aws::String &bucketName,
                             const Aws::String &ownerID,
                             const Aws::String &granteePermission,
                             const Aws::String &granteeType,
                             const Aws::String &granteeID,
                             const Aws::Client::ClientConfiguration
&clientConfig,
                             const Aws::String &granteeDisplayName,
```

```
        const Aws::String &granteeEmailAddress,  
        const Aws::String &granteeURI) {  
    Aws::S3::S3Client s3_client(clientConfig);  
  
    Aws::S3::Model::Owner owner;  
    owner.SetID(ownerID);  
  
    Aws::S3::Model::Grantee grantee;  
    grantee.SetType(SetGranteeType(granteeType));  
  
    if (!granteeEmailAddress.empty()) {  
        grantee.SetEmailAddress(granteeEmailAddress);  
    }  
  
    if (!granteeID.empty()) {  
        grantee.SetID(granteeID);  
    }  
  
    if (!granteeDisplayName.empty()) {  
        grantee.SetDisplayName(granteeDisplayName);  
    }  
  
    if (!granteeURI.empty()) {  
        grantee.SetURI(granteeURI);  
    }  
  
    Aws::S3::Model::Grant grant;  
    grant.SetGrantee(grantee);  
    grant.SetPermission(SetGranteePermission(granteePermission));  
  
    Aws::Vector<Aws::S3::Model::Grant> grants;  
    grants.push_back(grant);  
  
    Aws::S3::Model::AccessControlPolicy acp;  
    acp.SetOwner(owner);  
    acp.SetGrants(grants);  
  
    Aws::S3::Model::PutBucketAclRequest request;  
    request.SetAccessControlPolicy(acp);  
    request.SetBucket(bucketName);  
  
    Aws::S3::Model::PutBucketAclOutcome outcome =  
        s3_client.PutBucketAcl(request);
```

```

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &error = outcome.GetError();

        std::cerr << "Error: PutBucketAcl: " << error.GetExceptionName()
            << " - " << error.GetMessage() << std::endl;
    }
    else {
        std::cout << "Successfully added an ACL to the bucket '" << bucketName
            << "'." << std::endl;
    }

    return outcome.IsSuccess();
}

//! Routine which converts a human-readable string to a built-in type
enumeration.
/*!
 \sa SetGranteePermission()
 \param access Human readable string.
 */

Aws::S3::Model::Permission SetGranteePermission(const Aws::String &access) {
    if (access == "FULL_CONTROL")
        return Aws::S3::Model::Permission::FULL_CONTROL;
    if (access == "WRITE")
        return Aws::S3::Model::Permission::WRITE;
    if (access == "READ")
        return Aws::S3::Model::Permission::READ;
    if (access == "WRITE_ACP")
        return Aws::S3::Model::Permission::WRITE_ACP;
    if (access == "READ_ACP")
        return Aws::S3::Model::Permission::READ_ACP;
    return Aws::S3::Model::Permission::NOT_SET;
}

//! Routine which converts a human-readable string to a built-in type
enumeration.
/*!
 \sa SetGranteeType()
 \param type Human readable string.
 */

Aws::S3::Model::Type SetGranteeType(const Aws::String &type) {
    if (type == "Amazon customer by email")

```

```
    return Aws::S3::Model::Type::AmazonCustomerByEmail;
    if (type == "Canonical user")
        return Aws::S3::Model::Type::CanonicalUser;
    if (type == "Group")
        return Aws::S3::Model::Type::Group;
    return Aws::S3::Model::Type::NOT_SET;
}
```

- Para obter detalhes da API, consulte [PutBucketAcl](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

Este exemplo concede `full control` para dois usuários da AWS (`user1@example.com` e `user2@example.com`) e permissão `read` para todos:

```
aws s3api put-bucket-acl --bucket MyBucket --grant-full-control
emailaddress=user1@example.com,emailaddress=user2@example.com --grant-read
uri=http://acs.amazonaws.com/groups/global/AllUsers
```

Consulte <http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html> para obter detalhes sobre ACLs personalizadas (os comandos da ACL `s3api`, como `put-bucket-acl`, usam a mesma notação abreviada de argumentos).

- Para obter detalhes da API, consulte [PutBucketAcl](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.AccessControlPolicy;
import software.amazon.awssdk.services.s3.model.Grant;
import software.amazon.awssdk.services.s3.model.Permission;
import software.amazon.awssdk.services.s3.model.PutBucketAclRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.Type;

import java.util.ArrayList;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SetAcl {
    public static void main(String[] args) {
        final String usage = ""

                Usage:
                <bucketName> <id>\s

                Where:
                bucketName - The Amazon S3 bucket to grant permissions on.\s
                id - The ID of the owner of this bucket (you can get this value
from the AWS Management Console).
                """;

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

        String bucketName = args[0];
        String id = args[1];
        System.out.format("Setting access \n");
        System.out.println(" in bucket: " + bucketName);
    }
}
```

```
    Region region = Region.US_EAST_1;
    S3Client s3 = S3Client.builder()
        .region(region)
        .build();

    setBucketAcl(s3, bucketName, id);
    System.out.println("Done!");
    s3.close();
}

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 AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Coloque a ACL do bucket.

```
import { PutBucketAclCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// Most Amazon S3 use cases don't require the use of access control lists (ACLs).
// We recommend that you disable ACLs, except in unusual circumstances where
// you need to control access for each object individually.
// Consider a policy instead. For more information see https://
docs.aws.amazon.com/AmazonS3/latest/userguide/bucket-policies.html.
export const main = async () => {
  // Grant a user READ access to a bucket.
  const command = new PutBucketAclCommand({
    Bucket: "test-bucket",
    AccessControlPolicy: {
      Grants: [
        {
          Grantee: {
            // The canonical ID of the user. This ID is an obfuscated form of
            // your AWS account number.
            // It's unique to Amazon S3 and can't be found elsewhere.
            // For more information, see https://docs.aws.amazon.com/AmazonS3/
latest/userguide/finding-canonical-user-id.html.
            ID: "canonical-id-1",
            Type: "CanonicalUser",
          },
          // One of FULL_CONTROL | READ | WRITE | READ_ACP | WRITE_ACP
          // https://docs.aws.amazon.com/AmazonS3/latest/API/
API_Grant.html#AmazonS3-Type-Grant-Permission
          Permission: "FULL_CONTROL",
        },
      ],
    },
  });
};
```



```
        Owner: {
            ID: "canonical-id-2",
        },
    },
});

try {
    const response = await client.send(command);
    console.log(response);
} catch (err) {
    console.error(err);
}
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).
- Para obter detalhes da API, consulte [PutBucketAcl](#) na Referência da API AWS SDK for JavaScript.

Kotlin

SDK for Kotlin

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Kotlin.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutBucketCors** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `PutBucketCors`.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <summary>
/// Add CORS configuration to the Amazon S3 bucket.
/// </summary>
/// <param name="client">The initialized Amazon S3 client object used
/// to apply the CORS configuration to an Amazon S3 bucket.</param>
/// <param name="configuration">The CORS configuration to apply.</param>
private static async Task PutCORSConfigurationAsync(AmazonS3Client
client, CORSConfiguration configuration)
{
    PutCORSConfigurationRequest request = new
PutCORSConfigurationRequest()
    {
        BucketName = BucketName,
        Configuration = configuration,
    };

    _ = await client.PutCORSConfigurationAsync(request);
}
```

- Para obter detalhes da API, consulte [PutBucketCors](#) na Referência da API AWS SDK for .NET.

CLI

AWS CLI

O seguinte exemplo habilita solicitações PUT, POST e DELETE de `www.example.com` e solicitações GET de qualquer domínio:

```
aws s3api put-bucket-cors --bucket MyBucket --cors-configuration file://cors.json

cors.json:
{
  "CORSRules": [
    {
      "AllowedOrigins": ["http://www.example.com"],
      "AllowedHeaders": ["*"],
      "AllowedMethods": ["PUT", "POST", "DELETE"],
      "MaxAgeSeconds": 3000,
      "ExposeHeaders": ["x-amz-server-side-encryption"]
    },
    {
      "AllowedOrigins": ["*"],
      "AllowedHeaders": ["Authorization"],
      "AllowedMethods": ["GET"],
      "MaxAgeSeconds": 3000
    }
  ]
}
```

- Para obter detalhes da API, consulte [PutBucketCors](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import java.util.ArrayList;
import java.util.List;
import software.amazon.awssdk.services.s3.model.GetBucketCorsRequest;
import software.amazon.awssdk.services.s3.model.GetBucketCorsResponse;
import software.amazon.awssdk.services.s3.model.DeleteBucketCorsRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.CORSRule;
import software.amazon.awssdk.services.s3.model.CORSConfiguration;
import software.amazon.awssdk.services.s3.model.PutBucketCorsRequest;

/**
 * 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 S3Cors {
    public static void main(String[] args) {
        final String usage = ""

                Usage:
                <bucketName> <accountId>\s

                Where:
                bucketName - The Amazon S3 bucket to upload an object into.
                accountId - The id of the account that owns the Amazon S3
                bucket.
```

```
        """;

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

    String bucketName = args[0];
    String accountId = args[1];
    Region region = Region.US_EAST_1;
    S3Client s3 = S3Client.builder()
        .region(region)
        .build();

    setCorsInformation(s3, bucketName, accountId);
    getBucketCorsInformation(s3, bucketName, accountId);
    deleteBucketCorsInformation(s3, bucketName, accountId);
    s3.close();
}

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 AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Adicione uma regra de CORS.

```
import { PutBucketCorsCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// By default, Amazon S3 doesn't allow cross-origin requests. Use this command
// to explicitly allow cross-origin requests.
export const main = async () => {
    const command = new PutBucketCorsCommand({
        Bucket: "test-bucket",
        CORSConfiguration: {
            CORSRules: [
                {
                    // Allow all headers to be sent to this bucket.
                    AllowedHeaders: ["*"],
                    // Allow only GET and PUT methods to be sent to this bucket.
                }
            ]
        }
    });
};
```

```
    AllowedMethods: ["GET", "PUT"],
    // Allow only requests from the specified origin.
    AllowedOrigins: ["https://www.example.com"],
    // Allow the entity tag (ETag) header to be returned in the response.
The ETag header
    // The entity tag represents a specific version of the object. The ETag
reflects
    // changes only to the contents of an object, not its metadata.
    ExposeHeaders: ["ETag"],
    // How long the requesting browser should cache the preflight response.
After
    // this time, the preflight request will have to be made again.
    MaxAgeSeconds: 3600,
  },
],
},
});

try {
  const response = await client.send(command);
  console.log(response);
} catch (err) {
  console.error(err);
}
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).
- Para obter detalhes da API, consulte [PutBucketCors](#) na Referência da API AWS SDK for JavaScript.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK for Ruby.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutBucketEncryption** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `PutBucketEncryption`.

CLI

AWS CLI

Para configurar a criptografia do lado do servidor para um bucket

O exemplo `put-bucket-encryption` a seguir define a criptografia AES256 como padrão para o bucket especificado.

```
aws s3api put-bucket-encryption \
  --bucket my-bucket \
  --server-side-encryption-configuration '{"Rules":
  [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}}]}'
```

Este comando não produz saída.

- Para obter detalhes da API, consulte [PutBucketEncryption](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando habilita a criptografia padrão do lado do servidor AES256 com chaves gerenciadas do Amazon S3 (SSE-S3) no bucket em questão.

```
$Encryptionconfig = @{ServerSideEncryptionByDefault =  
    @{ServerSideEncryptionAlgorithm = "AES256"}}  
Set-S3BucketEncryption -BucketName 's3testbucket' -  
ServerSideEncryptionConfiguration_ServerSideEncryptionRule $Encryptionconfig
```

- Para obter detalhes da API, consulte [PutBucketEncryption](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutBucketLifecycleConfiguration** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `PutBucketLifecycleConfiguration`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Trabalhar com objetos versionados](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <summary>
/// Adds lifecycle configuration information to the S3 bucket named in
/// the bucketName parameter.
/// </summary>
/// <param name="client">The S3 client used to call the
/// PutLifecycleConfigurationAsync method.</param>
/// <param name="bucketName">A string representing the S3 bucket to
/// which configuration information will be added.</param>
/// <param name="configuration">A LifecycleConfiguration object that
/// will be applied to the S3 bucket.</param>
public static async Task AddExampleLifecycleConfigAsync(IAmazonS3 client,
string bucketName, LifecycleConfiguration configuration)
{
    var request = new PutLifecycleConfigurationRequest()
    {
        BucketName = bucketName,
        Configuration = configuration,
    };
    var response = await client.PutLifecycleConfigurationAsync(request);
}
```

- Para obter informações da API, consulte [PutBucketLifecycleConfiguration](#) em Referência da API AWS SDK for .NET .

CLI

AWS CLI

O seguinte comando aplica a configuração de ciclo de vida ao bucket my-bucket:

```
aws s3api put-bucket-lifecycle-configuration --bucket my-bucket --lifecycle-configuration file://lifecycle.json
```

O arquivo `lifecycle.json` é um documento JSON na pasta atual que especifica duas regras:

```
{
  "Rules": [
```

```
{
  "ID": "Move rotated logs to Glacier",
  "Prefix": "rotated/",
  "Status": "Enabled",
  "Transitions": [
    {
      "Date": "2015-11-10T00:00:00.000Z",
      "StorageClass": "GLACIER"
    }
  ],
},
{
  "Status": "Enabled",
  "Prefix": "",
  "NoncurrentVersionTransitions": [
    {
      "NoncurrentDays": 2,
      "StorageClass": "GLACIER"
    }
  ],
  "ID": "Move old versions to Glacier"
}
]
```

A primeira regra move arquivos com o prefixo `rotated` para o Glacier na data especificada. A segunda regra move versões antigas de objetos para o Glacier quando elas não estão mais atualizadas. Para obter mais informações sobre formatos de data/hora aceitáveis, consulte [Especificar valores de parâmetro](#) Guia do usuário da AWS CLI.

- Para obter detalhes da API, consulte [PutBucketLifecycleConfiguration](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).


```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.LifecycleRuleFilter;
import software.amazon.awssdk.services.s3.model.Transition;
import
    software.amazon.awssdk.services.s3.model.GetBucketLifecycleConfigurationRequest;
import
    software.amazon.awssdk.services.s3.model.GetBucketLifecycleConfigurationResponse;
import software.amazon.awssdk.services.s3.model.DeleteBucketLifecycleRequest;
import software.amazon.awssdk.services.s3.model.TransitionStorageClass;
import software.amazon.awssdk.services.s3.model.LifecycleRule;
import software.amazon.awssdk.services.s3.model.ExpirationStatus;
import software.amazon.awssdk.services.s3.model.BucketLifecycleConfiguration;
import
    software.amazon.awssdk.services.s3.model.PutBucketLifecycleConfigurationRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.util.ArrayList;
import java.util.List;

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

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

            Usage:
                <bucketName> <accountId>\s

            Where:
                bucketName - The Amazon Simple Storage Service
                (Amazon S3) bucket to upload an object into.
                accountId - The id of the account that owns the
                Amazon S3 bucket.

            """;
```

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

        String bucketName = args[0];
        String accountId = args[1];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        setLifecycleConfig(s3, bucketName, accountId);
        getLifecycleConfig(s3, bucketName, accountId);
        deleteLifecycleConfig(s3, bucketName, accountId);
        System.out.println("You have successfully created, updated, and
deleted a Lifecycle configuration");
        s3.close();
    }

    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 informações da API, consulte [PutBucketLifecycleConfiguration](#) em Referência da API AWS SDK for Java 2.x .

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 informações 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](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutBucketLogging** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o PutBucketLogging.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
using System;
using System.IO;
```

```
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;
using Microsoft.Extensions.Configuration;

/// <summary>
/// This example shows how to enable logging on an Amazon Simple Storage
/// Service (Amazon S3) bucket. You need to have two Amazon S3 buckets for
/// this example. The first is the bucket for which you wish to enable
/// logging, and the second is the location where you want to store the
/// logs.
/// </summary>
public class ServerAccessLogging
{
    private static IConfiguration _configuration = null!;

    public static async Task Main()
    {
        LoadConfig();

        string bucketName = _configuration["BucketName"];
        string logBucketName = _configuration["LogBucketName"];
        string logObjectKeyPrefix = _configuration["LogObjectKeyPrefix"];
        string accountId = _configuration["AccountId"];

        // If the AWS Region defined for your default user is different
        // from the Region where your Amazon S3 bucket is located,
        // pass the Region name to the Amazon S3 client object's constructor.
        // For example: RegionEndpoint.USWest2 or RegionEndpoint.USEast2.
        IAmazonS3 client = new AmazonS3Client();

        try
        {
            // Update bucket policy for target bucket to allow delivery of
logs to it.
            await SetBucketPolicyToAllowLogDelivery(
                client,
                bucketName,
                logBucketName,
                logObjectKeyPrefix,
                accountId);

            // Enable logging on the source bucket.
            await EnableLoggingAsync(
```



```

        client,
        bucketName,
        logBucketName,
        logObjectKeyPrefix);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine($"Error: {e.Message}");
    }
}

/// <summary>
/// This method grants appropriate permissions for logging to the
/// Amazon S3 bucket where the logs will be stored.
/// </summary>
/// <param name="client">The initialized Amazon S3 client which will be
used
/// to apply the bucket policy.</param>
/// <param name="sourceBucketName">The name of the source bucket.</param>
/// <param name="logBucketName">The name of the bucket where logging
/// information will be stored.</param>
/// <param name="logPrefix">The logging prefix where the logs should be
delivered.</param>
/// <param name="accountId">The account id of the account where the
source bucket exists.</param>
/// <returns>Async task.</returns>
public static async Task SetBucketPolicyToAllowLogDelivery(
    IAmazonS3 client,
    string sourceBucketName,
    string logBucketName,
    string logPrefix,
    string accountId)
{
    var resourceArn = @$"arn:aws:s3:::" + logBucketName + "/" +
logPrefix + @$""";

    var newPolicy = @"{
        ""Statement"": [{
            ""Sid"": ""S3ServerAccessLogsPolicy"",
            ""Effect"": ""Allow"",
            ""Principal"": { ""Service"":
""logging.s3.amazonaws.com"" },
            ""Action"": [""s3:PutObject""],
            ""Resource"": [" + resourceArn + @"],

```

```

        ""Condition"": {
            ""ArnLike"": { ""aws:SourceArn"":
""arn:aws:s3:::" + sourceBucketName + @"""" },
            ""StringEquals"": { ""aws:SourceAccount"": """" +
accountId + @"""" }
        }
    }
}];
    Console.WriteLine($"The policy to apply to bucket {logBucketName} to
enable logging:");
    Console.WriteLine(newPolicy);

    PutBucketPolicyRequest putRequest = new PutBucketPolicyRequest
    {
        BucketName = logBucketName,
        Policy = newPolicy,
    };
    await client.PutBucketPolicyAsync(putRequest);
    Console.WriteLine("Policy applied.");
}

/// <summary>
/// This method enables logging for an Amazon S3 bucket. Logs will be
stored
/// in the bucket you selected for logging. Selected prefix
/// will be prepended to each log object.
/// </summary>
/// <param name="client">The initialized Amazon S3 client which will be
used
/// to configure and apply logging to the selected Amazon S3 bucket.</
param>
/// <param name="bucketName">The name of the Amazon S3 bucket for which
you
/// wish to enable logging.</param>
/// <param name="logBucketName">The name of the Amazon S3 bucket where
logging
/// information will be stored.</param>
/// <param name="logObjectKeyPrefix">The prefix to prepend to each
/// object key.</param>
/// <returns>Async task.</returns>
public static async Task EnableLoggingAsync(
    IAmazonS3 client,
    string bucketName,
    string logBucketName,

```

```
        string logObjectKeyPrefix)
    {
        Console.WriteLine($"Enabling logging for bucket {bucketName}.");
        var loggingConfig = new S3BucketLoggingConfig
        {
            TargetBucketName = logBucketName,
            TargetPrefix = logObjectKeyPrefix,
        };

        var putBucketLoggingRequest = new PutBucketLoggingRequest
        {
            BucketName = bucketName,
            LoggingConfig = loggingConfig,
        };
        await client.PutBucketLoggingAsync(putBucketLoggingRequest);
        Console.WriteLine($"Logging enabled.");
    }

    /// <summary>
    /// Loads configuration from settings files.
    /// </summary>
    public static void LoadConfig()
    {
        _configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load settings from .json file.
            .AddJsonFile("settings.local.json", true) // Optionally, load
local settings.
            .Build();
    }
}
```

- Para obter detalhes da API, consulte [PutBucketLogging](#) na Referência da API AWS SDK for .NET.

CLI

AWS CLI

Exemplo 1: definir o registro em log da política de bucket

O exemplo de `put-bucket-logging` a seguir define a política de registro em log para `MyBucket`. Primeiro, conceda permissão à entidade principal do serviço de registro em log na política de bucket usando o comando `put-bucket-policy`.

```
aws s3api put-bucket-policy \  
  --bucket MyBucket \  
  --policy file://policy.json
```

Conteúdo de `policy.json`:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "S3ServerAccessLogsPolicy",  
      "Effect": "Allow",  
      "Principal": {"Service": "logging.s3.amazonaws.com"},  
      "Action": "s3:PutObject",  
      "Resource": "arn:aws:s3:::MyBucket/Logs/*",  
      "Condition": {  
        "ArnLike": {"aws:SourceARN": "arn:aws:s3:::SOURCE-BUCKET-NAME"},  
        "StringEquals": {"aws:SourceAccount": "SOURCE-AWS-ACCOUNT-ID"}  
      }  
    }  
  ]  
}
```

Para aplicar a política de registro em log, use `put-bucket-logging`.

```
aws s3api put-bucket-logging \  
  --bucket MyBucket \  
  --bucket-logging-status file://logging.json
```

Conteúdo de `logging.json`:

```
{  
  "LoggingEnabled": {  
    "TargetBucket": "MyBucket",  
    "TargetPrefix": "Logs/"  
  }  
}
```

O comando `put-bucket-policy` é necessário para conceder as permissões `s3:PutObject` à entidade principal do serviço de registro em log.

Consulte mais informações em [Registrar em log as solicitações com registro em log de acesso ao servidor](#) no Guia do usuário do Amazon S3.

Exemplo 2: definir uma política de bucket para registrar em log o acesso a um único usuário

O exemplo de `put-bucket-logging` a seguir define a política de registro em log para `MyBucket`. O usuário `bob@exemplo.com` da AWS terá controle total sobre os arquivos de log e ninguém mais terá acesso. Primeiro, conceda permissão ao S3 com `put-bucket-acl`.

```
aws s3api put-bucket-acl \  
  --bucket MyBucket \  
  --grant-write URI=http://acs.amazonaws.com/groups/s3/LogDelivery \  
  --grant-read-acp URI=http://acs.amazonaws.com/groups/s3/LogDelivery
```

Depois, aplique a política de registro em log usando `put-bucket-logging`.

```
aws s3api put-bucket-logging \  
  --bucket MyBucket \  
  --bucket-logging-status file://logging.json
```

Conteúdo de `logging.json`:

```
{  
  "LoggingEnabled": {  
    "TargetBucket": "MyBucket",  
    "TargetPrefix": "MyBucketLogs/",  
    "TargetGrants": [  
      {  
        "Grantee": {  
          "Type": "AmazonCustomerByEmail",  
          "EmailAddress": "bob@example.com"  
        },  
        "Permission": "FULL_CONTROL"  
      }  
    ]  
  }  
}
```

O comando `put-bucket-acl` é necessário para conceder as permissões necessárias ao sistema de entrega de log do S3 (permissões de gravação e `read-acp`).

Consulte mais informações em [Habilitar o log de acesso ao servidor do Amazon S3](#) no Guia do desenvolvedor do Amazon S3.

- Consulte detalhes da API em [PutBucketLogging](#) na Referência de comandos da AWS CLI.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutBucketNotification** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `PutBucketNotification`.

CLI

AWS CLI

O exemplo aplica uma configuração de notificação ao bucket `my-bucket`:

```
aws s3api put-bucket-notification --bucket my-bucket --notification-configuration
file://notification.json
```

O arquivo `notification.json` é um documento JSON na pasta atual que especifica um tópico do SNS e o tipo de evento a ser monitorado:

```
{
  "TopicConfiguration": {
    "Event": "s3:ObjectCreated:*",
    "Topic": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic"
  }
}
```

O tópico do SNS deve ter uma política do IAM anexada que permita que o Amazon S3 publique nele:

```
{
```

```
"Version": "2008-10-17",
"Id": "example-ID",
"Statement": [
  {
    "Sid": "example-statement-ID",
    "Effect": "Allow",
    "Principal": {
      "Service": "s3.amazonaws.com"
    },
    "Action": [
      "SNS:Publish"
    ],
    "Resource": "arn:aws:sns:us-west-2:123456789012:my-bucket",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:s3:*:*:my-bucket"
      }
    }
  }
]
```

- Consulte detalhes da API em [PutBucketNotification](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este exemplo define a configuração do tópico do SNS para o evento ObjectRemovedDelete do S3 e habilita a notificação para o bucket do S3 em questão.

```
$topic = [Amazon.S3.Model.TopicConfiguration] @{
  Id = "delete-event"
  Topic = "arn:aws:sns:eu-west-1:123456789012:topic-1"
  Event = [Amazon.S3.EventType]::ObjectRemovedDelete
}

Write-S3BucketNotification -BucketName kt-tools -TopicConfiguration $topic
```

Exemplo 2: este exemplo habilita notificações de ObjectCreatedAll para o bucket em questão, enviando-as à função do Lambda.

```

$lambdaConfig = [Amazon.S3.Model.LambdaFunctionConfiguration] @{
    Events = "s3:ObjectCreated:*"
    FunctionArn = "arn:aws:lambda:eu-west-1:123456789012:function:rdplock"
    Id = "ObjectCreated-Lambda"
    Filter = @{
        S3KeyFilter = @{
            FilterRules = @(
                @{Name="Prefix";Value="dada"}
                @{Name="Suffix";Value=".pem"}
            )
        }
    }
}

Write-S3BucketNotification -BucketName ssm-editor -LambdaFunctionConfiguration
$lambdaConfig

```

Exemplo 3: este exemplo cria duas configurações diferentes do Lambda com base em diferentes sufixos de chave e configura ambas em um único comando.

```

#Lambda Config 1

$firstLambdaConfig = [Amazon.S3.Model.LambdaFunctionConfiguration] @{
    Events = "s3:ObjectCreated:*"
    FunctionArn = "arn:aws:lambda:eu-west-1:123456789012:function:verifynet"
    Id = "ObjectCreated-dada-ps1"
    Filter = @{
        S3KeyFilter = @{
            FilterRules = @(
                @{Name="Prefix";Value="dada"}
                @{Name="Suffix";Value=".ps1"}
            )
        }
    }
}

#Lambda Config 2

$secondLambdaConfig = [Amazon.S3.Model.LambdaFunctionConfiguration] @{
    Events = [Amazon.S3.EventType]::ObjectCreatedAll
    FunctionArn = "arn:aws:lambda:eu-west-1:123456789012:function:verifyssm"
    Id = "ObjectCreated-dada-json"
    Filter = @{

```



```
S3KeyFilter = @{
    FilterRules = @(
        @{Name="Prefix";Value="dada"}
        @{Name="Suffix";Value=".json"}
    )
}
}
```

```
Write-S3BucketNotification -BucketName ssm-editor -LambdaFunctionConfiguration
    $firstLambdaConfig,$secondLambdaConfig
```

- Consulte detalhes da API em [PutBucketNotification](#) em AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutBucketNotificationConfiguration** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `PutBucketNotificationConfiguration`.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;
```

```
/// <summary>
/// This example shows how to enable notifications for an Amazon Simple
/// Storage Service (Amazon S3) bucket.
/// </summary>
public class EnableNotifications
{
    public static async Task Main()
    {
        const string bucketName = "doc-example-bucket1";
        const string snsTopic = "arn:aws:sns:us-east-2:0123456789ab:bucket-
notify";
        const string sqsQueue = "arn:aws:sqs:us-
east-2:0123456789ab:Example_Queue";

        IAmazonS3 client = new AmazonS3Client(Amazon.RegionEndpoint.USEast2);
        await EnableNotificationAsync(client, bucketName, snsTopic,
sqsQueue);
    }

    /// <summary>
    /// This method makes the call to the PutBucketNotificationAsync method.
    /// </summary>
    /// <param name="client">An initialized Amazon S3 client used to call
    /// the PutBucketNotificationAsync method.</param>
    /// <param name="bucketName">The name of the bucket for which
    /// notifications will be turned on.</param>
    /// <param name="snsTopic">The ARN for the Amazon Simple Notification
    /// Service (Amazon SNS) topic associated with the S3 bucket.</param>
    /// <param name="sqsQueue">The ARN of the Amazon Simple Queue Service
    /// (Amazon SQS) queue to which notifications will be pushed.</param>
    public static async Task EnableNotificationAsync(
        IAmazonS3 client,
        string bucketName,
        string snsTopic,
        string sqsQueue)
    {
        try
        {
            // The bucket for which we are setting up notifications.
            var request = new PutBucketNotificationRequest()
            {
                BucketName = bucketName,
            };
        }
    }
}
```

```
// Defines the topic to use when sending a notification.
var topicConfig = new TopicConfiguration()
{
    Events = new List<EventType> { EventType.ObjectCreatedCopy },
    Topic = snsTopic,
};
request.TopicConfigurations = new List<TopicConfiguration>
{
    topicConfig,
};
request.QueueConfigurations = new List<QueueConfiguration>
{
    new QueueConfiguration()
    {
        Events = new List<EventType>
{ EventType.ObjectCreatedPut },
        Queue = sqsQueue,
    },
};

// Now apply the notification settings to the bucket.
PutBucketNotificationResponse response = await
client.PutBucketNotificationAsync(request);
}
catch (AmazonS3Exception ex)
{
    Console.WriteLine($"Error: {ex.Message}");
}
}
```

- Para obter detalhes da API, consulte [PutBucketNotificationConfiguration](#) na Referência da API AWS SDK for .NET.

CLI

AWS CLI

Como habilitar as notificações especificadas em um bucket

O exemplo de `put-bucket-notification-configuration` a seguir aplica uma configuração de notificação a um bucket chamado `my-bucket`. O arquivo `notification.json` é um documento JSON na pasta atual que especifica um tópico do SNS e o tipo de evento a ser monitorado.

```
aws s3api put-bucket-notification-configuration \  
  --bucket my-bucket \  
  --notification-configuration file://notification.json
```

Conteúdo de `notification.json`:

```
{  
  "TopicConfigurations": [  
    {  
      "TopicArn": "arn:aws:sns:us-west-2:123456789012:s3-notification-  
topic",  
      "Events": [  
        "s3:ObjectCreated:*"  
      ]  
    }  
  ]  
}
```

O tópico do SNS deve ter uma política do IAM anexada que permita que o Amazon S3 publique nele.

```
{  
  "Version": "2008-10-17",  
  "Id": "example-ID",  
  "Statement": [  
    {  
      "Sid": "example-statement-ID",  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "s3.amazonaws.com"  
      },  
      "Action": [  
        "SNS:Publish"  
      ],  
      "Resource": "arn:aws:sns:us-west-2:123456789012::s3-notification-  
topic",  
      "Condition": {
```

```
        "ArnLike": {
            "aws:SourceArn": "arn:aws:s3:*:*:my-bucket"
        }
    }
}
]
```

- Para obter detalhes da API, consulte [PutBucketNotificationConfiguration](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.Event;
import software.amazon.awssdk.services.s3.model.NotificationConfiguration;
import
    software.amazon.awssdk.services.s3.model.PutBucketNotificationConfigurationRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.TopicConfiguration;
import java.util.ArrayList;
import java.util.List;

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

                Usage:
                <bucketName>\s

                Where:
                bucketName - The Amazon S3 bucket.\s
```

```
        topicArn - The Simple Notification Service topic ARN.\s
        id - An id value used for the topic configuration. This value
is displayed in the AWS Management Console.\s
        """;

    if (args.length != 3) {
        System.out.println(usage);
        System.exit(1);
    }

    String bucketName = args[0];
    String topicArn = args[1];
    String id = args[2];
    Region region = Region.US_EAST_1;
    S3Client s3Client = S3Client.builder()
        .region(region)
        .build();

    setBucketNotification(s3Client, bucketName, topicArn, id);
    s3Client.close();
}

public static void setBucketNotification(S3Client s3Client, String
bucketName, String topicArn, String id) {
    try {
        List<Event> events = new ArrayList<>();
        events.add(Event.S3_OBJECT_CREATED_PUT);

        TopicConfiguration config = TopicConfiguration.builder()
            .topicArn(topicArn)
            .events(events)
            .id(id)
            .build();

        List<TopicConfiguration> topics = new ArrayList<>();
        topics.add(config);

        NotificationConfiguration configuration =
NotificationConfiguration.builder()
            .topicConfigurations(topics)
            .build();

        PutBucketNotificationConfigurationRequest configurationRequest =
PutBucketNotificationConfigurationRequest
```

```
        .builder()
        .bucket(bucketName)
        .notificationConfiguration(configuration)
        .skipDestinationValidation(true)
        .build();

        // Set the bucket notification configuration.
        s3Client.putBucketNotificationConfiguration(configurationRequest);
        System.out.println("Added bucket " + bucketName + " with EventBridge
events enabled.");

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- Para obter detalhes da API, consulte [PutBucketNotificationConfiguration](#) na Referência da API do AWS SDK for Java 2.x.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutBucketPolicy** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o PutBucketPolicy.

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```

bool AwsDoc::S3::PutBucketPolicy(const Aws::String &bucketName,
                                const Aws::String &policyBody,
                                const Aws::Client::ClientConfiguration
&clientConfig) {
    Aws::S3::S3Client s3_client(clientConfig);

    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::cerr << "Error: PutBucketPolicy: "
                  << outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Set the following policy body for the bucket '" <<
                  bucketName << "':" << std::endl << std::endl;
        std::cout << policyBody << std::endl;
    }

    return outcome.IsSuccess();
}

//! Build a policy JSON string.
/*!
    \sa GetPolicyString()
    \param userArn Aws user Amazon Resource Name (ARN).
        For more information, see https://docs.aws.amazon.com/IAM/latest/UserGuide/reference\_identifiers.html#identifiers-arns.
    \param bucketName Name of a bucket.
*/

Aws::String GetPolicyString(const Aws::String &userArn,
                            const Aws::String &bucketName) {
    return

```



```

    "{\n"
    "  \"Version\": \"2012-10-17\", \n"
    "  \"Statement\": [\n"
    "    {\n"
    "      \"Sid\": \"1\", \n"
    "      \"Effect\": \"Allow\", \n"
    "      \"Principal\": {\n"
    "        \"AWS\": \"\"
+ userArn +
    \"\n\"      }, \n"
    "      \"Action\": [ \"s3:GetObject\" ], \n"
    "      \"Resource\": [ \"arn:aws:s3::\"
+ bucketName +
    \"/*\" ] \n"
    "    } \n"
    "  ] \n"
    "};
}

```

- Para obter detalhes da API, consulte [PutBucketPolicy](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

Este exemplo permite que todos os usuários recuperem qualquer objeto em MyBucket, exceto os objetos em MySecretFolder. Ele também concede permissões put e delete ao usuário raiz da conta da AWS 1234-5678-9012:

```
aws s3api put-bucket-policy --bucket MyBucket --policy file:///policy.json
```

policy.json:

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::MyBucket/*"
    },

```

```
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::MyBucket/MySecretFolder/*"
},
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::123456789012:root"
  },
  "Action": [
    "s3:DeleteObject",
    "s3:PutObject"
  ],
  "Resource": "arn:aws:s3:::MyBucket/*"
}
]
```

- Para obter detalhes da API, consulte [PutBucketPolicy](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutBucketPolicyRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.regions.Region;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
```

```
import java.util.List;
import com.fasterxml.jackson.core.JsonParser;
import com.fasterxml.jackson.databind.ObjectMapper;

/**
 * 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 SetBucketPolicy {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <bucketName> <polFile>

            Where:
                bucketName - The Amazon S3 bucket to set the policy on.
                polFile - A JSON file containing the policy (see the Amazon
S3 Readme for an example).\s
            """;

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

        String bucketName = args[0];
        String polFile = args[1];
        String policyText = getBucketPolicyFromFile(polFile);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        setPolicy(s3, bucketName, policyText);
        s3.close();
    }
}
```

```
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 informações da API, consulte [PutBucketPolicy](#) na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Adicione a política.

```
import { PutBucketPolicyCommand, S3Client } from "@aws-sdk/client-s3";  
  
const client = new S3Client({});  
  
export const main = async () => {  
    const command = new PutBucketPolicyCommand({  
        Policy: JSON.stringify({  
            Version: "2012-10-17",  
            Statement: [  
                {  
                    Sid: "AllowGetObject",  
                    // Allow this particular user to call GetObject on any object in this  
                    bucket.  
                    Effect: "Allow",  
                    Principal: {  
                        AWS: "arn:aws:iam::ACCOUNT-ID:user/USERNAME",
```

```
    },
    Action: "s3:GetObject",
    Resource: "arn:aws:s3:::BUCKET-NAME/*",
  },
],
}),
// Apply the preceding policy to this bucket.
Bucket: "BUCKET-NAME",
});

try {
  const response = await client.send(command);
  console.log(response);
} catch (err) {
  console.error(err);
}
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).
- Para obter detalhes da API, consulte [PutBucketPolicy](#) na Referência da API AWS SDK for JavaScript.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 informações da API, consulte [PutBucketPolicy](#) na Referência da API AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
# 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 informações 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](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutBucketReplication** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o PutBucketReplication.

CLI

AWS CLI

Como configurar a replicação para um bucket do S3

O exemplo `put-bucket-replication` a seguir aplica uma configuração de replicação ao bucket do S3 especificado.

```
aws s3api put-bucket-replication \
```



```
--bucket AWSDOC-EXAMPLE-BUCKET1 \  
--replication-configuration file://replication.json
```

Conteúdo de replication.json:

```
{  
  "Role": "arn:aws:iam::123456789012:role/s3-replication-role",  
  "Rules": [  
    {  
      "Status": "Enabled",  
      "Priority": 1,  
      "DeleteMarkerReplication": { "Status": "Disabled" },  
      "Filter" : { "Prefix": ""},  
      "Destination": {  
        "Bucket": "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET2"  
      }  
    }  
  ]  
}
```

O bucket de destino deve ter o versionamento habilitado. A função especificada deve ter permissão para gravar no bucket de destino e ter uma relação de confiança que permita que o Amazon S3 assuma a função.

Exemplo de política de permissão de função:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:GetReplicationConfiguration",  
        "s3:ListBucket"  
      ],  
      "Resource": [  
        "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET1"  
      ]  
    },  
    {  
      "Effect": "Allow",  
      "Action": [  

```

```
        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
    ],
    "Resource": [
        "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET1/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete",
        "s3:ReplicateTags"
    ],
    "Resource": "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET2/*"
}
]
```

Exemplo de política de relação de confiança:

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

Este comando não produz saída.

Para obter mais informações, consulte [This is the topic title](#) no Guia do usuário do Amazon Simple Storage Service.

- Para obter detalhes da API, consulte [PutBucketReplication](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este exemplo define uma configuração de replicação com uma única regra que permite a replicação para o bucket “exampletargetbucket” de qualquer novo objeto criado com o prefixo de nome de chave “TaxDocs” no bucket “examplebucket”.

```
$rule1 = New-Object Amazon.S3.Model.ReplicationRule
$rule1.ID = "Rule-1"
$rule1.Status = "Enabled"
$rule1.Prefix = "TaxDocs"
$rule1.Destination = @{ BucketArn = "arn:aws:s3:::exampletargetbucket" }

$params = @{
    BucketName = "examplebucket"
    Configuration_Role = "arn:aws:iam::35667example:role/
CrossRegionReplicationRoleForS3"
    Configuration_Rule = $rule1
}

Write-S3BucketReplication @params
```

Exemplo 2: este exemplo define uma configuração de replicação com várias regras que permitem a replicação para o bucket “exampletargetbucket” de qualquer novo objeto criado com o prefixo de nome de chave “TaxDocs” ou “OtherDocs”. Os prefixos de chave não devem se sobrepor.

```
$rule1 = New-Object Amazon.S3.Model.ReplicationRule
$rule1.ID = "Rule-1"
$rule1.Status = "Enabled"
$rule1.Prefix = "TaxDocs"
$rule1.Destination = @{ BucketArn = "arn:aws:s3:::exampletargetbucket" }

$rule2 = New-Object Amazon.S3.Model.ReplicationRule
$rule2.ID = "Rule-2"
$rule2.Status = "Enabled"
$rule2.Prefix = "OtherDocs"
$rule2.Destination = @{ BucketArn = "arn:aws:s3:::exampletargetbucket" }

$params = @{
    BucketName = "examplebucket"
```

```

    Configuration_Role = "arn:aws:iam::35667example:role/
CrossRegionReplicationRoleForS3"
    Configuration_Rule = $rule1,$rule2
}

Write-S3BucketReplication @params

```

Exemplo 3: este exemplo atualiza a configuração de replicação no bucket especificado para desabilitar a regra que controla a replicação de objetos com o prefixo de nome de chave “TaxDocs” para o bucket “exampletargetbucket.

```

$rule1 = New-Object Amazon.S3.Model.ReplicationRule
$rule1.ID = "Rule-1"
$rule1.Status = "Disabled"
$rule1.Prefix = "TaxDocs"
$rule1.Destination = @{ BucketArn = "arn:aws:s3:::exampletargetbucket" }

$params = @{
    BucketName = "examplebucket"
    Configuration_Role = "arn:aws:iam::35667example:role/
CrossRegionReplicationRoleForS3"
    Configuration_Rule = $rule1
}

Write-S3BucketReplication @params

```

- Para obter detalhes, consulte o [PutBucketReplication](#), na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutBucketRequestPayment** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o PutBucketRequestPayment.

CLI

AWS CLI

Exemplo 1: para habilitar a configuração “requester pays” para um bucket

O exemplo `put-bucket-request-payment` a seguir habilita `requester pays` do bucket especificado.

```
aws s3api put-bucket-request-payment \  
  --bucket my-bucket \  
  --request-payment-configuration '{"Payer":"Requester"}'
```

Este comando não produz saída.

Exemplo 2: para desabilitar a configuração “requester pays” para um bucket

O exemplo `put-bucket-request-payment` a seguir desabilita `requester pays` do bucket especificado.

```
aws s3api put-bucket-request-payment \  
  --bucket my-bucket \  
  --request-payment-configuration '{"Payer":"BucketOwner"}'
```

Este comando não produz saída.

- Para obter detalhes da API, consulte [PutBucketRequestPayment](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: atualiza a configuração de pagamento de solicitação do bucket denominado “mybucket” para que a pessoa que está solicitando downloads pelo bucket seja cobrada pelo download. Por padrão, o proprietário do bucket paga pelos downloads. Para definir o pagamento de solicitação de volta ao padrão, use “BucketOwner” para o parâmetro `RequestPaymentConfiguration_Payer`.

```
Write-S3BucketRequestPayment -BucketName mybucket -  
RequestPaymentConfiguration_Payer Requester
```

- Para obter detalhes da API, consulte [PutBucketRequestPayment](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutBucketTagging** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o PutBucketTagging.

CLI

AWS CLI

O seguinte comando aplica a configuração de marcação a um bucket denominado my-bucket:

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging file://tagging.json
```

O arquivo `tagging.json` é um documento JSON na pasta atual que especifica duas tags:

```
{
  "TagSet": [
    {
      "Key": "organization",
      "Value": "marketing"
    }
  ]
}
```

Ou aplique uma configuração de marcação a my-bucket diretamente da linha de comando:

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging
'TagSet=[{Key=organization,Value=marketing}]'
```

- Para obter detalhes da API, consulte [PutBucketTagging](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: este comando aplica duas tags a um bucket denominado **cloudtrail-test-2018** (uma tag com uma chave de Stage e um valor de Test e uma tag com uma chave de Environment e um valor de Alpha). Para verificar se as tags foram adicionadas ao bucket, execute **Get-S3BucketTagging -BucketName bucket_name**. Os resultados devem mostrar as tags que você aplicou ao bucket no primeiro comando. Observe que **Write-S3BucketTagging** substitui todo o conjunto de tags existente em um bucket. Para adicionar ou excluir tags individuais, execute os cmdlets da API de grupos de recursos e marcação: **Add-RGResourceTag** e **Remove-RGResourceTag**. Também é possível usar o editor de tags no Console de Gerenciamento da AWS para gerenciar as tags de bucket do S3.

```
Write-S3BucketTagging -BucketName cloudtrail-test-2018 -TagSet @( @{ Key="Stage"; Value="Test" }, @{ Key="Environment"; Value="Alpha" } )
```

Exemplo 2: este comando envia um bucket denominado **cloudtrail-test-2018** para o cmdlet **Write-S3BucketTagging**. Ele aplica as tags Stage:Production e Department:Finance ao bucket. Observe que **Write-S3BucketTagging** substitui todo o conjunto de tags existente em um bucket.

```
Get-S3Bucket -BucketName cloudtrail-test-2018 | Write-S3BucketTagging -TagSet @( @{ Key="Stage"; Value="Production" }, @{ Key="Department"; Value="Finance" } )
```

- Para obter detalhes da API, consulte [PutBucketTagging](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutBucketVersioning** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o **PutBucketVersioning**.

CLI

AWS CLI

O comando a seguir habilita o versionamento em um bucket denominado my-bucket.

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-configuration
Status=Enabled
```

O comando a seguir habilita o versionamento e usa um código mfa.

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-configuration
Status=Enabled --mfa "SERIAL 123456"
```

- Consulte detalhes da API em [PutBucketVersioning](#) na Referência de comandos da AWS CLI.

PowerShell

Tools for PowerShell

Exemplo 1: o comando habilita o versionamento do bucket do S3 em questão.

```
Write-S3BucketVersioning -BucketName 's3testbucket' -VersioningConfig_Status
Enabled
```

- Para obter detalhes da API, consulte [PutBucketVersioning](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutBucketWebsite** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o PutBucketWebsite.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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

- Para obter detalhes da API, consulte [PutBucketWebsite](#) na Referência da API AWS SDK for .NET.

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
bool AwsDoc::S3::PutWebsiteConfig(const Aws::String &bucketName,
```

```

        const Aws::String &indexPath, const Aws::String
&errorPage,
        const Aws::Client::ClientConfiguration
&clientConfig) {
    Aws::S3::S3Client client(clientConfig);

    Aws::S3::Model::IndexDocument indexDocument;
    indexDocument.SetSuffix(indexPath);

    Aws::S3::Model::ErrorDocument errorDocument;
    errorDocument.SetKey(errorPage);

    Aws::S3::Model::WebsiteConfiguration websiteConfiguration;
    websiteConfiguration.SetIndexDocument(indexDocument);
    websiteConfiguration.SetErrorDocument(errorDocument);

    Aws::S3::Model::PutBucketWebsiteRequest request;
    request.SetBucket(bucketName);
    request.SetWebsiteConfiguration(websiteConfiguration);

    Aws::S3::Model::PutBucketWebsiteOutcome outcome =
        client.PutBucketWebsite(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error: PutBucketWebsite: "
            << outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Success: Set website configuration for bucket '"
            << bucketName << "'." << std::endl;
    }

    return outcome.IsSuccess();
}

```

- Para obter detalhes da API, consulte [PutBucketWebsite](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

Aplica uma configuração de site estático ao bucket `my-bucket`:

```
aws s3api put-bucket-website --bucket my-bucket --website-configuration file://website.json
```

O arquivo `website.json` é um documento JSON na pasta atual que especifica páginas de índice e erro para o site:

```
{
  "IndexDocument": {
    "Suffix": "index.html"
  },
  "ErrorDocument": {
    "Key": "error.html"
  }
}
```

- Para obter detalhes da API, consulte [PutBucketWebsite](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.IndexDocument;
import software.amazon.awssdk.services.s3.model.PutBucketWebsiteRequest;
import software.amazon.awssdk.services.s3.model.WebsiteConfiguration;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.regions.Region;
```

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

            Usage:    <bucketName> [indexdoc]\s

            Where:
                bucketName    - The Amazon S3 bucket to set the website
configuration on.\s
                indexdoc    - The index document, ex. 'index.html'
                            If not specified, 'index.html' will be set.

            """;

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

        String bucketName = args[0];
        String indexDoc = "index.html";
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        setWebsiteConfig(s3, bucketName, indexDoc);
        s3.close();
    }

    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 AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Defina a configuração do site.

```
import { PutBucketWebsiteCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// Set up a bucket as a static website.
```

```
// The bucket needs to be publicly accessible.
export const main = async () => {
  const command = new PutBucketWebsiteCommand({
    Bucket: "test-bucket",
    WebsiteConfiguration: {
      ErrorDocument: {
        // The object key name to use when a 4XX class error occurs.
        Key: "error.html",
      },
      IndexDocument: {
        // A suffix that is appended to a request that is for a directory.
        Suffix: "index.html",
      },
    },
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).
- Para obter detalhes da API, consulte [PutBucketWebsite](#) na Referência da API AWS SDK for JavaScript.

PowerShell

Tools for PowerShell

Exemplo 1: o comando habilita a hospedagem de sites para o bucket em questão com o documento de índice como “index.html” e o documento de erro como “error.html”.

```
Write-S3BucketWebsite -BucketName 's3testbucket' -
WebsiteConfiguration_IndexDocumentSuffix 'index.html' -
WebsiteConfiguration_ErrorDocument 'error.html'
```

- Para obter detalhes da API, consulte [PutBucketWebsite](#) na AWS Tools for PowerShell Cmdlet Reference.

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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
  end
end
```

```
rescue Aws::Errors::ServiceError => e
  puts "Couldn't configure #{@bucket_website.bucket.name} as a website. Here's
why: #{e.message}"
  false
end
end

# Example usage:
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 AWS SDK for Ruby.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutObject** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o PutObject.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- [Conceitos básicos de buckets e objetos](#)
- [Monitorar uploads e downloads](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <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;
    }
}
```

Faça upload de um objeto com criptografia do lado do servidor.

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

/// <summary>
/// This example shows how to upload an object to an Amazon Simple Storage
/// Service (Amazon S3) bucket with server-side encryption enabled.
/// </summary>
public class ServerSideEncryption
{
    public static async Task Main()
    {
        string bucketName = "doc-example-bucket";
        string keyName = "samplefile.txt";

        // If the AWS Region defined for your default user is different
        // from the Region where your Amazon S3 bucket is located,
        // pass the Region name to the Amazon S3 client object's constructor.
        // For example: RegionEndpoint.USWest2.
        IAmazonS3 client = new AmazonS3Client();

        await WritingAnObjectAsync(client, bucketName, keyName);
    }

    /// <summary>
    /// Upload a sample object include a setting for encryption.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
    /// to upload a file and apply server-side encryption.</param>
}
```

```
    /// <param name="bucketName">The name of the Amazon S3 bucket where the
    /// encrypted object will reside.</param>
    /// <param name="keyName">The name for the object that you want to
    /// create in the supplied bucket.</param>
    public static async Task WritingAnObjectAsync(IAmazonS3 client, string
bucketName, string keyName)
    {
        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($"Encryption method used: {0}",
objectEncryption.ToString());
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"Error: '{ex.Message}' when writing an
object");
        }
    }
}
```

- Para obter detalhes da API, consulte [PutObject](#) na Referência da API AWS SDK for .NET.

Bash

AWS CLI com script Bash

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
#####  
# function errecho  
#  
# This function outputs everything sent to it to STDERR (standard error output).  
#####  
function errecho() {  
    printf "%s\n" "$*" 1>&2  
}  
  
#####  
# function copy_file_to_bucket  
#  
# This function creates a file in the specified bucket.  
#  
# Parameters:  
#     $1 - The name of the bucket to copy the file to.  
#     $2 - The path and file name of the local file to copy to the bucket.  
#     $3 - The key (name) to call the copy of the file in the bucket.  
#  
# Returns:  
#     0 - If successful.  
#     1 - If it fails.  
#####  
function copy_file_to_bucket() {  
    local response bucket_name source_file destination_file_name  
    bucket_name=$1  
    source_file=$2  
    destination_file_name=$3
```

```

response=$(aws s3api put-object \
  --bucket "$bucket_name" \
  --body "$source_file" \
  --key "$destination_file_name")

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
  errecho "ERROR: AWS reports put-object operation failed.\n$response"
  return 1
fi
}

```

- Para obter detalhes da API, consulte [PutObject](#) na Referência de comandos da AWS CLI.

C++

SDK para C++

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```

bool AwsDoc::S3::PutObject(const Aws::String &bucketName,
                          const Aws::String &fileName,
                          const Aws::Client::ClientConfiguration &clientConfig)
{
    Aws::S3::S3Client s3_client(clientConfig);

    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 be set according to your retrieval
    needs.
    request.SetKey(fileName);

    std::shared_ptr<Aws::IOStream> inputData =
        Aws::MakeShared<Aws::FStream>("SampleAllocationTag",
                                     fileName.c_str(),

```

```
std::ios_base::in |
std::ios_base::binary);

if (!*inputData) {
    std::cerr << "Error unable to read file " << fileName << std::endl;
    return false;
}

request.SetBody(inputData);

Aws::S3::Model::PutObjectOutcome outcome =
    s3_client.PutObject(request);

if (!outcome.IsSuccess()) {
    std::cerr << "Error: PutObject: " <<
        outcome.GetError().GetMessage() << std::endl;
}
else {
    std::cout << "Added object '" << fileName << "' to bucket '"
        << bucketName << "'.";
}

return outcome.IsSuccess();
}
```

- Para obter detalhes da API, consulte [PutObject](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

O seguinte exemplo usa o comando `put-object` para fazer upload de um objeto no Amazon S3:

```
aws s3api put-object --bucket text-content --key dir-1/my_images.tar.bz2 --body
my_images.tar.bz2
```

O seguinte exemplo mostra o upload de um arquivo de vídeo (o arquivo de vídeo é especificado usando a sintaxe do sistema de arquivos do Windows):

```
aws s3api put-object --bucket text-content --key dir-1/big-video-file.mp4 --body
e:\media\videos\f-sharp-3-data-services.mp4
```

Para obter mais informações sobre o upload de objetos, [Uploading Objects in the Guia do desenvolvedor do Amazon S3](#).

- Para obter detalhes da API, consulte [PutObject](#) na Referência de comandos da AWS CLI.

Go

SDK para Go V2

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
bucket
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// UploadFile reads from a file and puts the data into an object in a bucket.
func (basics BucketBasics) UploadFile(bucketName string, objectKey string,
    fileName string) error {
    file, err := os.Open(fileName)
    if err != nil {
        log.Printf("Couldn't open file %v to upload. Here's why: %v\n", fileName, err)
    } else {
        defer file.Close()
        _, err = basics.S3Client.PutObject(context.TODO(), &s3.PutObjectInput{
            Bucket: aws.String(bucketName),
```

```
    Key:    aws.String(objectKey),
    Body:   file,
  })
  if err != nil {
    log.Printf("Couldn't upload file %v to %v:%v. Here's why: %v\n",
      fileName, bucketName, objectKey, err)
  }
}
return err
}
```

- Para obter detalhes da API, consulte [PutObject](#) na Referência da API AWS SDK for Go.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Faça upload de um arquivo em um bucket usando um [S3Client](#).

```
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.PutObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.util.HashMap;
import java.util.Map;

/**
 * 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 PutObject {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <bucketName> <objectKey> <objectPath>\s

            Where:
                bucketName - The Amazon S3 bucket to upload an object into.
                objectKey - The object to upload (for example, book.pdf).
                objectPath - The path where the file is located (for example,
C:/AWS/book2.pdf).\s
            """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String objectKey = args[1];
        String objectPath = args[2];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        putS3Object(s3, bucketName, objectKey, objectPath);
        s3.close();
    }

    // This example uses RequestBody.fromFile to avoid loading the whole file
into
    // memory.
    public static void 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();

        s3.putObject(putOb, RequestBody.fromFile(new File(objectPath)));
        System.out.println("Successfully placed " + objectKey + " into bucket
" + bucketName);

    } catch (S3Exception e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
}
}
```

Use [S3TransferManager](#) para [fazer upload de um arquivo](#) em um bucket. Veja o [arquivo completo](#) e [teste](#).

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedFileUpload;
import software.amazon.awssdk.transfer.s3.model.FileUpload;
import software.amazon.awssdk.transfer.s3.model.UploadFileRequest;
import software.amazon.awssdk.transfer.s3.progress.LoggingTransferListener;
import java.net.URI;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.file.Paths;
import java.util.UUID;

    public String uploadFile(S3TransferManager transferManager, String
bucketName,

        String key, URI filePathURI) {
        UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
            .putObjectRequest(b -> b.bucket(bucketName).key(key))
            .source(Paths.get(filePathURI))
            .build();

        FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);
```

```
CompletedFileUpload uploadResult = fileUpload.completionFuture().join();
return uploadResult.response().eTag();
}
```

Faça upload de um objeto em um bucket e defina etiquetas usando um [S3Client](#).

```
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);
    }
}

// Return a byte array.
private static byte[] getObjectFile(String filePath) {
    FileInputStream fileInputStream = null;
    byte[] byteArray = null;

    try {
        File file = new File(filePath);
        byteArray = new byte[(int) file.length()];
        fileInputStream = new FileInputStream(file);
        fileInputStream.read(byteArray);

    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (fileInputStream != null) {
            try {
                fileInputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    return byteArray;
}
}
```

Faça upload de um objeto em um bucket e defina metadados usando um [S3Client](#).

```
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.PutObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.util.HashMap;
import java.util.Map;

/**
 * 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 PutObjectMetadata {
    public static void main(String[] args) {
        final String USAGE = ""

            Usage:
                <bucketName> <objectKey> <objectPath>\s

            Where:
                bucketName - The Amazon S3 bucket to upload an object into.
                objectKey - The object to upload (for example, book.pdf).
                objectPath - The path where the file is located (for example,
C:/AWS/book2.pdf).\s
                """;

        if (args.length != 3) {
            System.out.println(USAGE);
            System.exit(1);
        }

        String bucketName = args[0];
        String objectKey = args[1];
        String objectPath = args[2];
        System.out.println("Putting object " + objectKey + " into bucket " +
bucketName);
        System.out.println("  in bucket: " + bucketName);
    }
}
```

```
    Region region = Region.US_EAST_1;
    S3Client s3 = S3Client.builder()
        .region(region)
        .build();

    putS3Object(s3, bucketName, objectKey, objectPath);
    s3.close();
}

// This example uses RequestBody.fromFile to avoid loading the whole file
into
// memory.
public static void 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();

        s3.putObject(putOb, RequestBody.fromFile(new File(objectPath)));
        System.out.println("Successfully placed " + objectKey + " into bucket
" + bucketName);

    } catch (S3Exception e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
}
}
```

Faça upload de um objeto em um bucket e defina um valor de retenção de objetos usando um [S3Client](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutObjectRetentionRequest;
```

```
import software.amazon.awssdk.services.s3.model.ObjectLockRetention;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.time.Instant;
import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.ZoneOffset;

/**
 * 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 PutObjectRetention {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <key> <bucketName>\s

            Where:
                key - The name of the object (for example, book.pdf).\s
                bucketName - The Amazon S3 bucket name that contains the
object (for example, bucket1).\s
            """;

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

        String key = args[0];
        String bucketName = args[1];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        setRetentionPeriod(s3, key, bucketName);
        s3.close();
    }
}
```



```
}

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 AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Carregue o objeto.

```
import { PutObjectCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new PutObjectCommand({
    Bucket: "test-bucket",
    Key: "hello-s3.txt",
    Body: "Hello S3!",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).
- Para obter detalhes da API, consulte [PutObject](#) na Referência da API AWS SDK for JavaScript.

Kotlin

SDK for Kotlin

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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).asByteArray()
    }

    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 AWS SDK para Kotlin.

PHP

SDK para PHP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Carregue um objeto em um bucket.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

$file_name = __DIR__ . "/local-file-" . uniqid();
try {
    $this->s3client->putObject([
        'Bucket' => $this->bucketName,
        'Key' => $file_name,
        'SourceFile' => __DIR__ . '/testfile.txt'
    ]);
    echo "Uploaded $file_name to $this->bucketName.\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 AWS SDK for PHP.

PowerShell

Tools for PowerShell

Exemplo 1: este comando carrega o arquivo único “local-sample.txt” no Amazon S3, criando um objeto com a chave “sample.txt” no bucket “test-files”.

```
Write-S3Object -BucketName test-files -Key "sample.txt" -File .\local-sample.txt
```

Exemplo 2: este comando carrega o arquivo único “sample.txt” no Amazon S3, criando um objeto com a chave “sample.txt” no bucket “test-files”. Quando o parâmetro -Key não é fornecido, usa-se o nome do arquivo como chave do objeto do S3.

```
Write-S3Object -BucketName test-files -File .\sample.txt
```

Exemplo 3: este comando carrega o arquivo único “local-sample.txt” no Amazon S3, criando um objeto com a chave “prefix/to/sample.txt” no bucket “test-files”.

```
Write-S3Object -BucketName test-files -Key "prefix/to/sample.txt" -File .\local-sample.txt
```

Exemplo 4: este comando carrega todos os arquivos no subdiretório “Scripts” no bucket “test-files” e aplica o prefixo de chave comum “SampleScripts” a cada objeto. Cada arquivo carregado terá uma chave “SampleScripts/filename”, em que “filename” varia.

```
Write-S3Object -BucketName test-files -Folder .\Scripts -KeyPrefix SampleScripts\
```

Exemplo 5: este comando carrega todos os arquivos *.ps1 no diretório local “Scripts” no bucket “test-files” e aplica o prefixo de chave comum “SampleScripts” a cada objeto. Cada arquivo carregado terá uma chave “SampleScripts/filename.ps1”, em que “filename” varia.

```
Write-S3Object -BucketName test-files -Folder .\Scripts -KeyPrefix SampleScripts\  
-SearchPattern *.ps1
```

Exemplo 6: este comando cria um objeto do S3 contendo a string de conteúdo especificada com a chave “sample.txt”.

```
Write-S3Object -BucketName test-files -Key "sample.txt" -Content "object  
contents"
```

Exemplo 7: este comando carrega o arquivo especificado (o nome do arquivo é usado como chave) e aplica as tags especificadas ao novo objeto.

```
Write-S3Object -BucketName test-files -File "sample.txt" -TagSet  
@{Key="key1";Value="value1"},@{Key="key2";Value="value2"}
```

Exemplo 8: este comando carrega recursivamente a pasta especificada e aplica as tags especificadas a todos os novos objetos.

```
Write-S3Object -BucketName test-files -Folder . -KeyPrefix "TaggedFiles" -Recurse  
-TagSet @{Key="key1";Value="value1"},@{Key="key2";Value="value2"}
```

- Para obter detalhes da API, consulte [PutObject](#) na AWS Tools for PowerShell Cmdlet Reference.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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

# Example usage:
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

# Example usage:
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 do 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

# Example usage:
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 AWS SDK for Ruby.

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
pub async fn upload_object(
    client: &Client,
    bucket_name: &str,
    file_name: &str,
    key: &str,
) -> Result<PutObjectOutput, SdkError<PutObjectError>> {
    let body = ByteStream::from_path(Path::new(file_name)).await;
    client
        .put_object()
        .bucket(bucket_name)
        .key(key)
        .body(body.unwrap())
        .send()
        .await
}
```

- Para obter detalhes da API, consulte [PutObject](#) na Referência da API AWS SDK para Rust.

SAP ABAP

SDK para SAP ABAP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
"Get contents of file from application server."
DATA lv_body TYPE xstring.
OPEN DATASET iv_file_name FOR INPUT IN BINARY MODE.
READ DATASET iv_file_name INTO lv_body.
CLOSE DATASET iv_file_name.

"Upload/put an object to an S3 bucket."
TRY.
    lo_s3->putobject(
        iv_bucket = iv_bucket_name
        iv_key = iv_file_name
```

```
        iv_body = lv_body
    ).
    MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- Para obter detalhes da API, consulte [PutObject](#) na Referência da API AWS SDK para SAP ABAP.

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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 ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutObjectAcl** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `PutObjectAcl`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Gerenciar listas de controle de acesso \(ACLs\)](#)

C++

SDK para C++

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
bool AwsDoc::S3::PutObjectAcl(const Aws::String &bucketName,
                              const Aws::String &objectKey,
                              const Aws::String &ownerID,
                              const Aws::String &granteePermission,
                              const Aws::String &granteeType,
                              const Aws::String &granteeID,
                              const Aws::Client::ClientConfiguration
&clientConfig,
                              const Aws::String &granteeDisplayName,
                              const Aws::String &granteeEmailAddress,
                              const Aws::String &granteeURI) {
    Aws::S3::S3Client s3_client(clientConfig);

    Aws::S3::Model::Owner owner;
    owner.SetID(ownerID);

    Aws::S3::Model::Grantee grantee;
    grantee.SetType(SetGranteeType(granteeType));

    if (!granteeEmailAddress.empty()) {
        grantee.SetEmailAddress(granteeEmailAddress);
    }

    if (!granteeID.empty()) {
        grantee.SetID(granteeID);
    }

    if (!granteeDisplayName.empty()) {
        grantee.SetDisplayName(granteeDisplayName);
    }

    if (!granteeURI.empty()) {
```

```

        grantee.SetURI(granteeURI);
    }

    Aws::S3::Model::Grant grant;
    grant.SetGrantee(grantee);
    grant.SetPermission(SetGranteePermission(granteePermission));

    Aws::Vector<Aws::S3::Model::Grant> grants;
    grants.push_back(grant);

    Aws::S3::Model::AccessControlPolicy acp;
    acp.SetOwner(owner);
    acp.SetGrants(grants);

    Aws::S3::Model::PutObjectAclRequest request;
    request.SetAccessControlPolicy(acp);
    request.SetBucket(bucketName);
    request.SetKey(objectKey);

    Aws::S3::Model::PutObjectAclOutcome outcome =
        s3_client.PutObjectAcl(request);

    if (!outcome.IsSuccess()) {
        auto error = outcome.GetError();
        std::cerr << "Error: PutObjectAcl: " << error.GetExceptionName()
            << " - " << error.GetMessage() << std::endl;
    }
    else {
        std::cout << "Successfully added an ACL to the object '" << objectKey
            << "' in the bucket '" << bucketName << "'." << std::endl;
    }

    return outcome.IsSuccess();
}

//! Routine which converts a human-readable string to a built-in type
enumeration.
/*!
 \sa SetGranteePermission()
 \param access Human readable string.
 */

Aws::S3::Model::Permission SetGranteePermission(const Aws::String &access) {
    if (access == "FULL_CONTROL")

```

```
        return Aws::S3::Model::Permission::FULL_CONTROL;
    if (access == "WRITE")
        return Aws::S3::Model::Permission::WRITE;
    if (access == "READ")
        return Aws::S3::Model::Permission::READ;
    if (access == "WRITE_ACP")
        return Aws::S3::Model::Permission::WRITE_ACP;
    if (access == "READ_ACP")
        return Aws::S3::Model::Permission::READ_ACP;
    return Aws::S3::Model::Permission::NOT_SET;
}

//! Routine which converts a human-readable string to a built-in type
    enumeration.
/*!
    \sa SetGranteeType()
    \param type Human readable string.
*/

Aws::S3::Model::Type SetGranteeType(const Aws::String &type) {
    if (type == "Amazon customer by email")
        return Aws::S3::Model::Type::AmazonCustomerByEmail;
    if (type == "Canonical user")
        return Aws::S3::Model::Type::CanonicalUser;
    if (type == "Group")
        return Aws::S3::Model::Type::Group;
    return Aws::S3::Model::Type::NOT_SET;
}
```

- Para obter detalhes da API, consulte [PutObjectAcl](#) na Referência da API AWS SDK for C++.

CLI

AWS CLI

O seguinte comando concede `full control` para dois usuários da AWS (`user1@example.com` e `user2@example.com`) e permissão `read` para todos:


```
aws s3api put-object-acl --bucket MyBucket --key file.txt --grant-full-control
emailaddress=user1@example.com,emailaddress=user2@example.com --grant-read
uri=http://acs.amazonaws.com/groups/global/AllUsers
```

Consulte <http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html> para obter detalhes sobre ACLs personalizadas (os comandos da ACL s3api, como `put-object-acl`, usam a mesma notação abreviada de argumentos).

- Para obter detalhes da API, consulte [PutObjectAcl](#) na Referência de comandos da AWS CLI.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutObjectLegalHold** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `PutObjectLegalHold`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Bloquear objetos do Amazon S3](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <summary>
/// Set or modify a legal hold on an object in an S3 bucket.
/// </summary>
/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The key of the object.</param>
/// <param name="holdStatus">The On or Off status for the legal hold.</param>
/// <returns>True if successful.</returns>
public async Task<bool> ModifyObjectLegalHold(string bucketName,
    string objectKey, ObjectLockLegalHoldStatus holdStatus)
{
    try
    {
        var request = new PutObjectLegalHoldRequest()
        {
            BucketName = bucketName,
            Key = objectKey,
            LegalHold = new ObjectLockLegalHold()
            {
                Status = holdStatus
            }
        };

        var response = await _amazonS3.PutObjectLegalHoldAsync(request);
        Console.WriteLine($"\\tModified legal hold for {objectKey} in
{bucketName}.");
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"\\tError modifying legal hold: '{ex.Message}'");
        return false;
    }
}
```

```
}
```

- Consulte detalhes da API em [PutObjectLegalHold](#) na Referência de API do AWS SDK for .NET.

CLI

AWS CLI

Como aplicar uma retenção legal a um objeto

O exemplo de `put-object-legal-hold` a seguir define uma retenção legal no objeto `doc1.rtf`.

```
aws s3api put-object-legal-hold \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf \  
  --legal-hold Status=ON
```

Este comando não produz saída.

- Consulte detalhes da API em [PutObjectLegalHold](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
// Set or modify a legal hold on an object in an S3 bucket.  
public void modifyObjectLegalHold(String bucketName, String objectKey,  
boolean legalHoldOn) {  
    ObjectLockLegalHold legalHold ;  
    if (legalHoldOn) {
```

```
        legalHold = ObjectLockLegalHold.builder()
            .status(ObjectLockLegalHoldStatus.ON)
            .build();
    } else {
        legalHold = ObjectLockLegalHold.builder()
            .status(ObjectLockLegalHoldStatus.OFF)
            .build();
    }

    PutObjectLegalHoldRequest legalHoldRequest =
PutObjectLegalHoldRequest.builder()
    .bucket(bucketName)
    .key(objectKey)
    .legalHold(legalHold)
    .build();

    getClient().putObjectLegalHold(legalHoldRequest) ;
    System.out.println("Modified legal hold for "+ objectKey +" in
"+bucketName +".");
}
```

- Consulte detalhes da API em [PutObjectLegalHold](#) na Referência de API do AWS SDK for Java 2.x.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutObjectLockConfiguration** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o `PutObjectLockConfiguration`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Bloquear objetos do Amazon S3](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Defina a configuração de Bloqueio de Objetos de um bucket.

```
/// <summary>
/// Enable object lock on an existing bucket.
/// </summary>
/// <param name="bucketName">The name of the bucket to modify.</param>
/// <returns>True if successful.</returns>
public async Task<bool> EnableObjectLockOnBucket(string bucketName)
{
    try
    {
        // First, enable Versioning on the bucket.
        await _amazonS3.PutBucketVersioningAsync(new
PutBucketVersioningRequest()
        {
            BucketName = bucketName,
            VersioningConfig = new S3BucketVersioningConfig()
            {
                EnableMfaDelete = false,
                Status = VersionStatus.Enabled
            }
        });

        var request = new PutObjectLockConfigurationRequest()
        {
            BucketName = bucketName,
            ObjectLockConfiguration = new ObjectLockConfiguration()
            {
                ObjectLockEnabled = new ObjectLockEnabled("Enabled"),
            },
        };
    }
}
```

```

        var response = await
        _amazonS3.PutObjectLockConfigurationAsync(request);
        Console.WriteLine($"{bucketName}\tAdded an object lock policy to bucket
{bucketName}.");
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error modifying object lock: '{ex.Message}'");
        return false;
    }
}

```

Defina o período de retenção padrão de um bucket.

```

/// <summary>
/// Set or modify a retention period on an S3 bucket.
/// </summary>
/// <param name="bucketName">The bucket to modify.</param>
/// <param name="retention">The retention mode.</param>
/// <param name="retainUntilDate">The date for retention until.</param>
/// <returns>True if successful.</returns>
public async Task<bool> ModifyBucketDefaultRetention(string bucketName, bool
enableObjectLock, ObjectLockRetentionMode retention, DateTime retainUntilDate)
{
    var enabledString = enableObjectLock ? "Enabled" : "Disabled";
    var timeDifference = retainUntilDate.Subtract(DateTime.Now);
    try
    {
        // First, enable Versioning on the bucket.
        await _amazonS3.PutBucketVersioningAsync(new
PutBucketVersioningRequest()
        {
            BucketName = bucketName,
            VersioningConfig = new S3BucketVersioningConfig()
            {
                EnableMfaDelete = false,
                Status = VersionStatus.Enabled
            }
        });

        var request = new PutObjectLockConfigurationRequest()

```

```
        {
            BucketName = bucketName,
            ObjectLockConfiguration = new ObjectLockConfiguration()
            {
                ObjectLockEnabled = new ObjectLockEnabled(enabledString),
                Rule = new ObjectLockRule()
                {
                    DefaultRetention = new DefaultRetention()
                    {
                        Mode = retention,
                        Days = timeDifference.Days // Can be specified in
days or years but not both.
                    }
                }
            }
        };

        var response = await
_amazonS3.PutObjectLockConfigurationAsync(request);
        Console.WriteLine($"\\tAdded a default retention to bucket
{bucketName}.");
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"\\tError modifying object lock: '{ex.Message}'");
        return false;
    }
}
```

- Consulte detalhes da API em [PutObjectLockConfiguration](#) na Referência de API do AWS SDK for .NET.

CLI

AWS CLI

Como definir a configuração de bloqueio de objetos em um bucket

O exemplo de `put-object-lock-configuration` a seguir define um bloqueio de objetos de 50 dias no bucket especificado.


```
aws s3api put-object-lock-configuration \  
  --bucket my-bucket-with-object-lock \  
  --object-lock-configuration '{ "ObjectLockEnabled": "Enabled", "Rule":  
  { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 } } }'
```

Este comando não produz saída.

- Consulte detalhes da API em [PutObjectLockConfiguration](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Defina a configuração de Bloqueio de Objetos de um bucket.

```
// Enable object lock on an existing bucket.  
public void enableObjectLockOnBucket(String bucketName) {  
    try {  
        VersioningConfiguration versioningConfiguration =  
VersioningConfiguration.builder()  
            .status(BucketVersioningStatus.ENABLED)  
            .build();  
  
        PutBucketVersioningRequest putBucketVersioningRequest =  
PutBucketVersioningRequest.builder()  
            .bucket(bucketName)  
            .versioningConfiguration(versioningConfiguration)  
            .build();  
  
        // Enable versioning on the bucket.  
        getClient().putBucketVersioning(putBucketVersioningRequest);  
        PutObjectLockConfigurationRequest request =  
PutObjectLockConfigurationRequest.builder()  
            .bucket(bucketName)  
            .objectLockConfiguration(ObjectLockConfiguration.builder()
```

```
        .objectLockEnabled(ObjectLockEnabled.ENABLED)
        .build())
    .build();

    getClient().putObjectLockConfiguration(request);
    System.out.println("Successfully enabled object lock on
"+bucketName);

    } catch (S3Exception ex) {
        System.out.println("Error modifying object lock: '" + ex.getMessage()
+ "'");
    }
}
```

Defina o período de retenção padrão de um bucket.

```
// Set or modify a retention period on an S3 bucket.
public void modifyBucketDefaultRetention(String bucketName) {
    VersioningConfiguration versioningConfiguration =
VersioningConfiguration.builder()
        .mfaDelete(MFADelete.DISABLED)
        .status(BucketVersioningStatus.ENABLED)
        .build();

    PutBucketVersioningRequest versioningRequest =
PutBucketVersioningRequest.builder()
        .bucket(bucketName)
        .versioningConfiguration(versioningConfiguration)
        .build();

    getClient().putBucketVersioning(versioningRequest);
    DefaultRetention rention = DefaultRetention.builder()
        .days(1)
        .mode(ObjectLockRetentionMode.GOVERNANCE)
        .build();

    ObjectLockRule lockRule = ObjectLockRule.builder()
        .defaultRetention(rention)
        .build();

    ObjectLockConfiguration objectLockConfiguration =
ObjectLockConfiguration.builder()
```

```
        .objectLockEnabled(ObjectLockEnabled.ENABLED)
        .rule(lockRule)
        .build();

        PutObjectLockConfigurationRequest putObjectLockConfigurationRequest =
PutObjectLockConfigurationRequest.builder()
        .bucket(bucketName)
        .objectLockConfiguration(objectLockConfiguration)
        .build();

getClient().putObjectLockConfiguration(putObjectLockConfigurationRequest) ;
        System.out.println("Added a default retention to bucket "+bucketName
+".");
    }
}
```

- Consulte detalhes da API em [PutObjectLockConfiguration](#) na Referência de API do AWS SDK for Java 2.x.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **PutObjectRetention** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `PutObjectRetention`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- [Bloquear objetos do Amazon S3](#)

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/// <summary>
/// Set or modify a retention period on an object in an S3 bucket.
/// </summary>
/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The key of the object.</param>
/// <param name="retention">The retention mode.</param>
/// <param name="retainUntilDate">The date retention expires.</param>
/// <returns>True if successful.</returns>
public async Task<bool> ModifyObjectRetentionPeriod(string bucketName,
    string objectKey, ObjectLockRetentionMode retention, DateTime
retainUntilDate)
{
    try
    {
        var request = new PutObjectRetentionRequest()
        {
            BucketName = bucketName,
            Key = objectKey,
            Retention = new ObjectLockRetention()
            {
                Mode = retention,
                RetainUntilDate = retainUntilDate
            }
        };

        var response = await _amazonS3.PutObjectRetentionAsync(request);
        Console.WriteLine($"{objectKey} in {bucketName}
until {retainUntilDate:d}.");
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
```

```
        Console.WriteLine($"{\tError modifying retention period:
'{ex.Message}'}");
        return false;
    }
}
```

- Consulte detalhes da API em [PutObjectRetention](#) na Referência de API do AWS SDK for .NET.

CLI

AWS CLI

Como definir uma configuração de retenção de objetos para um objeto

O exemplo de `put-object-retention` a seguir define uma configuração de retenção de objetos para o objeto especificado até 2025-01-01.

```
aws s3api put-object-retention \
  --bucket my-bucket-with-object-lock \
  --key doc1.rtf \
  --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate":
"2025-01-01T00:00:00" }'
```

Este comando não produz saída.

- Consulte detalhes da API em [PutObjectRetention](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
// Set or modify a retention period on an object in an S3 bucket.
```

```
public void modifyObjectRetentionPeriod(String bucketName, String objectKey)
{
    // Calculate the instant one day from now.
    Instant futureInstant = Instant.now().plus(1, ChronoUnit.DAYS);

    // Convert the Instant to a ZonedDateTime object with a specific time
    zone.
    ZonedDateTime zonedDateTime =
futureInstant.atZone(ZoneId.systemDefault());

    // Define a formatter for human-readable output.
    DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd
HH:mm:ss");

    // Format the ZonedDateTime object to a human-readable date string.
    String humanReadableDate = formatter.format(zonedDateTime);

    // Print the formatted date string.
    System.out.println("Formatted Date: " + humanReadableDate);
    ObjectLockRetention retention = ObjectLockRetention.builder()
        .mode(ObjectLockRetentionMode.GOVERNANCE)
        .retainUntilDate(futureInstant)
        .build();

    PutObjectRetentionRequest retentionRequest =
PutObjectRetentionRequest.builder()
        .bucket(bucketName)
        .key(objectKey)
        .retention(retention)
        .build();

    getClient().putObjectRetention(retentionRequest);
    System.out.println("Set retention for "+objectKey +" in " +bucketName +"
until "+ humanReadableDate +".");
}
```

- Consulte detalhes da API em [PutObjectRetention](#) na Referência de API do AWS SDK for Java 2.x.

PowerShell

Tools for PowerShell

Exemplo 1: o comando habilita o modo de retenção de governança até a data “31st Dec 2019 00:00:00” para o objeto “testfile.txt” no bucket do S3 em questão.

```
Write-S3ObjectRetention -BucketName 's3buckettesting' -Key 'testfile.txt' -
Retention_Mode GOVERNANCE -Retention_RetainUntilDate "2019-12-31T00:00:00"
```

- Para obter detalhes da API, consulte [PutObjectRetention](#) na AWS Tools for PowerShell Cmdlet Reference.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **RestoreObject** com um SDK da AWS ou uma ferramenta de linha de comandos

Os exemplos de código a seguir mostram como usar o RestoreObject.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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

/// <summary>
/// This example shows how to restore an archived object in an Amazon
```

```
/// Simple Storage Service (Amazon S3) bucket.
/// </summary>
public class RestoreArchivedObject
{
    public static void Main()
    {
        string bucketName = "doc-example-bucket";
        string objectKey = "archived-object.txt";

        // Specify your bucket region (an example region is shown).
        RegionEndpoint bucketRegion = RegionEndpoint.USWest2;

        IAmazonS3 client = new AmazonS3Client(bucketRegion);
        RestoreObjectAsync(client, bucketName, objectKey).Wait();
    }

    /// <summary>
    /// This method restores an archived object from an Amazon S3 bucket.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// RestoreObjectAsync.</param>
    /// <param name="bucketName">A string representing the name of the
    /// bucket where the object was located before it was archived.</param>
    /// <param name="objectKey">A string representing the name of the
    /// archived object to restore.</param>
    public 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($"Error: {amazonS3Exception.Message}");
        }
    }

    /// <summary>
    /// This method retrieves the status of the object's restoration.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// GetObjectMetadataAsync.</param>
    /// <param name="bucketName">A string representing the name of the Amazon
    /// S3 bucket which contains the archived object.</param>
    /// <param name="objectKey">A string representing the name of the
    /// archived object you want to restore.</param>
    public static async Task CheckRestorationStatusAsync(IAmazonS3 client,
string bucketName, string objectKey)
    {
        GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest()
        {
            BucketName = bucketName,
            Key = objectKey,
        };

        GetObjectMetadataResponse response = await
client.GetObjectMetadataAsync(metadataRequest);

        var restStatus = response.RestoreInProgress ? "in-progress" :
"finished or failed";
        Console.WriteLine($"Restoration status: {restStatus}");
    }
}
```

- Para obter detalhes da API, consulte [RestoreObject](#) em Referência da API AWS SDK for .NET.

CLI

AWS CLI

Como criar uma solicitação de restauração para um objeto

O exemplo de `restore-object` a seguir restaura o objeto Amazon S3 Glacier especificado do bucket `my-glacier-bucket` por 10 dias.

```
aws s3api restore-object \  
  --bucket my-glacier-bucket \  
  --key doc1.rtf \  
  --restore-request Days=10
```

Este comando não produz saída.

- Para obter detalhes da API, consulte [RestoreObject](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.s3.S3Client;  
import software.amazon.awssdk.services.s3.model.RestoreRequest;  
import software.amazon.awssdk.services.s3.model.GlacierJobParameters;  
import software.amazon.awssdk.services.s3.model.RestoreObjectRequest;  
import software.amazon.awssdk.services.s3.model.S3Exception;  
import software.amazon.awssdk.services.s3.model.Tier;  
  
/*  
 * For more information about restoring an object, see "Restoring an archived  
 * object" at  
 * https://docs.aws.amazon.com/AmazonS3/latest/userguide/restoring-objects.html  
 */
```

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

            Usage:
                <bucketName> <keyName> <expectedBucketOwner>

            Where:
                bucketName - The Amazon S3 bucket name.\s
                keyName - The key name of an object with a Storage class
value of Glacier.\s
                expectedBucketOwner - The account that owns the bucket (you
can obtain this value from the AWS Management Console).\s
            """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String keyName = args[1];
        String expectedBucketOwner = args[2];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        restoreS3Object(s3, bucketName, keyName, expectedBucketOwner);
        s3.close();
    }

    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 AWS SDK for Java 2.x.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **SelectObjectContent** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o `SelectObjectContent`.

CLI

AWS CLI

Como filtrar o conteúdo de um objeto do Amazon S3 com base em uma instrução SQL

O exemplo de `select-object-content` a seguir filtra o objeto `my-data-file.csv` com a instrução SQL especificada e envia a saída para um arquivo.

```
aws s3api select-object-content \  
  --bucket my-bucket \  
  --key my-data-file.csv \  
  --expression "select * from s3object limit 100" \  
  --expression-type 'SQL' \  
  --input-serialization '{"CSV": {}, "CompressionType": "NONE"}' \  
  --output-serialization '{"CSV": {}}' "output.csv"
```

Este comando não produz saída.

- Consulte detalhes da API em [SelectObjectContent](#) na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

O exemplo a seguir mostra uma consulta usando um objeto JSON. O [exemplo completo](#) também mostra o uso de um objeto CSV.

```
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import software.amazon.awssdk.core.async.AsyncRequestBody;  
import software.amazon.awssdk.core.async.BlockingInputStreamAsyncRequestBody;  
import software.amazon.awssdk.core.exception.SdkException;  
import software.amazon.awssdk.services.s3.S3AsyncClient;  
import software.amazon.awssdk.services.s3.model.CSVInput;  
import software.amazon.awssdk.services.s3.model.CSVOutput;  
import software.amazon.awssdk.services.s3.model.CompressionType;  
import software.amazon.awssdk.services.s3.model.ExpressionType;  
import software.amazon.awssdk.services.s3.model.FileHeaderInfo;  
import software.amazon.awssdk.services.s3.model.InputSerialization;  
import software.amazon.awssdk.services.s3.model.JSONInput;  
import software.amazon.awssdk.services.s3.model.JSONOutput;
```

```
import software.amazon.awssdk.services.s3.model.JSONType;
import software.amazon.awssdk.services.s3.model.ObjectIdentifier;
import software.amazon.awssdk.services.s3.model.OutputSerialization;
import software.amazon.awssdk.services.s3.model.Progress;
import software.amazon.awssdk.services.s3.model.PutObjectResponse;
import software.amazon.awssdk.services.s3.model.SelectObjectContentRequest;
import
    software.amazon.awssdk.services.s3.model.SelectObjectContentResponseHandler;
import software.amazon.awssdk.services.s3.model.Stats;

import java.io.IOException;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;
import java.util.UUID;
import java.util.concurrent.CompletableFuture;

public class SelectObjectContentExample {
    static final Logger logger =
        LoggerFactory.getLogger(SelectObjectContentExample.class);
    static final String BUCKET_NAME = "select-object-content-" +
        UUID.randomUUID();
    static final S3AsyncClient s3AsyncClient = S3AsyncClient.create();
    static String FILE_CSV = "csv";
    static String FILE_JSON = "json";
    static String URL_CSV = "https://raw.githubusercontent.com/mledoze/countries/
master/dist/countries.csv";
    static String URL_JSON = "https://raw.githubusercontent.com/mledoze/
countries/master/dist/countries.json";

    public static void main(String[] args) {
        SelectObjectContentExample selectObjectContentExample = new
        SelectObjectContentExample();
        try {
            SelectObjectContentExample.setUp();
            selectObjectContentExample.runSelectObjectContentMethodForJSON();
            selectObjectContentExample.runSelectObjectContentMethodForCSV();
        } catch (SdkException e) {
            logger.error(e.getMessage(), e);
            System.exit(1);
        } finally {
            SelectObjectContentExample.tearDown();
        }
    }
}
```

```
EventStreamInfo runSelectObjectContentMethodForJSON() {
    // Set up request parameters.
    final String queryExpression = "select * from s3object[*][*] c where
c.area < 350000";
    final String fileType = FILE_JSON;

    InputSerialization inputSerialization = InputSerialization.builder()
        .json(JSONInput.builder().type(JSONType.DOCUMENT).build())
        .compressionType(CompressionType.NONE)
        .build();

    OutputSerialization outputSerialization = OutputSerialization.builder()
        .json(JSONOutput.builder().recordDelimiter(null).build())
        .build();

    // Build the SelectObjectContentRequest.
    SelectObjectContentRequest select = SelectObjectContentRequest.builder()
        .bucket(BUCKET_NAME)
        .key(FILE_JSON)
        .expression(queryExpression)
        .expressionType(ExpressionType.SQL)
        .inputSerialization(inputSerialization)
        .outputSerialization(outputSerialization)
        .build();

    EventStreamInfo eventStreamInfo = new EventStreamInfo();
    // Call the selectObjectContent method with the request and a response
    handler.
    // Supply an EventStreamInfo object to the response handler to gather
    records and information from the response.
    s3AsyncClient.selectObjectContent(select,
    buildResponseHandler(eventStreamInfo)).join();

    // Log out information gathered while processing the response stream.
    long recordCount = eventStreamInfo.getRecords().stream().mapToInt(record
->
        record.split("\n").length
    ).sum();
    logger.info("Total records {}: {}", fileType, recordCount);
    logger.info("Visitor onRecords for fileType {} called {} times",
fileType, eventStreamInfo.getCountOnRecordsCalled());
    logger.info("Visitor onStats for fileType {}, {}", fileType,
eventStreamInfo.getStats());
}
```

```
        logger.info("Visitor onContinuations for fileType {}, {}", fileType,
eventStreamInfo.getCountContinuationEvents());
        return eventStreamInfo;
    }

    static SelectObjectContentResponseHandler
buildResponseHandler(EventStreamInfo eventStreamInfo) {
    // Use a Visitor to process the response stream. This visitor logs
information and gathers details while processing.
    final SelectObjectContentResponseHandler.Visitor visitor =
SelectObjectContentResponseHandler.Visitor.builder()
        .onRecords(r -> {
            logger.info("Record event received.");
            eventStreamInfo.addRecord(r.payload().asUtf8String());
            eventStreamInfo.incrementOnRecordsCalled();
        })
        .onCont(ce -> {
            logger.info("Continuation event received.");
            eventStreamInfo.incrementContinuationEvents();
        })
        .onProgress(pe -> {
            Progress progress = pe.details();
            logger.info("Progress event received:\n bytesScanned:
{}\nbytesProcessed: {}\nbytesReturned:{}",
                progress.bytesScanned(),
                progress.bytesProcessed(),
                progress.bytesReturned());
        })
        .onEnd(ee -> logger.info("End event received."))
        .onStats(se -> {
            logger.info("Stats event received.");
            eventStreamInfo.addStats(se.details());
        })
        .build();

    // Build the SelectObjectContentResponseHandler with the visitor that
processes the stream.
    return SelectObjectContentResponseHandler.builder()
        .subscriber(visitor).build();
}

// The EventStreamInfo class is used to store information gathered while
processing the response stream.
static class EventStreamInfo {
```



```
private final List<String> records = new ArrayList<>();
private Integer countOnRecordsCalled = 0;
private Integer countContinuationEvents = 0;
private Stats stats;

void incrementOnRecordsCalled() {
    countOnRecordsCalled++;
}

void incrementContinuationEvents() {
    countContinuationEvents++;
}

void addRecord(String record) {
    records.add(record);
}

void addStats(Stats stats) {
    this.stats = stats;
}

public List<String> getRecords() {
    return records;
}

public Integer getCountOnRecordsCalled() {
    return countOnRecordsCalled;
}

public Integer getCountContinuationEvents() {
    return countContinuationEvents;
}

public Stats getStats() {
    return stats;
}
}
```

- Consulte detalhes da API em [SelectObjectContent](#) na Referência de API do AWS SDK for Java 2.x.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **UploadPart** com um AWS SDK ou ferramenta de linha de comando

Os exemplos de código a seguir mostram como usar o UploadPart.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- [Realizar um carregamento fracionado](#)
- [Usar somas de verificação](#)

CLI

AWS CLI

O seguinte comando faz o upload da primeira parte de um upload fracionado iniciado com o comando `create-multipart-upload`:

```
aws s3api upload-part --bucket my-bucket --key 'multipart/01' --part-number 1 --  
body part01 --upload-id  
"dfRtDYU0WwCCcH43C3WfbkR0NycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6YjlX.cph0gtNBtJ8P3"
```

A opção `body` usa o nome ou o caminho de um arquivo local para upload (não use o prefixo `file://`). O tamanho mínimo de uma parte é de 5 MB. O ID de upload é retornado por `create-multipart-upload` e também pode ser recuperado com `list-multipart-uploads`. O bucket e a chave são especificados ao criar o upload fracionado.

Saída:

```
{  
  "ETag": "\"e868e0f4719e394144ef36531ee6824c\""  
}
```

Salve o valor do ETag de cada parte para etapas posteriores. Eles são necessários para concluir o upload fracionado.

- Para obter detalhes da API, consulte [UploadPart](#) na Referência de comandos da AWS CLI.

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
let upload_part_res = client
    .upload_part()
    .key(&key)
    .bucket(&bucket_name)
    .upload_id(upload_id)
    .body(stream)
    .part_number(part_number)
    .send()
    .await?;
upload_parts.push(
    CompletedPart::builder()
        .e_tag(upload_part_res.e_tag.unwrap_or_default())
        .part_number(part_number)
        .build(),
);

let completed_multipart_upload: CompletedMultipartUpload =
CompletedMultipartUpload::builder()
    .set_parts(Some(upload_parts))
    .build();
```

- Para obter detalhes da API, consulte [UploadPart](#) na Referência da API do AWS SDK para Rust.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). 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, em que é possível encontrar instruções sobre como configurar e executar o código.

Exemplos

- [Criar um URL pré-assinado para o Amazon S3 usando um AWS SDK](#)
- [Uma página da web que oferece uma lista de objetos do Amazon S3 usando um AWS SDK](#)
- [Fazer download de todos os objetos de um bucket do Amazon Simple Storage Service \(Amazon S3\) em um diretório local](#)
- [Obter um objeto do Amazon S3 de um ponto de acesso multirregional usando um AWS SDK](#)
- [Obter um objeto de um bucket do Amazon S3 usando um AWS SDK e especificando um cabeçalho If-Modified-Since](#)
- [Conceitos básicos de buckets e objetos do Amazon S3 usando um AWS SDK](#)
- [Conceitos básicos de criptografia para objetos do Amazon S3 usando um AWS SDK](#)
- [Conceitos básicos de etiquetas para objetos do Amazon S3 usando um AWS SDK](#)
- [Obter a configuração de retenção legal de um objeto do Amazon S3 usando um SDK da AWS](#)
- [Trabalhar com os recursos de bloqueio de objetos do Amazon S3 usando um SDK da AWS](#)
- [Gerenciar listas de controle de acesso \(ACLs\) para buckets do Amazon S3 usando um AWS SDK](#)
- [Gerenciar objetos do Amazon S3 versionados em lotes com uma função do Lambda usando um AWS SDK](#)
- [Analisar URIs do Amazon S3 usando um AWS SDK](#)
- [Executar uma cópia multipart de um objeto do Amazon S3 usando um AWS SDK](#)
- [Realizar um carregamento fragmentado em um objeto do Amazon S3 usando um SDK da AWS](#)
- [Monitorar o upload ou download de um objeto do Amazon S3 usando um SDK da AWS](#)
- [Exemplos de abordagens para testes de unidade e integração com SDK da AWS](#)
- [Fazer upload recursivo de um diretório local em um bucket do Amazon Simple Storage Service \(Amazon S3\)](#)
- [Fazer upload ou download de arquivos grandes de e para o Amazon S3 usando um AWS SDK](#)

- [Faça upload de um fluxo de tamanho desconhecido em um objeto do Amazon S3 usando um AWS SDK](#)
- [Usar somas de verificação para trabalhar com um objeto do Amazon S3 usando um AWS SDK](#)
- [Trabalhar com objetos versionados do Amazon S3 usando um AWS SDK](#)

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 Amazon S3 e carregar um objeto.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Gere um URL pré-assinado que possa executar uma ação do Amazon S3 por tempo limitado.

```
using System;
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;

public class GenPresignedUrl
{
    public static void Main()
    {
        const string bucketName = "doc-example-bucket";
        const string objectKey = "sample.txt";

        // Specify how long the presigned URL lasts, in hours
        const double timeoutDuration = 12;

        // Specify the AWS Region of your Amazon S3 bucket. If it is
        // different from the Region defined for the default user,
        // pass the Region to the constructor for the client. For
```

```
// example: new AmazonS3Client(RegionEndpoint.USEast1);

// If using the Region us-east-1, and server-side encryption with AWS
KMS, you must specify Signature Version 4.
// Region us-east-1 defaults to Signature Version 2 unless explicitly
set to Version 4 as shown below.
// For more details, see https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingAWSSDK.html#specify-signature-version
// and https://docs.aws.amazon.com/sdkfornet/v3/apidocs/items/Amazon/TAWSConfigsS3.html
AWSConfigsS3.UseSignatureVersion4 = true;
IAmazonS3 s3Client = new AmazonS3Client(RegionEndpoint.USEast1);

string urlString = GeneratePresignedURL(s3Client, bucketName,
objectKey, timeoutDuration);
Console.WriteLine($"The generated URL is: {urlString}.");
}

/// <summary>
/// Generate a presigned URL that can be used to access the file named
/// in the objectKey parameter for the amount of time specified in the
/// duration parameter.
/// </summary>
/// <param name="client">An initialized S3 client object used to call
/// the GetPresignedUrl method.</param>
/// <param name="bucketName">The name of the S3 bucket containing the
/// object for which to create the presigned URL.</param>
/// <param name="objectKey">The name of the object to access with the
/// presigned URL.</param>
/// <param name="duration">The length of time for which the presigned
/// URL will be valid.</param>
/// <returns>A string representing the generated presigned URL.</returns>
public static string GeneratePresignedURL(IAmazonS3 client, string
bucketName, string objectKey, double duration)
{
    string urlString = string.Empty;
    try
    {
        var request = new GetPreSignedUrlRequest()
        {
            BucketName = bucketName,
            Key = objectKey,
            Expires = DateTime.UtcNow.AddHours(duration),
        };
    }
}
```

```
        urlString = client.GetPreSignedURL(request);
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error: '{ex.Message}'");
    }

    return urlString;
}
}
```

Gere um URL pré-assinado e faça um upload usando esse URL.

```
using System;
using System.IO;
using System.Net.Http;
using System.Threading.Tasks;
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to upload an object to an Amazon Simple Storage
/// Service (Amazon S3) bucket using a presigned URL. The code first
/// creates a presigned URL and then uses it to upload an object to an
/// Amazon S3 bucket using that URL.
/// </summary>
public class UploadUsingPresignedURL
{
    private static HttpClient httpClient = new HttpClient();

    public static async Task Main()
    {
        string bucketName = "doc-example-bucket";
        string keyName = "samplefile.txt";
        string filePath = $"source\\{keyName}";

        // Specify how long the signed URL will be valid in hours.
        double timeoutDuration = 12;

        // Specify the AWS Region of your Amazon S3 bucket. If it is
```

```
// different from the Region defined for the default user,
// pass the Region to the constructor for the client. For
// example: new AmazonS3Client(RegionEndpoint.USEast1);

// If using the Region us-east-1, and server-side encryption with AWS
KMS, you must specify Signature Version 4.
// Region us-east-1 defaults to Signature Version 2 unless explicitly
set to Version 4 as shown below.
// For more details, see https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingAWSSDK.html#specify-signature-version
// and https://docs.aws.amazon.com/sdkfornet/v3/apidocs/items/Amazon/TAWSConfigsS3.html
AWSConfigsS3.UseSignatureVersion4 = true;
IAmazonS3 client = new AmazonS3Client(RegionEndpoint.USEast1);

var url = GeneratePreSignedURL(client, bucketName, keyName,
timeoutDuration);
var success = await UploadObject(filePath, url);

if (success)
{
    Console.WriteLine("Upload succeeded.");
}
else
{
    Console.WriteLine("Upload failed.");
}
}

/// <summary>
/// Uploads an object to an Amazon S3 bucket using the presigned URL
passed in
/// the url parameter.
/// </summary>
/// <param name="filePath">The path (including file name) to the local
/// file you want to upload.</param>
/// <param name="url">The presigned URL that will be used to upload the
/// file to the Amazon S3 bucket.</param>
/// <returns>A Boolean value indicating the success or failure of the
/// operation, based on the HttpResponseMessage.</returns>
public static async Task<bool> UploadObject(string filePath, string url)
{
    using var streamContent = new StreamContent(
        new FileStream(filePath, FileMode.Open, FileAccess.Read));
```




```
        var response = await httpClient.PutAsync(url, streamContent);
        return response.IsSuccessStatusCode;
    }

    /// <summary>
    /// Generates a presigned URL which will be used to upload an object to
    /// an Amazon S3 bucket.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// GetPreSignedURL.</param>
    /// <param name="bucketName">The name of the Amazon S3 bucket to which
the
    /// presigned URL will point.</param>
    /// <param name="objectKey">The name of the file that will be uploaded.</
param>
    /// <param name="duration">How long (in hours) the presigned URL will
    /// be valid.</param>
    /// <returns>The generated URL.</returns>
    public static string GeneratePreSignedURL(
        IAmazonS3 client,
        string bucketName,
        string objectKey,
        double duration)
    {
        var request = new GetPreSignedUrlRequest
        {
            BucketName = bucketName,
            Key = objectKey,
            Verb = HttpVerb.PUT,
            Expires = DateTime.UtcNow.AddHours(duration),
        };

        string url = client.GetPreSignedURL(request);
        return url;
    }
}
```

C++

SDK para C++

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Gere um URL pré-assinado para baixar um objeto.

```
//! Routine which demonstrates creating a pre-signed URL to download an object
    from an
    //! Amazon Simple Storage Service (Amazon S3) bucket.
    /*!
        \param bucketName: Name of the bucket.
        \param key: Name of an object key.
        \param expirationSeconds: Expiration in seconds for pre-signed URL.
        \param clientConfig: Aws client configuration.
        \return Aws::String: A pre-signed URL.
    */
    Aws::String AwsDoc::S3::GeneratePreSignedGetObjectURL(const Aws::String
        &bucketName,
                                                            const Aws::String &key,
                                                            uint64_t expirationSeconds,
                                                            const
    Aws::Client::ClientConfiguration &clientConfig) {
        Aws::S3::S3Client client(clientConfig);
        return client.GeneratePresignedUrl(bucketName, key,
    Aws::Http::HttpMethod::HTTP_GET,
                                                            expirationSeconds);
    }
```

Baixe usando a libcurl.

```
static size_t myCurlWriteBack(char *buffer, size_t size, size_t nitems, void
    *userdata) {
    Aws::StringStream *str = (Aws::StringStream *) userdata;

    if (nitems > 0) {
```

```
        str->write(buffer, size * nitems);
    }
    return size * nitems;
}

//! Utility routine to test GetObject with a pre-signed URL.
/*!
 \param presignedURL: A pre-signed URL to get an object from a bucket.
 \param resultString: A string to hold the result.
 \return bool: Function succeeded.
 */
bool AwsDoc::S3::GetObjectWithPresignedObjectURL(const Aws::String &presignedURL,
                                                  Aws::String &resultString) {
    CURL *curl = curl_easy_init();
    CURLcode result;

    std::stringstream outWriteString;

    result = curl_easy_setopt(curl, CURLOPT_WRITEDATA, &outWriteString);

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_WRITEDATA " << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, myCurlWriteBack);

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_WRITEFUNCTION" << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_URL, presignedURL.c_str());

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_URL" << std::endl;
        return false;
    }

    result = curl_easy_perform(curl);

    if (result != CURLE_OK) {
        std::cerr << "Failed to perform CURL request" << std::endl;
        return false;
    }
}
```

```

    }

    resultString = outWriteString.str();

    if (resultString.find( "<?xml" ) == 0)
    {
        std::cerr << "Failed to get object, response:\n" << resultString <<
std::endl;
        return false;
    }

    return true;
}

```

Gere um URL pré-assinado para carregar um objeto.

```

/*! Routine which demonstrates creating a pre-signed URL to upload an object to
an
Amazon Simple Storage Service (Amazon S3) bucket.
*/
\param bucketName: Name of the bucket.
\param key: Name of an object key.
\param clientConfig: Aws client configuration.
\return Aws::String: A pre-signed URL.
*/
Aws::String AwsDoc::S3::GeneratePreSignedPutObjectURL(const Aws::String
&bucketName,
                                                    const Aws::String &key,
                                                    uint64_t expirationSeconds,
                                                    const
Aws::Client::ClientConfiguration &clientConfig) {
    Aws::S3::S3Client client(clientConfig);
    return client.GeneratePresignedUrl(bucketName, key,
    Aws::Http::HttpMethod::HTTP_PUT,
                                                    expirationSeconds);
}

```

Carregue usando a libcurl.

```

static size_t myCurlReadBack(char *buffer, size_t size, size_t nitems, void
*userdata) {

```

```
Aws::StringStream *str = (Aws::StringStream *) userdata;

str->read(buffer, size * nitems);

return str->gcount();
}

static size_t myCurlWriteBack(char *buffer, size_t size, size_t nitems, void
*userdata) {
    Aws::StringStream *str = (Aws::StringStream *) userdata;

    if (nitems > 0) {
        str->write(buffer, size * nitems);
    }
    return size * nitems;
}

//! Utility routine to test PutObject with a pre-signed URL.
/*!
    \param presignedURL: A pre-signed URL to put an object in a bucket.
    \param data: Body of the PutObject request.
    \return bool: Function succeeded.
*/
bool AwsDoc::S3::PutStringWithPresignedObjectURL(const Aws::String &presignedURL,
                                                  const Aws::String &data) {

    CURL *curl = curl_easy_init();
    CURLcode result;

    Aws::StringStream readStringStream;
    readStringStream << data;
    result = curl_easy_setopt(curl, CURLOPT_READFUNCTION, myCurlReadBack);

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_READFUNCTION" << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_READDATA, &readStringStream);
    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_READDATA" << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_INFILESIZE_LARGE,
```

```
        (curl_off_t)data.size());

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_INFILESIZE_LARGE" << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, myCurlWriteBack);

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_WRITEFUNCTION" << std::endl;
        return false;
    }

    std::stringstream outWriteString;

    result = curl_easy_setopt(curl, CURLOPT_WRITEDATA, &outWriteString);

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_WRITEDATA " << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_URL, presignedURL.c_str());

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_URL" << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_UPLOAD, 1L);

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_PUT" << std::endl;
        return false;
    }

    result = curl_easy_perform(curl);

    if (result != CURLE_OK) {
        std::cerr << "Failed to perform CURL request" << std::endl;
        return false;
    }
}
```

```
std::string outString = outWriteString.str();
if (outString.empty()) {
    std::cout << "Successfully put object." << std::endl;
    return true;
}
else {
    std::cout << "A server error was encountered, output:\n" << outString
        << std::endl;
    return false;
}
}
```

Go

SDK para Go V2

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Crie funções que envolvam ações de pré-assinatura do S3.

```
// Presigner encapsulates the Amazon Simple Storage Service (Amazon S3) presign
actions
// used in the examples.
// It contains PresignClient, a client that is used to presign requests to Amazon
S3.
// Presigned requests contain temporary credentials and can be made from any HTTP
client.
type Presigner struct {
    PresignClient *s3.PresignClient
}

// GetObject makes a presigned request that can be used to get an object from a
bucket.
// The presigned request is valid for the specified number of seconds.
```

```
func (presigner Presigner) GetObject(
    bucketName string, objectKey string, lifetimeSecs int64)
(*v4.PresignedHTTPRequest, error) {
    request, err := presigner.PresignClient.PresignGetObject(context.TODO(),
    &s3.GetObjectInput{
        Bucket: aws.String(bucketName),
        Key:    aws.String(objectKey),
    }, func(opts *s3.PresignOptions) {
        opts.Expires = time.Duration(lifetimeSecs * int64(time.Second))
    })
    if err != nil {
        log.Printf("Couldn't get a presigned request to get %v:%v. Here's why: %v\n",
            bucketName, objectKey, err)
    }
    return request, err
}

// PutObject makes a presigned request that can be used to put an object in a
// bucket.
// The presigned request is valid for the specified number of seconds.
func (presigner Presigner) PutObject(
    bucketName string, objectKey string, lifetimeSecs int64)
(*v4.PresignedHTTPRequest, error) {
    request, err := presigner.PresignClient.PresignPutObject(context.TODO(),
    &s3.PutObjectInput{
        Bucket: aws.String(bucketName),
        Key:    aws.String(objectKey),
    }, func(opts *s3.PresignOptions) {
        opts.Expires = time.Duration(lifetimeSecs * int64(time.Second))
    })
    if err != nil {
        log.Printf("Couldn't get a presigned request to put %v:%v. Here's why: %v\n",
            bucketName, objectKey, err)
    }
    return request, err
}

// DeleteObject makes a presigned request that can be used to delete an object
// from a bucket.
```



```
func (presigner Presigner) DeleteObject(bucketName string, objectKey string)
(*v4.PresignedHTTPRequest, error) {
    request, err := presigner.PresignClient.PresignDeleteObject(context.TODO(),
&s3.DeleteObjectInput{
    Bucket: aws.String(bucketName),
    Key:    aws.String(objectKey),
})
    if err != nil {
        log.Printf("Couldn't get a presigned request to delete object %v. Here's why:
%v\n", objectKey, err)
    }
    return request, err
}
```

Execute um exemplo interativo que gera e usa URLs pré-assinados para fazer upload, fazer download e excluir um objeto do S3.

```
// RunPresigningScenario is an interactive example that shows you how to get
presigned
// HTTP requests that you can use to move data into and out of Amazon Simple
Storage
// Service (Amazon S3). The presigned requests contain temporary credentials and
can
// be used by an HTTP client.
//
// 1. Get a presigned request to put an object in a bucket.
// 2. Use the net/http package to use the presigned request to upload a local
file to the bucket.
// 3. Get a presigned request to get an object from a bucket.
// 4. Use the net/http package to use the presigned request to download the
object to a local file.
// 5. Get a presigned request to delete an object from a bucket.
// 6. Use the net/http package to use the presigned request to delete the object.
//
// This example creates an Amazon S3 presign client from the specified sdkConfig
so that
// you can replace it with a mocked or stubbed config for unit testing.
//
// It uses a questioner from the `demotools` package to get input during the
example.
```

```
// This package can be found in the ..\..\demotools folder of this repo.
//
// It uses an IHttpRequester interface to abstract HTTP requests so they can be
// mocked
// during testing.
func RunPresigningScenario(sdkConfig aws.Config, questioner
demotools.IQuestioner, httpRequester IHttpRequester) {
defer func() {
if r := recover(); r != nil {
fmt.Printf("Something went wrong with the demo.")
}
}()

log.Println(strings.Repeat("-", 88))
log.Println("Welcome to the Amazon S3 presigning demo.")
log.Println(strings.Repeat("-", 88))

s3Client := s3.NewFromConfig(sdkConfig)
bucketBasics := actions.BucketBasics{S3Client: s3Client}
presignClient := s3.NewPresignClient(s3Client)
presigner := actions.Presigner{PresignClient: presignClient}

bucketName := questioner.Ask("We'll need a bucket. Enter a name for a bucket "+
"you own or one you want to create:", demotools.NotEmpty{})
bucketExists, err := bucketBasics.BucketExists(bucketName)
if err != nil {
panic(err)
}
if !bucketExists {
err = bucketBasics.CreateBucket(bucketName, sdkConfig.Region)
if err != nil {
panic(err)
} else {
log.Println("Bucket created.")
}
}
log.Println(strings.Repeat("-", 88))

log.Printf("Let's presign a request to upload a file to your bucket.")
uploadFilename := questioner.Ask("Enter the path to a file you want to upload:",
demotools.NotEmpty{})
uploadKey := questioner.Ask("What would you like to name the uploaded object?",
demotools.NotEmpty{})
uploadFile, err := os.Open(uploadFilename)
```

```
if err != nil {
    panic(err)
}
defer uploadFile.Close()
presignedPutRequest, err := presigner.PutObject(bucketName, uploadKey, 60)
if err != nil {
    panic(err)
}
log.Printf("Got a presigned %v request to URL:\n\t%v\n",
presignedPutRequest.Method,
presignedPutRequest.URL)
log.Println("Using net/http to send the request...")
info, err := uploadFile.Stat()
if err != nil {
    panic(err)
}
putResponse, err := httpRequester.Put(presignedPutRequest.URL, info.Size(),
uploadFile)
if err != nil {
    panic(err)
}
log.Printf("%v object %v with presigned URL returned %v.",
presignedPutRequest.Method,
uploadKey, putResponse.StatusCode)
log.Println(strings.Repeat("-", 88))

log.Printf("Let's presign a request to download the object.")
questioner.Ask("Press Enter when you're ready.")
presignedGetRequest, err := presigner.GetObject(bucketName, uploadKey, 60)
if err != nil {
    panic(err)
}
log.Printf("Got a presigned %v request to URL:\n\t%v\n",
presignedGetRequest.Method,
presignedGetRequest.URL)
log.Println("Using net/http to send the request...")
getResponse, err := httpRequester.Get(presignedGetRequest.URL)
if err != nil {
    panic(err)
}
log.Printf("%v object %v with presigned URL returned %v.",
presignedGetRequest.Method,
uploadKey, getResponse.StatusCode)
defer getResponse.Body.Close()
```

```

downloadBody, err := io.ReadAll(getResponse.Body)
if err != nil {
    panic(err)
}
log.Printf("Downloaded %v bytes. Here are the first 100 of them:\n",
len(downloadBody))
log.Println(strings.Repeat("-", 88))
log.Println(string(downloadBody[:100]))
log.Println(strings.Repeat("-", 88))

log.Println("Let's presign a request to delete the object.")
questioner.Ask("Press Enter when you're ready.")
presignedDelRequest, err := presigner.DeleteObject(bucketName, uploadKey)
if err != nil {
    panic(err)
}
log.Printf("Got a presigned %v request to URL:\n\t%v\n",
presignedDelRequest.Method,
presignedDelRequest.URL)
log.Println("Using net/http to send the request...")
delResponse, err := httpRequester.Delete(presignedDelRequest.URL)
if err != nil {
    panic(err)
}
log.Printf("%v object %v with presigned URL returned %v.\n",
presignedDelRequest.Method,
uploadKey, delResponse.StatusCode)
log.Println(strings.Repeat("-", 88))

log.Println("Thanks for watching!")
log.Println(strings.Repeat("-", 88))
}

```

Defina um wrapper de solicitação HTTP usado pelo exemplo para fazer solicitações HTTP.

```

// IHttpRequester abstracts HTTP requests into an interface so it can be mocked
// during
// unit testing.
type IHttpRequester interface {
    Get(url string) (resp *http.Response, err error)
}

```

```
Put(url string, contentType int64, body io.Reader) (resp *http.Response, err
error)
Delete(url string) (resp *http.Response, err error)
}

// HttpRequester uses the net/http package to make HTTP requests during the
scenario.
type HttpRequester struct{}

func (httpReq HttpRequester) Get(url string) (resp *http.Response, err error) {
return http.Get(url)
}
func (httpReq HttpRequester) Put(url string, contentType int64, body io.Reader)
(resp *http.Response, err error) {
putRequest, err := http.NewRequest("PUT", url, body)
if err != nil {
return nil, err
}
putRequest.ContentLength = contentType
return http.DefaultClient.Do(putRequest)
}
func (httpReq HttpRequester) Delete(url string) (resp *http.Response, err error)
{
delRequest, err := http.NewRequest("DELETE", url, nil)
if err != nil {
return nil, err
}
return http.DefaultClient.Do(delRequest)
}
```

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Gere um URL pré-assinado para um objeto e faça download dele (solicitação GET).

Importações.

```
import com.example.s3.util.PresignUrlUtils;
import org.slf4j.Logger;
import software.amazon.awssdk.http.HttpExecuteRequest;
import software.amazon.awssdk.http.HttpExecuteResponse;
import software.amazon.awssdk.http.SdkHttpClient;
import software.amazon.awssdk.http.SdkHttpMethod;
import software.amazon.awssdk.http.SdkHttpRequest;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.presigner.S3Presigner;
import
    software.amazon.awssdk.services.s3.presigner.model.GetObjectPresignRequest;
import
    software.amazon.awssdk.services.s3.presigner.model.PresignedGetObjectRequest;
import software.amazon.awssdk.utils.IoUtils;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URISyntaxException;
import java.net.URL;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
import java.nio.file.Paths;
import java.time.Duration;
import java.util.UUID;
```

Gere o URL.

```
/* Create a pre-signed URL to download an object in a subsequent GET request.
*/
public String createPresignedGetUrl(String bucketName, String keyName) {
    try (S3Presigner presigner = S3Presigner.create()) {
```

```

        GetObjectRequest objectRequest = GetObjectRequest.builder()
            .bucket(bucketName)
            .key(keyName)
            .build();

        GetObjectPresignRequest presignRequest =
        GetObjectPresignRequest.builder()
            .signatureDuration(Duration.ofMinutes(10)) // The URL will
            expire in 10 minutes.
            .getObjectRequest(objectRequest)
            .build();

        PresignedGetObjectRequest presignedRequest =
        presigner.presignGetObject(presignRequest);
        logger.info("Presigned URL: [{}]",
        presignedRequest.url().toString());
        logger.info("HTTP method: [{}]",
        presignedRequest.httpRequest().method());

        return presignedRequest.url().toExternalForm();
    }
}

```

Faça o download do objeto usando uma das três abordagens a seguir.

Use a classe `HttpURLConnection` do JDK (desde a v1.1) para fazer o download.

```

/* Use the JDK HttpURLConnection (since v1.1) class to do the download. */
public byte[] useHttpURLConnectionToGet(String presignedUrlString) {
    ByteArrayOutputStream byteArrayOutputStream = new
    ByteArrayOutputStream(); // Capture the response body to a byte array.

    try {
        URL presignedUrl = new URL(presignedUrlString);
        HttpURLConnection connection = (HttpURLConnection)
        presignedUrl.openConnection();
        connection.setRequestMethod("GET");
        // Download the result of executing the request.
        try (InputStream content = connection.getInputStream()) {
            IoUtils.copy(content, byteArrayOutputStream);
        }
        logger.info("HTTP response code is " + connection.getResponseCode());
    }
}

```

```
    } catch (S3Exception | IOException e) {
        logger.error(e.getMessage(), e);
    }
    return byteArrayOutputStream.toByteArray();
}
```

Use a classe `HttpClient` do JDK (desde a v11) para fazer o download.

```
/* Use the JDK HttpClient (since v11) class to do the download. */
public byte[] useHttpClientToGet(String presignedUrlString) {
    ByteArrayOutputStream byteArrayOutputStream = new
    ByteArrayOutputStream(); // Capture the response body to a byte array.

    HttpRequest.Builder requestBuilder = HttpRequest.newBuilder();
    HttpClient httpClient = HttpClient.newHttpClient();
    try {
        URL presignedUrl = new URL(presignedUrlString);
        HttpResponse<InputStream> response = httpClient.send(requestBuilder
            .uri(presignedUrl.toURI())
            .GET()
            .build(),
            HttpResponse.BodyHandlers.ofInputStream());

        IoUtils.copy(response.body(), byteArrayOutputStream);

        logger.info("HTTP response code is " + response.statusCode());
    } catch (URISyntaxException | InterruptedException | IOException e) {
        logger.error(e.getMessage(), e);
    }
    return byteArrayOutputStream.toByteArray();
}
```

Use a classe `SdkHttpClient` do AWS SDK para Java para fazer o download.

```
/* Use the AWS SDK for Java SdkHttpClient class to do the download. */
public byte[] useSdkHttpClientToPut(String presignedUrlString) {

    ByteArrayOutputStream byteArrayOutputStream = new
    ByteArrayOutputStream(); // Capture the response body to a byte array.
```



```

    try {
        URL presignedUrl = new URL(presignedUrlString);
        SdkHttpRequest request = SdkHttpRequest.builder()
            .method(SdkHttpMethod.GET)
            .uri(presignedUrl.toURI())
            .build();

        HttpExecuteRequest executeRequest = HttpExecuteRequest.builder()
            .request(request)
            .build();

        try (SdkHttpClient sdkHttpClient = ApacheHttpClient.create()) {
            HttpExecuteResponse response =
                sdkHttpClient.prepareRequest(executeRequest).call();
            response.responseBody().ifPresentOrElse(
                abortableInputStream -> {
                    try {
                        IoUtils.copy(abortableInputStream,
byteArrayOutputStream);
                    } catch (IOException e) {
                        throw new RuntimeException(e);
                    }
                },
                () -> logger.error("No response body."));

            logger.info("HTTP Response code is {}",
                response.httpResponse().statusCode());
        } catch (URISyntaxException | IOException e) {
            logger.error(e.getMessage(), e);
        }
        return byteArrayOutputStream.toByteArray();
    }

```

Gere um URL atribuído previamente para um upload e faça o upload de um arquivo (solicitação PUT).

Importações.

```

import com.example.s3.util.PresignUrlUtils;
import org.slf4j.Logger;
import software.amazon.awssdk.core.internal.sync.FileContentStreamProvider;

```

```
import software.amazon.awssdk.http.HttpExecuteRequest;
import software.amazon.awssdk.http.HttpExecuteResponse;
import software.amazon.awssdk.http.SdkHttpClient;
import software.amazon.awssdk.http.SdkHttpMethod;
import software.amazon.awssdk.http.SdkHttpRequest;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.presigner.S3Presigner;
import
    software.amazon.awssdk.services.s3.presigner.model.PresignedPutObjectRequest;
import
    software.amazon.awssdk.services.s3.presigner.model.PutObjectPresignRequest;

import java.io.File;
import java.io.IOException;
import java.io.OutputStream;
import java.io.RandomAccessFile;
import java.net.HttpURLConnection;
import java.net.URISyntaxException;
import java.net.URL;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
import java.nio.ByteBuffer;
import java.nio.channels.FileChannel;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.time.Duration;
import java.util.Map;
import java.util.UUID;
```

Gere o URL.

```
/* Create a presigned URL to use in a subsequent PUT request */
public String createPresignedUrl(String bucketName, String keyName,
    Map<String, String> metadata) {
    try (S3Presigner presigner = S3Presigner.create()) {

        PutObjectRequest objectRequest = PutObjectRequest.builder()
            .bucket(bucketName)
```

```

        .key(keyName)
        .metadata(metadata)
        .build();

        PutObjectPresignRequest presignRequest =
PutObjectPresignRequest.builder()
        .signatureDuration(Duration.ofMinutes(10)) // The URL
expires in 10 minutes.
        .putObjectRequest(objectRequest)
        .build();

        PresignedPutObjectRequest presignedRequest =
presigner.presignPutObject(presignRequest);
        String myURL = presignedRequest.url().toString();
        logger.info("Presigned URL to upload a file to: [{}]", myURL);
        logger.info("HTTP method: [{}]",
presignedRequest.httpRequest().method());

        return presignedRequest.url().toExternalForm();
    }
}

```

Faça upload de um objeto de arquivo usando uma das três abordagens a seguir.

Use a classe `URLConnection` do JDK (desde a v1.1) para fazer o upload.

```

/* Use the JDK HttpURLConnection (since v1.1) class to do the upload. */
public void useURLConnectionToPut(String presignedUrlString, File
fileToPut, Map<String, String> metadata) {
    logger.info("Begin [{}] upload", fileToPut.toString());
    try {
        URL presignedUrl = new URL(presignedUrlString);
        HttpURLConnection connection = (HttpURLConnection)
presignedUrl.openConnection();
        connection.setDoOutput(true);
        metadata.forEach((k, v) -> connection.setRequestProperty("x-amz-
meta-" + k, v));
        connection.setRequestMethod("PUT");
        OutputStream out = connection.getOutputStream();

        try (RandomAccessFile file = new RandomAccessFile(fileToPut, "r");
            FileChannel inChannel = file.getChannel()) {

```

```

8k      ByteBuffer buffer = ByteBuffer.allocate(8192); //Buffer size is

        while (inChannel.read(buffer) > 0) {
            buffer.flip();
            for (int i = 0; i < buffer.limit(); i++) {
                out.write(buffer.get());
            }
            buffer.clear();
        }
    } catch (IOException e) {
        logger.error(e.getMessage(), e);
    }

    out.close();
    connection.getResponseCode();
    logger.info("HTTP response code is " + connection.getResponseCode());

} catch (S3Exception | IOException e) {
    logger.error(e.getMessage(), e);
}
}
}

```

Use a classe `HttpClient` do JDK (desde a v11) para fazer o upload.

```

/* Use the JDK HttpClient (since v11) class to do the upload. */
public void useHttpClientToPut(String presignedUrlString, File fileToPut,
Map<String, String> metadata) {
    logger.info("Begin [{}] upload", fileToPut.toString());

    HttpRequest.Builder requestBuilder = HttpRequest.newBuilder();
    metadata.forEach((k, v) -> requestBuilder.header("x-amz-meta-" + k, v));

    HttpClient httpClient = HttpClient.newHttpClient();
    try {
        final HttpResponse<Void> response = httpClient.send(requestBuilder
            .uri(new URL(presignedUrlString).toURI())

            .PUT(HttpRequest.BodyPublishers.ofFile(Path.of(fileToPut.toURI()))
                .build(),
                HttpResponse.BodyHandlers.discarding());
    }
}

```

```
        logger.info("HTTP response code is " + response.statusCode());
    } catch (URISyntaxException | InterruptedException | IOException e) {
        logger.error(e.getMessage(), e);
    }
}
```

Use a classe `SdkHttpClient` do AWS para Java V2 para fazer o upload.

```
/* Use the AWS SDK for Java V2 SdkHttpClient class to do the upload. */
public void useSdkHttpClientToPut(String presignedUrlString, File fileToPut,
Map<String, String> metadata) {
    logger.info("Begin [{}] upload", fileToPut.toString());

    try {
        URL presignedUrl = new URL(presignedUrlString);

        SdkHttpRequest.Builder requestBuilder = SdkHttpRequest.builder()
            .method(SdkHttpMethod.PUT)
            .uri(presignedUrl.toURI());
        // Add headers
        metadata.forEach((k, v) -> requestBuilder.putHeader("x-amz-meta-" +
k, v));
        // Finish building the request.
        SdkHttpRequest request = requestBuilder.build();

        HttpExecuteRequest executeRequest = HttpExecuteRequest.builder()
            .request(request)
            .contentStreamProvider(new
FileContentStreamProvider(fileToPut.toPath()))
            .build();

        try (SdkHttpClient sdkHttpClient = ApacheHttpClient.create()) {
            HttpExecuteResponse response =
sdkHttpClient.prepareRequest(executeRequest).call();
            logger.info("Response code: {}",
response.httpResponse().statusCode());
        }
    } catch (URISyntaxException | IOException e) {
        logger.error(e.getMessage(), e);
    }
}
```

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Crie um URL pré-assinado para carregar um objeto em um bucket.

```
import https from "https";
import { PutObjectCommand, S3Client } from "@aws-sdk/client-s3";
import { fromIni } from "@aws-sdk/credential-providers";
import { HttpRequest } from "@smithy/protocol-http";
import {
  getSignedUrl,
  S3RequestPresigner,
} from "@aws-sdk/s3-request-presigner";
import { parseUrl } from "@smithy/url-parser";
import { formatUrl } from "@aws-sdk/util-format-url";
import { Hash } from "@smithy/hash-node";

const createPresignedUrlWithoutClient = async ({ region, bucket, key }) => {
  const url = parseUrl(`https://${bucket}.s3.${region}.amazonaws.com/${key}`);
  const presigner = new S3RequestPresigner({
    credentials: fromIni(),
    region,
    sha256: Hash.bind(null, "sha256"),
  });

  const signedUrlObject = await presigner.presign(
    new HttpRequest({ ...url, method: "PUT" })
  );
  return formatUrl(signedUrlObject);
};

const createPresignedUrlWithClient = ({ region, bucket, key }) => {
  const client = new S3Client({ region });
```

```
const command = new PutObjectCommand({ Bucket: bucket, Key: key });
return getSignedUrl(client, command, { expiresIn: 3600 });
};

function put(url, data) {
  return new Promise((resolve, reject) => {
    const req = https.request(
      url,
      { method: "PUT", headers: { "Content-Length": new Blob([data]).size } },
      (res) => {
        let responseBody = "";
        res.on("data", (chunk) => {
          responseBody += chunk;
        });
        res.on("end", () => {
          resolve(responseBody);
        });
      },
    );
    req.on("error", (err) => {
      reject(err);
    });
    req.write(data);
    req.end();
  });
}

export const main = async () => {
  const REGION = "us-east-1";
  const BUCKET = "example_bucket";
  const KEY = "example_file.txt";

  // There are two ways to generate a presigned URL.
  // 1. Use createPresignedUrl without the S3 client.
  // 2. Use getSignedUrl in conjunction with the S3 client and GetObjectCommand.
  try {
    const noClientUrl = await createPresignedUrlWithoutClient({
      region: REGION,
      bucket: BUCKET,
      key: KEY,
    });

    const clientUrl = await createPresignedUrlWithClient({
      region: REGION,
```

```
    bucket: BUCKET,
    key: KEY,
  });

  // After you get the presigned URL, you can provide your own file
  // data. Refer to put() above.
  console.log("Calling PUT using presigned URL without client");
  await put(noClientUrl, "Hello World");

  console.log("Calling PUT using presigned URL with client");
  await put(clientUrl, "Hello World");

  console.log("\nDone. Check your S3 console.");
} catch (err) {
  console.error(err);
}
};
```

Crie um URL pré-assinado para baixar um objeto de um bucket.

```
import { GetObjectCommand, S3Client } from "@aws-sdk/client-s3";
import { fromIni } from "@aws-sdk/credential-providers";
import { HttpRequest } from "@smithy/protocol-http";
import {
  getSignedUrl,
  S3RequestPresigner,
} from "@aws-sdk/s3-request-presigner";
import { parseUrl } from "@smithy/url-parser";
import { formatUrl } from "@aws-sdk/util-format-url";
import { Hash } from "@smithy/hash-node";

const createPresignedUrlWithoutClient = async ({ region, bucket, key }) => {
  const url = parseUrl(`https://${bucket}.s3.${region}.amazonaws.com/${key}`);
  const presigner = new S3RequestPresigner({
    credentials: fromIni(),
    region,
    sha256: Hash.bind(null, "sha256"),
  });

  const signedUrlObject = await presigner.presign(new HttpRequest(url));
  return formatUrl(signedUrlObject);
};
```



```
const createPresignedUrlWithClient = ({ region, bucket, key }) => {
  const client = new S3Client({ region });
  const command = new GetObjectCommand({ Bucket: bucket, Key: key });
  return getSignedUrl(client, command, { expiresIn: 3600 });
};

export const main = async () => {
  const REGION = "us-east-1";
  const BUCKET = "example_bucket";
  const KEY = "example_file.jpg";

  try {
    const noClientUrl = await createPresignedUrlWithoutClient({
      region: REGION,
      bucket: BUCKET,
      key: KEY,
    });

    const clientUrl = await createPresignedUrlWithClient({
      region: REGION,
      bucket: BUCKET,
      key: KEY,
    });

    console.log("Presigned URL without client");
    console.log(noClientUrl);
    console.log("\n");

    console.log("Presigned URL with client");
    console.log(clientUrl);
  } catch (err) {
    console.error(err);
  }
};
```

- Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK for JavaScript](#).

Kotlin

SDK for Kotlin

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Crie uma solicitação pré-assinada de `GetObject` e use o URL para fazer download de um objeto.

```
suspend fun getObjectPresigned(s3: S3Client, bucketName: String, keyName:
String): String {
    // Create a GetObjectRequest.
    val unsignedRequest = GetObjectRequest {
        bucket = bucketName
        key = keyName
    }

    // Presign the GetObject request.
    val presignedRequest = s3.presignGetObject(unsignedRequest, 24.hours)

    // Use the URL from the presigned HttpRequest in a subsequent HTTP GET
    request to retrieve the object.
    val objectContents = URL(presignedRequest.url.toString()).readText()

    return objectContents
}
```

Crie uma solicitação `GetObject` atribuída previamente com opções avançadas.

```
suspend fun getObjectPresignedMoreOptions(s3: S3Client, bucketName: String,
keyName: String): HttpRequest {
    // Create a GetObjectRequest.
    val unsignedRequest = GetObjectRequest {
        bucket = bucketName
        key = keyName
    }
}
```

```
// Presign the GetObject request.
val presignedRequest = s3.presignGetObject(unsignedRequest, signer =
    CrtAwsSigner) {
    signingDate = Instant.now() + 12.hours // Presigned request can be used
    12 hours from now.
    algorithm = AwsSigningAlgorithm.SIGV4_ASYMMETRIC
    signatureType = AwsSignatureType.HTTP_REQUEST_VIA_QUERY_PARAMS
    expiresAfter = 8.hours // Presigned request expires 8 hours later.
}
return presignedRequest
}
```

Crie uma solicitação pré-assinada de `PutObject` e use-a para fazer upload de um objeto.

```
suspend fun putObjectPresigned(s3: S3Client, bucketName: String, keyName: String,
    content: String) {
    // Create a PutObjectRequest.
    val unsignedRequest = PutObjectRequest {
        bucket = bucketName
        key = keyName
    }

    // Presign the request.
    val presignedRequest = s3.presignPutObject(unsignedRequest, 24.hours)

    // Use the URL and any headers from the presigned HttpRequest in a subsequent
    HTTP PUT request to retrieve the object.
    // Create a PUT request using the OkHttpClient API.
    val putRequest = Request
        .Builder()
        .url(presignedRequest.url.toString())
        .apply {
            presignedRequest.headers.forEach { key, values ->
                header(key, values.joinToString(", "))
            }
        }
        .put(content.toRequestBody())
        .build()

    val response = OkHttpClient().newCall(putRequest).execute()
    assert(response.isSuccessful)
}
```

- Para ter mais informações, consulte o [Guia do desenvolvedor do AWS SDK para Kotlin](#).

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Gere um URL pré-assinado que possa realizar uma ação do S3 por tempo limitado. Use o pacote Requests 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 para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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

# Example usage:
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__
```

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Crie solicitações pré-assinadas para objetos GET e PUT do S3.

```
async fn get_object(
    client: &Client,
    bucket: &str,
    object: &str,
    expires_in: u64,
) -> Result<(), Box<dyn Error>> {
    let expires_in = Duration::from_secs(expires_in);
    let presigned_request = client
        .get_object()
        .bucket(bucket)
        .key(object)
        .presigned(PresigningConfig::expires_in(expires_in)?)
        .await?;

    println!("Object URI: {}", presigned_request.uri());

    Ok(())
}

async fn put_object(
    client: &Client,
    bucket: &str,
    object: &str,
    expires_in: u64,
) -> Result<(), Box<dyn Error>> {
    let expires_in = Duration::from_secs(expires_in);

    let presigned_request = client
        .put_object()
        .bucket(bucket)
        .key(object)
        .presigned(PresigningConfig::expires_in(expires_in)?)
        .await?;

    println!("Object URI: {}", presigned_request.uri());

    Ok(())
}
```

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Uma página da web que oferece uma lista de objetos do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como listar objetos do Amazon S3 em uma página da web.

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

O código a seguir é o componente relevante do React que faz chamadas para o AWS SDK. Uma versão executável da aplicação que contém esse componente pode ser encontrada no link anterior do GitHub.

```
import { useEffect, useState } from "react";
import {
  ListObjectsCommand,
  ListObjectsCommandOutput,
  S3Client,
} from "@aws-sdk/client-s3";
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";
import "./App.css";

function App() {
  const [objects, setObjects] = useState<
    Required<ListObjectsCommandOutput>["Contents"]
  >([]);

  useEffect(() => {
    const client = new S3Client({
      region: "us-east-1",
      // Unless you have a public bucket, you'll need access to a private bucket.
```

```

    // One way to do this is to create an Amazon Cognito identity pool, attach
    a role to the pool,
    // and grant the role access to the 's3:GetObject' action.
    //
    // You'll also need to configure the CORS settings on the bucket to allow
    traffic from
    // this example site. Here's an example configuration that allows all
    origins. Don't
    // do this in production.
    // [
    // {
    //   "AllowedHeaders": ["*"],
    //   "AllowedMethods": ["GET"],
    //   "AllowedOrigins": ["*"],
    //   "ExposeHeaders": [],
    // },
    // ]
    //
    credentials: fromCognitoIdentityPool({
      clientConfig: { region: "us-east-1" },
      identityPoolId: "<YOUR_IDENTITY_POOL_ID>",
    }),
  });
  const command = new ListObjectsCommand({ Bucket: "bucket-name" });
  client.send(command).then(({ Contents }) => setObjects(Contents || []));
}, []);

return (
  <div className="App">
    {objects.map((o) => (
      <div key={o.ETag}>{o.Key}</div>
    ))}
  </div>
);
}

export default App;

```

- Para obter detalhes da API, consulte [ListObjects](#) na Referência da API AWS SDK for JavaScript.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Fazer download de todos os objetos de um bucket do Amazon Simple Storage Service (Amazon S3) em um diretório local

O exemplo de código a seguir mostra como fazer download de todos os objetos de um bucket do Amazon Simple Storage Service (Amazon S3) em um diretório local.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Use [S3TransferManager](#) para [fazer download de todos os objetos do S3](#) no mesmo bucket do S3. Veja o [arquivo completo](#) e [teste](#).

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.services.s3.model.ObjectIdentifier;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedDirectoryDownload;
import software.amazon.awssdk.transfer.s3.model.DirectoryDownload;
import software.amazon.awssdk.transfer.s3.model.DownloadDirectoryRequest;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.HashSet;
import java.util.Set;
import java.util.UUID;
import java.util.stream.Collectors;
```

```
public Integer downloadObjectsToDirectory(S3TransferManager transferManager,
    URI destinationPathURI, String bucketName) {
    DirectoryDownload directoryDownload =
transferManager.downloadDirectory(DownloadDirectoryRequest.builder()
    .destination(Paths.get(destinationPathURI))
    .bucket(bucketName)
    .build());
    CompletedDirectoryDownload completedDirectoryDownload =
directoryDownload.completionFuture().join();

    completedDirectoryDownload.failedTransfers()
        .forEach(fail -> logger.warn("Object [{}] failed to transfer",
fail.toString()));
    return completedDirectoryDownload.failedTransfers().size();
}
```

- Para obter detalhes da API, consulte [DownloadDirectory](#) na Referência da API do AWS SDK for Java 2.x.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Obter um objeto do Amazon S3 de um ponto de acesso multirregional usando um AWS SDK

O exemplo de código a seguir demonstra como obter um objeto de um ponto de acesso multirregional.

Kotlin

SDK for Kotlin

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Configure o cliente do S3 para usar o algoritmo de assinatura Asymmetric Sigv4 (Sigv4a).

```
suspend fun createS3Client(): S3Client {
    // Configure your S3Client to use the Asymmetric Sigv4 (Sigv4a)
    signing algorithm.
    val sigV4AScheme = SigV4AsymmetricAuthScheme(CrtAwsSigner)
    val s3 = S3Client.fromEnvironment {
        authSchemes = listOf(sigV4AScheme)
    }
    return s3
}
```

Use o ARN do ponto de acesso multirregional em vez de um nome de bucket para recuperar o objeto.

```
suspend fun getObjectFromMrap(s3: S3Client, mrapArn: String, keyName:
String): String? {
    val request = GetObjectRequest {
        bucket = mrapArn // Use the ARN instead of the bucket name for object
        operations.
        key = keyName
    }

    var stringObj: String? = null
    s3.getObject(request) { resp ->
        stringObj = resp.body?.decodeToString()
        if (stringObj != null) {
            println("Successfully read $keyName from $mrapArn")
        }
    }
    return stringObj
}
```

- Para ter mais informações, consulte o [Guia do desenvolvedor do AWS SDK para Kotlin](#).
- Para obter detalhes da API, consulte [GetObject](#) na Referência da API AWS SDK for Kotlin.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). 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 e especificando um cabeçalho If-Modified-Since

O exemplo de código a seguir mostra como ler dados de um objeto em um bucket do S3, mas somente se esse bucket não tiver sido modificado desde a última recuperação.

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
use aws_sdk_s3::{
    error::SdkError,
    operation::head_object::HeadObjectError,
    primitives::{ByteStream, DateTime, DateTimeFormat},
    Client, Error,
};
use tracing::{error, warn};

const KEY: &str = "key";
const BODY: &str = "Hello, world!";

/// Demonstrate how `if-modified-since` reports that matching objects haven't
/// changed.
///
/// # Steps
/// - Create a bucket.
/// - Put an object in the bucket.
/// - Get the bucket headers.
/// - Get the bucket headers again but only if modified.
/// - Delete the bucket.
#[tokio::main]
async fn main() -> Result<(), Error> {
    tracing_subscriber::fmt::init();

    // Get a new UUID to use when creating a unique bucket name.
```

```
let uuid = uuid::Uuid::new_v4();

// Load the AWS configuration from the environment.
let client = Client::new(&aws_config::load_from_env().await);

// Generate a unique bucket name using the previously generated UUID.
// Then create a new bucket with that name.
let bucket_name = format!("if-modified-since-{{uuid}}");
client
    .create_bucket()
    .bucket(bucket_name.clone())
    .send()
    .await?;

// Create a new object in the bucket whose name is `KEY` and whose
// contents are `BODY`.
let put_object_output = client
    .put_object()
    .bucket(bucket_name.as_str())
    .key(KEY)
    .body(ByteStream::from_static(BODY.as_bytes()))
    .send()
    .await;

// If the `PutObject` succeeded, get the eTag string from it. Otherwise,
// report an error and return an empty string.
let e_tag_1 = match put_object_output {
    Ok(put_object) => put_object.e_tag.unwrap(),
    Err(err) => {
        error!("{err:?}");
        String::new()
    }
};

// Request the object's headers.
let head_object_output = client
    .head_object()
    .bucket(bucket_name.as_str())
    .key(KEY)
    .send()
    .await;

// If the `HeadObject` request succeeded, create a tuple containing the
// values of the headers `last-modified` and `etag`. If the request
```



```
// failed, return the error in a tuple instead.
let (last_modified, e_tag_2) = match head_object_output {
    Ok(head_object) => (
        Ok(head_object.last_modified().cloned().unwrap()),
        head_object.e_tag.unwrap(),
    ),
    Err(err) => (Err(err), String::new()),
};

warn!("last modified: {last_modified:?}");
assert_eq!(
    e_tag_1, e_tag_2,
    "PutObject and first GetObject had differing eTags"
);

println!("First value of last_modified: {last_modified:?}");
println!("First tag: {}\n", e_tag_1);

// Send a second `HeadObject` request. This time, the `if_modified_since`
// option is specified, giving the `last_modified` value returned by the
// first call to `HeadObject`.
//
// Since the object hasn't been changed, and there are no other objects in
// the bucket, there should be no matching objects.

let head_object_output = client
    .head_object()
    .bucket(bucket_name.as_str())
    .key(KEY)
    .if_modified_since(last_modified.unwrap())
    .send()
    .await;

// If the `HeadObject` request succeeded, the result is a tuple containing
// the `last_modified` and `e_tag_1` properties. This is not the expected
// result.
//
// The expected result of the second call to `HeadObject` is an
// `SdkError::ServiceError` containing the HTTP error response. If that's
// the case and the HTTP status is 304 (not modified), the output is a
// tuple containing the values of the HTTP `last-modified` and `etag`
// headers.
//
// If any other HTTP error occurred, the error is returned as an
```

```

// `SdkError::ServiceError`.

let (last_modified, e_tag_2): (Result<DateTime, SdkError<HeadObjectError>>,
String) =
  match head_object_output {
    Ok(head_object) => (
      Ok(head_object.last_modified().cloned().unwrap()),
      head_object.e_tag.unwrap(),
    ),
    Err(err) => match err {
      SdkError::ServiceError(err) => {
        // Get the raw HTTP response. If its status is 304, the
        // object has not changed. This is the expected code path.
        let http = err.raw();
        match http.status().as_u16() {
          // If the HTTP status is 304: Not Modified, return a
          // tuple containing the values of the HTTP
          // `last-modified` and `etag` headers.
          304 => (
            Ok(DateTime::from_str(
              http.headers().get("last-modified").unwrap(),
              DateTimeFormat::HttpDate,
            )
              .unwrap()),
            http.headers().get("etag").map(|t|
t.into()).unwrap(),
          ),
          // Any other HTTP status code is returned as an
          // `SdkError::ServiceError`.
          _ => (Err(SdkError::ServiceError(err)), String::new()),
        }
      }
      // Any other kind of error is returned in a tuple containing the
      // error and an empty string.
      _ => (Err(err), String::new()),
    },
  };

warn!("last modified: {last_modified:?}");
assert_eq!(
  e_tag_1, e_tag_2,
  "PutObject and second HeadObject had different eTags"
);

```

```
println!("Second value of last modified: {last_modified:?}");
println!("Second tag: {}", e_tag_2);

// Clean up by deleting the object and the bucket.
client
    .delete_object()
    .bucket(bucket_name.as_str())
    .key(KEY)
    .send()
    .await?;

client
    .delete_bucket()
    .bucket(bucket_name.as_str())
    .send()
    .await?;

Ok(())
}
```

- Para obter detalhes da API, consulte [GetObject](#) na Referência da API AWS SDK para Rust.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Conceitos básicos de buckets e objetos do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como:

- Criar um bucket e fazer upload de um arquivo para ele.
- Baixar um objeto de um bucket.
- Copiar um objeto em uma subpasta em um bucket.
- Listar os objetos em um bucket.
- Excluir os objetos do bucket e o bucket.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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;

        var sepBar = new string('-', Console.WindowWidth);

        Console.WriteLine(sepBar);
        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(sepBar);

        // Create a bucket.
        Console.WriteLine($"{sepBar}");
        Console.WriteLine("\nCreate a new Amazon S3 bucket.\n");
        Console.WriteLine(sepBar);
    }
}
```

```
Console.WriteLine("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(sepBar);
Console.WriteLine("Upload a file to the new bucket.");
Console.WriteLine(sepBar);

// Get the local path and filename for the file to upload.
while (string.IsNullOrEmpty(filePath))
{
    Console.WriteLine("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.Write("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 AWS SDK for .NET.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)

- [ListObjectsV2](#)
- [PutObject](#)

Bash

AWS CLI com script Bash

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
#####
# function s3_getting_started
#
# This function creates, copies, and deletes S3 buckets and objects.
#
# Returns:
#     0 - If successful.
#     1 - If an error occurred.
#####
function s3_getting_started() {
    {
        if [ "$BUCKET_OPERATIONS_SOURCED" != "True" ]; then
            cd bucket-lifecycle-operations || exit

            source ./bucket_operations.sh
            cd ..
        fi
    }

    echo_repeat "*" 88
    echo "Welcome to the Amazon S3 getting started demo."
    echo_repeat "*" 88

    local bucket_name
    bucket_name=$(generate_random_name "doc-example-bucket")

    local region_code
    region_code=$(aws configure get region)
```



```
if create_bucket -b "$bucket_name" -r "$region_code"; then
    echo "Created demo bucket named $bucket_name"
else
    errecho "The bucket failed to create. This demo will exit."
    return 1
fi

local file_name
while [ -z "$file_name" ]; do
    echo -n "Enter a file you want to upload to your bucket: "
    get_input
    file_name=$get_input_result

    if [ ! -f "$file_name" ]; then
        echo "Could not find file $file_name. Are you sure it exists?"
        file_name=""
    fi
done

local key
key="$(basename "$file_name")"

local result=0
if copy_file_to_bucket "$bucket_name" "$file_name" "$key"; then
    echo "Uploaded file $file_name into bucket $bucket_name with key $key."
else
    result=1
fi

local destination_file
destination_file="$file_name.download"
if yes_no_input "Would you like to download $key to the file $destination_file?
(y/n) "; then
    if download_object_from_bucket "$bucket_name" "$destination_file" "$key";
then
        echo "Downloaded $key in the bucket $bucket_name to the file
$destination_file."
    else
        result=1
    fi
fi
```

```
if yes_no_input "Would you like to copy $key a new object key in your bucket?
(y/n) "; then
    local to_key
    to_key="demo/$key"
    if copy_item_in_bucket "$bucket_name" "$key" "$to_key"; then
        echo "Copied $key in the bucket $bucket_name to the $to_key."
    else
        result=1
    fi
fi

local bucket_items
bucket_items=$(list_items_in_bucket "$bucket_name")

# shellcheck disable=SC2181
if [[ $? -ne 0 ]]; then
    result=1
fi

echo "Your bucket contains the following items."
echo -e "Name\t\tSize"
echo "$bucket_items"

if yes_no_input "Delete the bucket, $bucket_name, as well as the objects in it?
(y/n) "; then
    bucket_items=$(echo "$bucket_items" | cut -f 1)

    if delete_items_in_bucket "$bucket_name" "$bucket_items"; then
        echo "The following items were deleted from the bucket $bucket_name"
        echo "$bucket_items"
    else
        result=1
    fi

    if delete_bucket "$bucket_name"; then
        echo "Deleted the bucket $bucket_name"
    else
        result=1
    fi
fi

return $result
}
```

As funções do Amazon S3 usadas nesse cenário.

```
#####
# function create-bucket
#
# This function creates the specified bucket in the specified AWS Region, unless
# it already exists.
#
# Parameters:
#     -b bucket_name  -- The name of the bucket to create.
#     -r region_code  -- The code for an AWS Region in which to
#                       create the bucket.
#
# Returns:
#     The URL of the bucket that was created.
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
function create_bucket() {
    local bucket_name region_code response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function create_bucket"
        echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
        echo "  -b bucket_name    The name of the bucket. It must be globally
unique."
        echo "  [-r region_code]  The code for an AWS Region in which the bucket is
created."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "b:r:h" option; do
        case "${option}" in
            b) bucket_name="${OPTARG}" ;;
            r) region_code="${OPTARG}" ;;
            h)
                usage
            ;;
        esac
    done
```

```
        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done

if [[ -z "$bucket_name" ]]; then
    errecho "ERROR: You must provide a bucket name with the -b parameter."
    usage
    return 1
fi

local bucket_config_arg
# A location constraint for "us-east-1" returns an error.
if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then
    bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
fi

iecho "Parameters:\n"
iecho "    Bucket name:  $bucket_name"
iecho "    Region code:  $region_code"
iecho ""

# If the bucket already exists, we don't want to try to create it.
if (bucket_exists "$bucket_name"); then
    errecho "ERROR: A bucket with that name already exists. Try again."
    return 1
fi

# shellcheck disable=SC2086
response=$(aws s3api create-bucket \
    --bucket "$bucket_name" \
    $bucket_config_arg)

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
fi
```

```
}

#####
# function copy_file_to_bucket
#
# This function creates a file in the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket to copy the file to.
#     $2 - The path and file name of the local file to copy to the bucket.
#     $3 - The key (name) to call the copy of the file in the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function copy_file_to_bucket() {
    local response bucket_name source_file destination_file_name
    bucket_name=$1
    source_file=$2
    destination_file_name=$3

    response=$(aws s3api put-object \
        --bucket "$bucket_name" \
        --body "$source_file" \
        --key "$destination_file_name")

    # shellcheck disable=SC2181
    if [[ ${?} -ne 0 ]]; then
        errecho "ERROR: AWS reports put-object operation failed.\n$response"
        return 1
    fi
}

#####
# function download_object_from_bucket
#
# This function downloads an object in a bucket to a file.
#
# Parameters:
#     $1 - The name of the bucket to download the object from.
#     $2 - The path and file name to store the downloaded bucket.
#     $3 - The key (name) of the object in the bucket.
#
```

```

# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function download_object_from_bucket() {
    local bucket_name=$1
    local destination_file_name=$2
    local object_name=$3
    local response

    response=$(aws s3api get-object \
        --bucket "$bucket_name" \
        --key "$object_name" \
        "$destination_file_name")

    # shellcheck disable=SC2181
    if [[ ${?} -ne 0 ]]; then
        errecho "ERROR: AWS reports put-object operation failed.\n$response"
        return 1
    fi
}

#####
# function copy_item_in_bucket
#
# This function creates a copy of the specified file in the same bucket.
#
# Parameters:
#     $1 - The name of the bucket to copy the file from and to.
#     $2 - The key of the source file to copy.
#     $3 - The key of the destination file.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function copy_item_in_bucket() {
    local bucket_name=$1
    local source_key=$2
    local destination_key=$3
    local response

    response=$(aws s3api copy-object \
        --bucket "$bucket_name" \

```

```

--copy-source "$bucket_name/$source_key" \
--key "$destination_key")

# shellcheck disable=SC2181
if [[ $? -ne 0 ]]; then
    errecho "ERROR: AWS reports s3api copy-object operation failed.\n$response"
    return 1
fi
}

#####
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and
# size fields from the Contents collection.
#
# Parameters:
#     $1 - The name of the bucket.
#
# Returns:
#     The list of files in text format.
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
function list_items_in_bucket() {
    local bucket_name=$1
    local response

    response=$(aws s3api list-objects \
        --bucket "$bucket_name" \
        --output text \
        --query 'Contents[].{Key: Key, Size: Size}')

    # shellcheck disable=SC2181
    if [[ ${?} -eq 0 ]]; then
        echo "$response"
    else
        errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
        return 1
    fi
}

```

```
#####
# function delete_items_in_bucket
#
# This function deletes the specified list of keys from the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.
#     $2 - A list of keys in the bucket to delete.

# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_items_in_bucket() {
    local bucket_name=$1
    local keys=$2
    local response

    # Create the JSON for the items to delete.
    local delete_items
    delete_items="{\"Objects\":["
    for key in $keys; do
        delete_items="$delete_items{\"Key\": \"$key\"},"
    done
    delete_items=${delete_items%?} # Remove the final comma.
    delete_items="$delete_items]"

    response=$(aws s3api delete-objects \
        --bucket "$bucket_name" \
        --delete "$delete_items")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        errecho "ERROR: AWS reports s3api delete-object operation failed.\n
$response"
        return 1
    fi
}

#####
# function delete_bucket
#
# This function deletes the specified bucket.
#
```



```
# Parameters:
#     $1 - The name of the bucket.

# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_bucket() {
    local bucket_name=$1
    local response


    response=$(aws s3api delete-bucket \
        --bucket "$bucket_name")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        errecho "ERROR: AWS reports s3api delete-bucket failed.\n$response"
        return 1
    fi
}
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência de comandos da AWS CLI.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

C++

SDK para C++

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
#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/ListObjectsV2Request.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.
        */
        static bool
        DeleteBucket(const Aws::String &bucketName, Aws::S3::S3Client &client);

        //! 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.
    */
    static bool
    DeleteObjectFromBucket(const Aws::String &bucketName, const Aws::String
&key,
                        Aws::S3::S3Client &client);
    }
}

//! Scenario to create, copy, and delete S3 buckets and objects.
/*!
    \sa S3_GettingStartedScenario()
    \param uploadFilePath Path to file to upload to an Amazon S3 bucket.
    \param saveFilePath Path for saving a downloaded S3 object.
    \param clientConfig Aws client configuration.
*/
bool AwsDoc::S3::S3_GettingStartedScenario(const Aws::String &uploadFilePath,
                                           const Aws::String &saveFilePath,
                                           const Aws::Client::ClientConfiguration
&clientConfig) {

    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 {
        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::cerr << "Error: unable to open file, '" << uploadFilePath <<
            "'." << std::endl;
        AwsDoc::S3::DeleteBucket(bucketName, client);
        return false;
    }

    request.SetBody(input_data);

    Aws::S3::Model::PutObjectOutcome outcome =

```

```

        client.PutObject(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error: PutObject: " <<
            outcome.GetError().GetMessage() << std::endl;
        AwsDoc::S3::DeleteObjectFromBucket(bucketName, key, client);
        AwsDoc::S3::DeleteBucket(bucketName, client);
        return false;
    }
    else {
        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::cerr << "Error: GetObject: " <<
            err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    }
    else {
        std::cout << "Downloaded the object with the key, '" << key
            << "', in the bucket, '"
            << bucketName << "'." << std::endl;

        Aws::IOStream &ioStream = outcome.GetResultWithOwnership().
            GetBody();
        Aws::OFStream outputStream(saveFilePath,
            std::ios_base::out | std::ios_base::binary);
        if (!outStream.is_open()) {
            std::cout << "Error: unable to open file, '" << saveFilePath <<
"" << std::endl;

```

```
    }
    else {
        outputStream << inputStream.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::cerr << "Error: CopyObject: " <<
            outcome.GetError().GetMessage() << std::endl;
    }
    else {
        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::ListObjectsV2Request request;
    request.WithBucket(bucketName);

    Aws::String continuationToken;
    Aws::Vector<Aws::S3::Model::Object> allObjects;

    do {
        if (!continuationToken.empty()) {
            request.SetContinuationToken(continuationToken);
        }
        Aws::S3::Model::ListObjectsV2Outcome outcome = client.ListObjectsV2(
            request);
```

```

        if (!outcome.IsSuccess()) {
            std::cerr << "Error: ListObjects: " <<
                outcome.GetError().GetMessage() << std::endl;
            break;
        }
        else {
            Aws::Vector<Aws::S3::Model::Object> objects =
                outcome.GetResult().GetContents();
            allObjects.insert(allObjects.end(), objects.begin(),
objects.end());
            continuationToken = outcome.GetResult().GetContinuationToken();
        }
    } while (!continuationToken.empty());

    std::cout << allObjects.size() << " objects in the bucket, " <<
bucketName
        << ":" << std::endl;

    for (Aws::S3::Model::Object &object: allObjects) {
        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);
AwsDoc::S3::DeleteObjectFromBucket(bucketName, key, client);

// 7. Delete the bucket.
return AwsDoc::S3::DeleteBucket(bucketName, client);
}

bool AwsDoc::S3::DeleteObjectFromBucket(const Aws::String &bucketName,
                                        const Aws::String &key,
                                        Aws::S3::S3Client &client) {
    Aws::S3::Model::DeleteObjectRequest request;
    request.SetBucket(bucketName);
    request.SetKey(key);

    Aws::S3::Model::DeleteObjectOutcome outcome =
        client.DeleteObject(request);

    if (!outcome.IsSuccess()) {

```

```

        std::cerr << "Error: DeleteObject: " <<
            outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Deleted the object with the key, '" << key
            << "', from the bucket, '"
            << bucketName << "'.'" << std::endl;
    }

    return outcome.IsSuccess();
}

bool
AwsDoc::S3::DeleteBucket(const Aws::String &bucketName, Aws::S3::S3Client
&client) {
    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;
    }
    else {
        std::cout << "Deleted the bucket, '" << bucketName << "'.'" << std::endl;
    }
    return outcome.IsSuccess();
}

```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for C++.

- [CopyObject](#)
- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteObjects](#)
- [GetObject](#)

- [ListObjectsV2](#)
- [PutObject](#)

Go

SDK para Go V2

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Defina um struct que envolva as ações de bucket e objeto usadas pelo cenário.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
// actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// ListBuckets lists the buckets in the current account.
func (basics BucketBasics) ListBuckets() ([]types.Bucket, error) {
    result, err := basics.S3Client.ListBuckets(context.TODO(),
        &s3.ListBucketsInput{})
    var buckets []types.Bucket
    if err != nil {
        log.Printf("Couldn't list buckets for your account. Here's why: %v\n", err)
    } else {
        buckets = result.Buckets
    }
    return buckets, err
}
```

```
// BucketExists checks whether a bucket exists in the current account.
func (basics BucketBasics) BucketExists(bucketName string) (bool, error) {
    _, err := basics.S3Client.HeadBucket(context.TODO(), &s3.HeadBucketInput{
        Bucket: aws.String(bucketName),
    })
    exists := true
    if err != nil {
        var apiError smithy.APIError
        if errors.As(err, &apiError) {
            switch apiError.(type) {
            case *types.NotFound:
                log.Printf("Bucket %v is available.\n", bucketName)
                exists = false
                err = nil
            default:
                log.Printf("Either you don't have access to bucket %v or another error
occurred. "+
                    "Here's what happened: %v\n", bucketName, err)
            }
        }
    } else {
        log.Printf("Bucket %v exists and you already own it.", bucketName)
    }

    return exists, err
}

// CreateBucket creates a bucket with the specified name in the specified Region.
func (basics BucketBasics) CreateBucket(name string, region string) error {
    _, err := basics.S3Client.CreateBucket(context.TODO(), &s3.CreateBucketInput{
        Bucket: aws.String(name),
        CreateBucketConfiguration: &types.CreateBucketConfiguration{
            LocationConstraint: types.BucketLocationConstraint(region),
        },
    })
    if err != nil {
        log.Printf("Couldn't create bucket %v in Region %v. Here's why: %v\n",
            name, region, err)
    }

    return err
}
```

```
}

// UploadFile reads from a file and puts the data into an object in a bucket.
func (basics BucketBasics) UploadFile(bucketName string, objectKey string,
    fileName string) error {
    file, err := os.Open(fileName)
    if err != nil {
        log.Printf("Couldn't open file %v to upload. Here's why: %v\n", fileName, err)
    } else {
        defer file.Close()
        _, err = basics.S3Client.PutObject(context.TODO(), &s3.PutObjectInput{
            Bucket: aws.String(bucketName),
            Key:     aws.String(objectKey),
            Body:    file,
        })
        if err != nil {
            log.Printf("Couldn't upload file %v to %v:%v. Here's why: %v\n",
                fileName, bucketName, objectKey, err)
        }
    }
    return err
}

// UploadLargeObject uses an upload manager to upload data to an object in a
// bucket.
// The upload manager breaks large data into parts and uploads the parts
// concurrently.
func (basics BucketBasics) UploadLargeObject(bucketName string, objectKey string,
    largeObject []byte) error {
    largeBuffer := bytes.NewReader(largeObject)
    var partMiBs int64 = 10
    uploader := manager.NewUploader(basics.S3Client, func(u *manager.Uploader) {
        u.PartSize = partMiBs * 1024 * 1024
    })
    _, err := uploader.Upload(context.TODO(), &s3.PutObjectInput{
        Bucket: aws.String(bucketName),
        Key:     aws.String(objectKey),
        Body:    largeBuffer,
    })
    if err != nil {
```

```
    log.Printf("Couldn't upload large object to %v:%v. Here's why: %v\n",
        bucketName, objectKey, err)
}

return err
}

// DownloadFile gets an object from a bucket and stores it in a local file.
func (basics BucketBasics) DownloadFile(bucketName string, objectKey string,
    fileName string) error {
    result, err := basics.S3Client.GetObject(context.TODO(), &s3.GetObjectInput{
        Bucket: aws.String(bucketName),
        Key:    aws.String(objectKey),
    })
    if err != nil {
        log.Printf("Couldn't get object %v:%v. Here's why: %v\n", bucketName,
            objectKey, err)
        return err
    }
    defer result.Body.Close()
    file, err := os.Create(fileName)
    if err != nil {
        log.Printf("Couldn't create file %v. Here's why: %v\n", fileName, err)
        return err
    }
    defer file.Close()
    body, err := io.ReadAll(result.Body)
    if err != nil {
        log.Printf("Couldn't read object body from %v. Here's why: %v\n", objectKey,
            err)
    }
    _, err = file.Write(body)
    return err
}

// DownloadLargeObject uses a download manager to download an object from a
// bucket.
// The download manager gets the data in parts and writes them to a buffer until
// all of
// the data has been downloaded.
```

```
func (basics BucketBasics) DownloadLargeObject(bucketName string, objectKey
string) ([]byte, error) {
    var partMiBs int64 = 10
    downloader := manager.NewDownloader(basics.S3Client, func(d *manager.Downloader)
    {
        d.PartSize = partMiBs * 1024 * 1024
    })
    buffer := manager.NewWriteAtBuffer([]byte{})
    _, err := downloader.Download(context.TODO(), buffer, &s3.GetObjectInput{
        Bucket: aws.String(bucketName),
        Key:     aws.String(objectKey),
    })
    if err != nil {
        log.Printf("Couldn't download large object from %v:%v. Here's why: %v\n",
            bucketName, objectKey, err)
    }
    return buffer.Bytes(), err
}

// CopyToFolder copies an object in a bucket to a subfolder in the same bucket.
func (basics BucketBasics) CopyToFolder(bucketName string, objectKey string,
folderName string) error {
    _, err := basics.S3Client.CopyObject(context.TODO(), &s3.CopyObjectInput{
        Bucket:     aws.String(bucketName),
        CopySource: aws.String(fmt.Sprintf("%v/%v", bucketName, objectKey)),
        Key:        aws.String(fmt.Sprintf("%v/%v", folderName, objectKey)),
    })
    if err != nil {
        log.Printf("Couldn't copy object from %v:%v to %v:%v/%v. Here's why: %v\n",
            bucketName, objectKey, bucketName, folderName, objectKey, err)
    }
    return err
}

// CopyToBucket copies an object in a bucket to another bucket.
func (basics BucketBasics) CopyToBucket(sourceBucket string, destinationBucket
string, objectKey string) error {
    _, err := basics.S3Client.CopyObject(context.TODO(), &s3.CopyObjectInput{
        Bucket:     aws.String(destinationBucket),
        CopySource: aws.String(fmt.Sprintf("%v/%v", sourceBucket, objectKey)),
    })
}
```

```
    Key:      aws.String(objectKey),
  })
  if err != nil {
    log.Printf("Couldn't copy object from %v:%v to %v:%v. Here's why: %v\n",
      sourceBucket, objectKey, destinationBucket, objectKey, err)
  }
  return err
}

// ListObjects lists the objects in a bucket.
func (basics BucketBasics) ListObjects(bucketName string) ([]types.Object, error)
{
  result, err := basics.S3Client.ListObjectsV2(context.TODO(),
    &s3.ListObjectsV2Input{
      Bucket: aws.String(bucketName),
    })
  var contents []types.Object
  if err != nil {
    log.Printf("Couldn't list objects in bucket %v. Here's why: %v\n", bucketName,
      err)
  } else {
    contents = result.Contents
  }
  return contents, err
}

// DeleteObjects deletes a list of objects from a bucket.
func (basics BucketBasics) DeleteObjects(bucketName string, objectKeys []string)
error {
  var objectIds []types.ObjectIdentifier
  for _, key := range objectKeys {
    objectIds = append(objectIds, types.ObjectIdentifier{Key: aws.String(key)})
  }
  output, err := basics.S3Client.DeleteObjects(context.TODO(),
    &s3.DeleteObjectsInput{
      Bucket: aws.String(bucketName),
      Delete: &types.Delete{Objects: objectIds},
    })
  if err != nil {
```

```
    log.Printf("Couldn't delete objects from bucket %v. Here's why: %v\n",
bucketName, err)
} else {
    log.Printf("Deleted %v objects.\n", len(output.Deleted))
}
return err
}

// DeleteBucket deletes a bucket. The bucket must be empty or an error is
returned.
func (basics BucketBasics) DeleteBucket(bucketName string) error {
    _, err := basics.S3Client.DeleteBucket(context.TODO(), &s3.DeleteBucketInput{
        Bucket: aws.String(bucketName)})
    if err != nil {
        log.Printf("Couldn't delete bucket %v. Here's why: %v\n", bucketName, err)
    }
    return err
}
```

Execute um cenário interativo que mostre como trabalhar com buckets e objetos do S3.

```
// RunGetStartedScenario is an interactive example that shows you how to use
Amazon
// Simple Storage Service (Amazon S3) to create an S3 bucket and use it to store
objects.
//
// 1. Create a bucket.
// 2. Upload a local file to the bucket.
// 3. Upload a large object to the bucket by using an upload manager.
// 4. Download an object to a local file.
// 5. Download a large object by using a download manager.
// 6. Copy an object to a different folder in the bucket.
// 7. List objects in the bucket.
// 8. Delete all objects in the bucket.
// 9. Delete the bucket.
//
// This example creates an Amazon S3 service client from the specified sdkConfig
so that
```

```
// you can replace it with a mocked or stubbed config for unit testing.
//
// It uses a questioner from the `demotools` package to get input during the
// example.
// This package can be found in the ..\..\demotools folder of this repo.
func RunGetStartedScenario(sdkConfig aws.Config, questioner
demotools.IQuestioner) {
defer func() {
if r := recover(); r != nil {
fmt.Println("Something went wrong with the demo.\n", r)
}
}()

log.Println(strings.Repeat("-", 88))
log.Println("Welcome to the Amazon S3 getting started demo.")
log.Println(strings.Repeat("-", 88))

s3Client := s3.NewFromConfig(sdkConfig)
bucketBasics := actions.BucketBasics{S3Client: s3Client}

count := 10
log.Printf("Let's list up to %v buckets for your account:", count)
buckets, err := bucketBasics.ListBuckets()
if err != nil {
panic(err)
}
if len(buckets) == 0 {
log.Println("You don't have any buckets!")
} else {
if count > len(buckets) {
count = len(buckets)
}
for _, bucket := range buckets[:count] {
log.Printf("\t\t%v\n", *bucket.Name)
}
}

bucketName := questioner.Ask("Let's create a bucket. Enter a name for your
bucket:",
demotools.NotEmpty{})
bucketExists, err := bucketBasics.BucketExists(bucketName)
if err != nil {
panic(err)
}
}
```



```
if !bucketExists {
    err = bucketBasics.CreateBucket(bucketName, sdkConfig.Region)
    if err != nil {
        panic(err)
    } else {
        log.Println("Bucket created.")
    }
}
log.Println(strings.Repeat("-", 88))

fmt.Println("Let's upload a file to your bucket.")
smallFile := questioner.Ask("Enter the path to a file you want to upload:",
    demotools.NotEmpty{})
const smallKey = "doc-example-key"
err = bucketBasics.UploadFile(bucketName, smallKey, smallFile)
if err != nil {
    panic(err)
}
log.Printf("Uploaded %v as %v.\n", smallFile, smallKey)
log.Println(strings.Repeat("-", 88))

mibs := 30
log.Printf("Let's create a slice of %v MiB of random bytes and upload it to your
bucket. ", mibs)
questioner.Ask("Press Enter when you're ready.")
largeBytes := make([]byte, 1024*1024*mibs)
rand.Seed(time.Now().Unix())
rand.Read(largeBytes)
largeKey := "doc-example-large"
log.Println("Uploading...")
err = bucketBasics.UploadLargeObject(bucketName, largeKey, largeBytes)
if err != nil {
    panic(err)
}
log.Printf("Uploaded %v MiB object as %v", mibs, largeKey)
log.Println(strings.Repeat("-", 88))

log.Printf("Let's download %v to a file.", smallKey)
downloadFileName := questioner.Ask("Enter a name for the downloaded file:",
    demotools.NotEmpty{})
err = bucketBasics.DownloadFile(bucketName, smallKey, downloadFileName)
if err != nil {
    panic(err)
}
```

```
log.Printf("File %v downloaded.", downloadFileName)
log.Println(strings.Repeat("-", 88))

log.Printf("Let's download the %v MiB object.", mibs)
questioner.Ask("Press Enter when you're ready.")
log.Println("Downloading...")
largeDownload, err := bucketBasics.DownloadLargeObject(bucketName, largeKey)
if err != nil {
    panic(err)
}
log.Printf("Downloaded %v bytes.", len(largeDownload))
log.Println(strings.Repeat("-", 88))

log.Printf("Let's copy %v to a folder in the same bucket.", smallKey)
folderName := questioner.Ask("Enter a folder name: ", demotools.NotEmpty{})
err = bucketBasics.CopyToFolder(bucketName, smallKey, folderName)
if err != nil {
    panic(err)
}
log.Printf("Copied %v to %v/%v.\n", smallKey, folderName, smallKey)
log.Println(strings.Repeat("-", 88))

log.Println("Let's list the objects in your bucket.")
questioner.Ask("Press Enter when you're ready.")
objects, err := bucketBasics.ListObjects(bucketName)
if err != nil {
    panic(err)
}
log.Printf("Found %v objects.\n", len(objects))
var objKeys []string
for _, object := range objects {
    objKeys = append(objKeys, *object.Key)
    log.Printf("\t%v\n", *object.Key)
}
log.Println(strings.Repeat("-", 88))

if questioner.AskBool("Do you want to delete your bucket and all of its "+
    "contents? (y/n)", "y") {
    log.Println("Deleting objects.")
    err = bucketBasics.DeleteObjects(bucketName, objKeys)
    if err != nil {
        panic(err)
    }
}
log.Println("Deleting bucket.")
```

```
err = bucketBasics.DeleteBucket(bucketName)
if err != nil {
    panic(err)
}
log.Printf("Deleting downloaded file %v.\n", downloadFileName)
err = os.Remove(downloadFileName)
if err != nil {
    panic(err)
}
} else {
    log.Println("Okay. Don't forget to delete objects from your bucket to avoid
charges.")
}
log.Println(strings.Repeat("-", 88))

log.Println("Thanks for watching!")
log.Println(strings.Repeat("-", 88))
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for Go.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
/**
 * 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
 *
 * This Java code example performs the following tasks:
 *
 * 1. Creates an Amazon S3 bucket.
 * 2. Uploads an object to the bucket.
 * 3. Downloads the object to another local file.
 * 4. Uploads an object using multipart upload.
 * 5. List all objects located in the Amazon S3 bucket.
 * 6. Copies the object to another Amazon S3 bucket.
 * 7. Deletes the object from the Amazon S3 bucket.
 * 8. Deletes the Amazon S3 bucket.
 */

public class S3Scenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
    "-");

    public static void main(String[] args) throws IOException {
        final String 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).\s
        """;

    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];
    Region region = Region.US_EAST_1;
    S3Client s3 = S3Client.builder()
        .region(region)
        .build();

    System.out.println(DASHES);
    System.out.println("Welcome to the Amazon S3 example scenario.");
    System.out.println(DASHES);

    System.out.println(DASHES);
    System.out.println("1. Create an Amazon S3 bucket.");
    createBucket(s3, bucketName);
    System.out.println(DASHES);

    System.out.println(DASHES);
    System.out.println("2. Update a local file to the Amazon S3 bucket.");
    uploadLocalFile(s3, bucketName, key, objectPath);
    System.out.println(DASHES);

    System.out.println(DASHES);
    System.out.println("3. Download the object to another local file.");
    getObjectBytes(s3, bucketName, key, savePath);
    System.out.println(DASHES);
```

```
System.out.println(DASHES);
System.out.println("4. Perform a multipart upload.");
String multipartKey = "multiPartKey";
multipartUpload(s3, toBucket, multipartKey);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("5. List all objects located in the Amazon S3
bucket.");
listAllObjects(s3, bucketName);
anotherListExample(s3, bucketName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("6. Copy the object to another Amazon S3 bucket.");
copyBucketObject(s3, bucketName, key, toBucket);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("7. Delete the object from the Amazon S3 bucket.");
deleteObjectFromBucket(s3, bucketName, key);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("8. Delete the Amazon S3 bucket.");
deleteBucket(s3, bucketName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("All Amazon S3 operations were successfully
performed");
System.out.println(DASHES);
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.
 */
public 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);
    }

    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.fromFile(new File(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 AWS SDK for Java 2.x.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Primeiro, importe todos os módulos necessários.

```
// Used to check if currently running file is this file.
import { fileURLToPath } from "url";
import { readdirSync, readFileSync, writeFileSync } from "fs";

// Local helper utils.
import { dirnameFromMetaUrl } from "@aws-doc-sdk-examples/lib/utils/util-fs.js";
import { Prompter } from "@aws-doc-sdk-examples/lib/prompter.js";
import { wrapText } from "@aws-doc-sdk-examples/lib/utils/util-string.js";

import {
  S3Client,
  CreateBucketCommand,
  PutObjectCommand,
  ListObjectsCommand,
  CopyObjectCommand,
  GetObjectCommand,
  DeleteObjectsCommand,
```

```
DeleteBucketCommand,  
} from "@aws-sdk/client-s3";
```

As importações anteriores fazem referência a alguns utilitários auxiliares. Esses utilitários são locais no repositório do GitHub vinculado no início desta seção. Para sua referência, consulte as implementações a seguir desses utilitários.

```
export const dirnameFromMetaUrl = (metaUrl) =>  
  fileURLToPath(new URL(".", metaUrl));  
  
import { select, input, confirm, checkbox } from "@inquirer/prompts";  
  
export class Prompter {  
  /**  
   * @param {{ message: string, choices: { name: string, value: string }[] }}  
   options  
   */  
  select(options) {  
    return select(options);  
  }  
  
  /**  
   * @param {{ message: string }} options  
   */  
  input(options) {  
    return input(options);  
  }  
  
  /**  
   * @param {string} prompt  
   */  
  checkContinue = async (prompt = "") => {  
    const prefix = prompt && prompt + " ";  
    let ok = await this.confirm({  
      message: `${prefix}Continue?`,  
    });  
    if (!ok) throw new Error("Exiting...");  
  };  
  
  /**  
   * @param {{ message: string }} options  
   */
```

```

confirm(options) {
  return confirm(options);
}

/**
 * @param {{ message: string, choices: { name: string, value: string }[] }}
options
 */
checkbox(options) {
  return checkbox(options);
}
}

export const wrapText = (text, char = "=") => {
  const rule = char.repeat(80);
  return `${rule}\n  ${text}\n${rule}\n`;
};

```

Os objetos no S3 são armazenados em “buckets”. Vamos definir uma função para criar um bucket.

```

export const createBucket = async () => {
  const bucketName = await prompter.input({
    message: "Enter a bucket name. Bucket names must be globally unique:",
  });
  const command = new CreateBucketCommand({ Bucket: bucketName });
  await s3Client.send(command);
  console.log("Bucket created successfully.\n");
  return bucketName;
};

```

Os buckets contêm “objetos”. Essa função faz upload do conteúdo de um diretório para seu bucket como objetos.

```

export const uploadFilesToBucket = async ({ bucketName, folderPath }) => {
  console.log(`Uploading files from ${folderPath}\n`);
  const keys = readdirSync(folderPath);
  const files = keys.map((key) => {
    const filePath = `${folderPath}/${key}`;
    const fileContent = readFileSync(filePath);
    return {

```

```
    Key: key,
    Body: fileContent,
  });
});

for (let file of files) {
  await s3Client.send(
    new PutObjectCommand({
      Bucket: bucketName,
      Body: file.Body,
      Key: file.Key,
    }),
  );
  console.log(`${file.Key} uploaded successfully.`);
}
};
```

Depois de fazer upload dos objetos, confira se eles foram carregados corretamente. Você pode usar `ListObjects` para isso. Você usará a propriedade “Key”, mas também há outras propriedades úteis na resposta.

```
export const listFilesInBucket = async ({ bucketName }) => {
  const command = new ListObjectsCommand({ Bucket: bucketName });
  const { Contents } = await s3Client.send(command);
  const contentsList = Contents.map((c) => ` • ${c.Key}`).join("\n");
  console.log("\nHere's a list of files in the bucket:");
  console.log(contentsList + "\n");
};
```

Às vezes é necessário copiar um objeto de um bucket em outro. Use o comando `CopyObject` para isso.

```
export const copyFileFromBucket = async ({ destinationBucket }) => {
  const proceed = await prompter.confirm({
    message: "Would you like to copy an object from another bucket?",
  });

  if (!proceed) {
    return;
  } else {
```

```
const copy = async () => {
  try {
    const sourceBucket = await prompter.input({
      message: "Enter source bucket name:",
    });
    const sourceKey = await prompter.input({
      message: "Enter source key:",
    });
    const destinationKey = await prompter.input({
      message: "Enter destination key:",
    });

    const command = new CopyObjectCommand({
      Bucket: destinationBucket,
      CopySource: `${sourceBucket}/${sourceKey}`,
      Key: destinationKey,
    });
    await s3Client.send(command);
    await copyFileFromBucket({ destinationBucket });
  } catch (err) {
    console.error(`Copy error.`);
    console.error(err);
    const retryAnswer = await prompter.confirm({ message: "Try again?" });
    if (retryAnswer) {
      await copy();
    }
  }
};
await copy();
};
```

Não há um método de SDK para obter vários objetos de um bucket. Em vez disso, você criará uma lista de objetos para baixar e iterar sobre eles.

```
export const downloadFilesFromBucket = async ({ bucketName }) => {
  const { Contents } = await s3Client.send(
    new ListObjectsCommand({ Bucket: bucketName }),
  );
  const path = await prompter.input({
    message: "Enter destination path for files:",
  });
};
```



```
for (let content of Contents) {
  const obj = await s3Client.send(
    new GetObjectCommand({ Bucket: bucketName, Key: content.Key }),
  );
  writeFileSync(
    `${path}/${content.Key}`,
    await obj.Body.transformToByteArray(),
  );
}
console.log("Files downloaded successfully.\n");
};
```

É hora de limpar seus recursos. Um bucket deve estar vazio para poder ser excluído. Essas duas funções esvaziam e excluem o bucket.

```
export const emptyBucket = async ({ bucketName }) => {
  const listObjectsCommand = new ListObjectsCommand({ Bucket: bucketName });
  const { Contents } = await s3Client.send(listObjectsCommand);
  const keys = Contents.map((c) => c.Key);

  const deleteObjectsCommand = new DeleteObjectsCommand({
    Bucket: bucketName,
    Delete: { Objects: keys.map((key) => ({ Key: key })) },
  });
  await s3Client.send(deleteObjectsCommand);
  console.log(`${bucketName} emptied successfully.\n`);
};

export const deleteBucket = async ({ bucketName }) => {
  const command = new DeleteBucketCommand({ Bucket: bucketName });
  await s3Client.send(command);
  console.log(`${bucketName} deleted successfully.\n`);
};
```

A função “principal” reúne tudo. Se você executar esse arquivo diretamente, a função principal será chamada.

```
const main = async () => {
  const OBJECT_DIRECTORY = `${dirnameFromMetaUrl(
    import.meta.url,
```

```
})../..../resources/sample_files/.sample_media`;  
  
try {  
  console.log(wrapText("Welcome to the Amazon S3 getting started example."));  
  console.log("Let's create a bucket.");  
  const bucketName = await createBucket();  
  await prompter.confirm({ message: continueMessage });  
  
  console.log(wrapText("File upload."));  
  console.log(  
    "I have some default files ready to go. You can edit the source code to  
provide your own.",  
  );  
  await uploadFilesToBucket({  
    bucketName,  
    folderPath: OBJECT_DIRECTORY,  
  });  
  
  await listFilesInBucket({ bucketName });  
  await prompter.confirm({ message: continueMessage });  
  
  console.log(wrapText("Copy files."));  
  await copyFileFromBucket({ destinationBucket: bucketName });  
  await listFilesInBucket({ bucketName });  
  await prompter.confirm({ message: continueMessage });  
  
  console.log(wrapText("Download files."));  
  await downloadFilesFromBucket({ bucketName });  
  
  console.log(wrapText("Clean up."));  
  await emptyBucket({ bucketName });  
  await deleteBucket({ bucketName });  
} catch (err) {  
  console.error(err);  
}  
};
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for JavaScript.
 - [CopyObject](#)
 - [CreateBucket](#)

- [DeleteBucket](#)
- [DeleteObjects](#)
- [GetObject](#)
- [ListObjectsV2](#)
- [PutObject](#)

Kotlin

SDK for Kotlin

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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.
getObjectFromMrap(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 getObjectFromMrap(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: UnsupportedOperationException) {
    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 AWS SDK para Kotlin.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

PHP

SDK para PHP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
echo("\n");
echo("-----\n");
print("Welcome to the Amazon S3 getting started demo using PHP!\n");
echo("-----\n");

$region = 'us-west-2';

$this->s3client = new S3Client([
    'region' => $region,
]);
/* Inline declaration example
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);
*/
```

```
$this->bucketName = "doc-example-bucket-" . uniqid();

try {
    $this->s3client->createBucket([
        'Bucket' => $this->bucketName,
        'CreateBucketConfiguration' => ['LocationConstraint' => $region],
    ]);
    echo "Created bucket named: $this->bucketName \n";
} catch (Exception $exception) {
    echo "Failed to create bucket $this->bucketName with error: " .
    $exception->getMessage();
    exit("Please fix error with bucket creation before continuing.");
}

$fileName = __DIR__ . "/local-file-" . uniqid();
try {
    $this->s3client->putObject([
        'Bucket' => $this->bucketName,
        'Key' => $fileName,
        'SourceFile' => __DIR__ . '/testfile.txt'
    ]);
    echo "Uploaded $fileName to $this->bucketName.\n";
} catch (Exception $exception) {
    echo "Failed to upload $fileName with error: " . $exception-
    >getMessage();
    exit("Please fix error with file upload before continuing.");
}

try {
    $file = $this->s3client->getObject([
        'Bucket' => $this->bucketName,
        'Key' => $fileName,
    ]);
    $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 $fileName from $this->bucketName with error:
    " . $exception->getMessage();
    exit("Please fix error with file downloading before continuing.");
}

try {
```



```
$folder = "copied-folder";
$this->s3client->copyObject([
    'Bucket' => $this->bucketName,
    'CopySource' => "$this->bucketName/$fileName",
    'Key' => "$folder/$fileName-copy",
]);
echo "Copied $fileName to $folder/$fileName-copy.\n";
} catch (Exception $exception) {
    echo "Failed to copy $fileName with error: " . $exception-
>getMessage();
    exit("Please fix error with object copying before continuing.");
}

try {
    $contents = $this->s3client->listObjectsV2([
        'Bucket' => $this->bucketName,
    ]);
    echo "The contents of your bucket are: \n";
    foreach ($contents['Contents'] as $content) {
        echo $content['Key'] . "\n";
    }
} catch (Exception $exception) {
    echo "Failed to list objects in $this->bucketName with error: " .
$exception->getMessage();
    exit("Please fix error with listing objects before continuing.");
}

try {
    $objects = [];
    foreach ($contents['Contents'] as $content) {
        $objects[] = [
            'Key' => $content['Key'],
        ];
    }
    $this->s3client->deleteObjects([
        'Bucket' => $this->bucketName,
        'Delete' => [
            'Objects' => $objects,
        ],
    ]);
    $check = $this->s3client->listObjectsV2([
        'Bucket' => $this->bucketName,
    ]);
    if (count($check) <= 0) {
```

```
        throw new Exception("Bucket wasn't empty.");
    }
    echo "Deleted all objects and folders from $this->bucketName.\n";
} catch (Exception $exception) {
    echo "Failed to delete $fileName from $this->bucketName with error:
" . $exception->getMessage();
    exit("Please fix error with object deletion before continuing.");
}

try {
    $this->s3client->deleteBucket([
        'Bucket' => $this->bucketName,
    ]);
    echo "Deleted bucket $this->bucketName.\n";
} catch (Exception $exception) {
    echo "Failed to delete $this->bucketName 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 AWS SDK for PHP.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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 AWS SDK para Python (Boto3).
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)

- [GetObject](#)
- [ListObjectsV2](#)
- [PutObject](#)

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
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: "us-east-1" # Note: only certain regions permitted
      }
    )
    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!")
  end
end
```

```
    raise
  else
    bucket
  end

  # Requests a file name from the user.
  #
  # @return The name of the file.
  def create_file
    File.open("demo.txt", w) { |f| f.write("This is a demo file.") }
  end

  # Uploads a file to an Amazon S3 bucket.
  #
  # @param bucket [Aws::S3::Bucket] The bucket object representing the upload
  destination
  # @return [Aws::S3::Object] The Amazon S3 object that contains the uploaded
  file.
  def upload_file(bucket)
    File.open("demo.txt", "w+") { |f| f.write("This is a demo file.") }
    s3_object = bucket.object(File.basename("demo.txt"))
    s3_object.upload_file("demo.txt")
    puts("Uploaded file demo.txt into bucket #{bucket.name} with key
  #{s3_object.key}.")
    rescue Aws::Errors::ServiceError => e
      puts("Couldn't upload file demo.txt 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
  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
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
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 AWS SDK for Ruby.

- [CopyObject](#)
- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteObjects](#)
- [GetObject](#)
- [ListObjectsV2](#)
- [PutObject](#)

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Código para a caixa binária que executa o cenário.

```
use aws_config::meta::region::RegionProviderChain;
use aws_sdk_s3::{config::Region, Client};
use s3_service::error::Error;
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?;
    let _object = 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::operation::{
    copy_object::{CopyObjectError, CopyObjectOutput},
    create_bucket::{CreateBucketError, CreateBucketOutput},
    get_object::{GetObjectError, GetObjectOutput},
    list_objects_v2::ListObjectsV2Output,
    put_object::{PutObjectError, PutObjectOutput},
};
use aws_sdk_s3::types::{
    BucketLocationConstraint, CreateBucketConfiguration, Delete,
    ObjectIdentifier,
};
use aws_sdk_s3::{error::SdkError, primitives::ByteStream, Client};
use error::Error;
use std::path::Path;
use std::str;

pub mod error;

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<Vec<String>, Error> {
    let objects = client.list_objects_v2().bucket(bucket_name).send().await?;

    let mut delete_objects: Vec<ObjectIdentifier> = vec![];
    for obj in objects.contents() {
        let obj_id = ObjectIdentifier::builder()
            .set_key(Some(obj.key().unwrap().to_string()))
            .build()
            .map_err(Error::from)?;
        delete_objects.push(obj_id);
    }

    let return_keys = delete_objects.iter().map(|o| o.key.clone()).collect();

    if !delete_objects.is_empty() {

```

```

    client
      .delete_objects()
      .bucket(bucket_name)
      .delete(
        Delete::builder()
          .set_objects(Some(delete_objects))
          .build()
          .map_err(Error::from)?,
      )
      .send()
      .await?;
  }

  let objects: ListObjectsV2Output =
  client.list_objects_v2().bucket(bucket_name).send().await?;

  eprintln!("{objects:?}");

  match objects.key_count {
    Some(0) => Ok(return_keys),
    _ => Err(Error::unhandled(
      "There were still objects left in the bucket.",
    )),
  }
}

pub async fn list_objects(client: &Client, bucket: &str) -> Result<(), Error> {
  let mut response = client
    .list_objects_v2()
    .bucket(bucket.to_owned())
    .max_keys(10) // In this example, go 10 at a time.
    .into_paginator()
    .send();

  while let Some(result) = response.next().await {
    match result {
      Ok(output) => {
        for object in output.contents() {
          println!(" - {}", object.key().unwrap_or("Unknown"));
        }
      }
      Err(err) => {
        eprintln!("{err:?}")
      }
    }
  }
}

```

```
    }
  }

  Ok(())
}

pub async fn copy_object(
  client: &Client,
  bucket_name: &str,
  object_key: &str,
  target_key: &str,
) -> Result<CopyObjectOutput, SdkError<CopyObjectError>> {
  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
}

pub async fn download_object(
  client: &Client,
  bucket_name: &str,
  key: &str,
) -> Result<GetObjectOutput, SdkError<GetObjectError>> {
  client
    .get_object()
    .bucket(bucket_name)
    .key(key)
    .send()
    .await
}

pub async fn upload_object(
  client: &Client,
  bucket_name: &str,
  file_name: &str,
  key: &str,
```

```
) -> Result<PutObjectOutput, SdkError<PutObjectError>> {
    let body = ByteStream::from_path(Path::new(file_name)).await;
    client
        .put_object()
        .bucket(bucket_name)
        .key(key)
        .body(body.unwrap())
        .send()
        .await
}

pub async fn create_bucket(
    client: &Client,
    bucket_name: &str,
    region: &str,
) -> Result<CreateBucketOutput, SdkError<CreateBucketError>> {
    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
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para Rust.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

SAP ABAP

SDK para SAP ABAP

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_s3) = /aws1/cl_s3_factory=>create( lo_session ).

" Create an Amazon Simple Storage Service (Amazon S3) bucket. "
TRY.
  lo_s3->createbucket(
    iv_bucket = iv_bucket_name
  ).
  MESSAGE 'S3 bucket created.' TYPE 'I'.
CATCH /aws1/cx_s3_bucketalrddyexists.
  MESSAGE 'Bucket name already exists.' TYPE 'E'.
CATCH /aws1/cx_s3_bktalrddyownedbyyou.
  MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
ENDTRY.

"Upload an object to an S3 bucket."
TRY.
  "Get contents of file from application server."
  DATA lv_file_content TYPE xstring.
  OPEN DATASET iv_key FOR INPUT IN BINARY MODE.
  READ DATASET iv_key INTO lv_file_content.
  CLOSE DATASET iv_key.

  lo_s3->putobject(
    iv_bucket = iv_bucket_name
    iv_key = iv_key
    iv_body = lv_file_content
  ).
  MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
```



```
ENDTRY.

" Get an object from a bucket. "
TRY.
  DATA(lo_result) = lo_s3->getobject(
    iv_bucket = iv_bucket_name
    iv_key = iv_key
  ).
  DATA(lv_object_data) = lo_result->get_body( ).
  MESSAGE 'Object retrieved from S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
CATCH /aws1/cx_s3_nosuchkey.
  MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.

" Copy an object to a subfolder in a bucket. "
TRY.
  lo_s3->copyobject(
    iv_bucket = iv_bucket_name
    iv_key = |{ iv_copy_to_folder }/{ iv_key }|
    iv_copysource = |{ iv_bucket_name }/{ iv_key }|
  ).
  MESSAGE 'Object copied to a subfolder.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
CATCH /aws1/cx_s3_nosuchkey.
  MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.

" List objects in the bucket. "
TRY.
  DATA(lo_list) = lo_s3->listobjects(
    iv_bucket = iv_bucket_name
  ).
  MESSAGE 'Retrieved list of objects in S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
DATA text TYPE string VALUE 'Object List - '.
DATA lv_object_key TYPE /aws1/s3_objectkey.
LOOP AT lo_list->get_contents( ) INTO DATA(lo_object).
  lv_object_key = lo_object->get_key( ).
  CONCATENATE lv_object_key ', ' INTO text.
```

```
ENDLOOP.
MESSAGE text TYPE 'I'.

" Delete the objects in a bucket. "
TRY.
  lo_s3->deleteobject(
    iv_bucket = iv_bucket_name
    iv_key = iv_key
  ).
  lo_s3->deleteobject(
    iv_bucket = iv_bucket_name
    iv_key = |{ iv_copy_to_folder }/{ iv_key }|
  ).
  MESSAGE 'Objects deleted from S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.

" Delete the bucket. "
TRY.
  lo_s3->deletebucket(
    iv_bucket = iv_bucket_name
  ).
  MESSAGE 'Deleted S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para SAP ABAP.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

Swift

SDK para Swift

Note

Esta é a documentação de pré-lançamento de um SDK na versão de visualização. Está sujeita a alteração.

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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 S3Client(region: "us-east-2")
        } 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 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)
}
```

```
/// 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,
        let data = try await body.readData() else {
        return
    }
    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,
        let data = try await body.readData() else {
        return "".data(using: .utf8)!
    }

    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 AWS SDK para Swift.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Conceitos básicos de criptografia para objetos do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como começar a usar a criptografia de objetos do Amazon S3.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
using System;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to apply client encryption to an object in an
/// Amazon Simple Storage Service (Amazon S3) bucket.
/// </summary>
public class SSEClientEncryption
{
    public static async Task Main()
    {
        string bucketName = "doc-example-bucket";
        string keyName = "exampleobject.txt";
        string copyTargetKeyName = "examplecopy.txt";

        // If the AWS Region defined for your default user is different
        // from the Region where your Amazon S3 bucket is located,
        // pass the Region name to the Amazon S3 client object's constructor.
```

```
// For example: RegionEndpoint.USWest2.
IAmazonS3 client = new AmazonS3Client();

try
{
    // Create an encryption key.
    Aes aesEncryption = Aes.Create();
    aesEncryption.KeySize = 256;
    aesEncryption.GenerateKey();
    string base64Key = Convert.ToBase64String(aesEncryption.Key);

    // Upload the object.
    PutObjectRequest putObjectRequest = await
UploadObjectAsync(client, bucketName, keyName, base64Key);

    // Download the object and verify that its contents match what
you uploaded.
    await DownloadObjectAsync(client, bucketName, keyName, base64Key,
putObjectRequest);

    // Get object metadata and verify that the object uses AES-256
encryption.
    await GetObjectMetadataAsync(client, bucketName, keyName,
base64Key);

    // Copy both the source and target objects using server-side
encryption with
    // an encryption key.
    await CopyObjectAsync(client, bucketName, keyName,
copyTargetKeyName, aesEncryption, base64Key);
}
catch (AmazonS3Exception ex)
{
    Console.WriteLine($"Error: {ex.Message}");
}
}

/// <summary>
/// Uploads an object to an Amazon S3 bucket.
/// </summary>
/// <param name="client">The initialized Amazon S3 client object used to
call
/// PutObjectAsync.</param>
```

```

    /// <param name="bucketName">The name of the Amazon S3 bucket to which
the
    /// object will be uploaded.</param>
    /// <param name="keyName">The name of the object to upload to the Amazon
S3
    /// bucket.</param>
    /// <param name="base64Key">The encryption key.</param>
    /// <returns>The PutObjectRequest object for use by
DownloadObjectAsync.</returns>
    public static async Task<PutObjectRequest> UploadObjectAsync(
        IAmazonS3 client,
        string bucketName,
        string keyName,
        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;
    }

    /// <summary>
    /// Downloads an encrypted object from an Amazon S3 bucket.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// GetObjectAsync.</param>
    /// <param name="bucketName">The name of the Amazon S3 bucket where the
object
    /// is located.</param>
    /// <param name="keyName">The name of the Amazon S3 object to download.</
param>
    /// <param name="base64Key">The encryption key used to encrypt the
    /// object.</param>
    /// <param name="putObjectRequest">The PutObjectRequest used to upload
    /// the object.</param>

```

```
public static async Task DownloadObjectAsync(
    IAmazonS3 client,
    string bucketName,
    string keyName,
    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");
            }
        }
    }
}
```

```

        }
    }
}

/// <summary>
/// Retrieves the metadata associated with an Amazon S3 object.
/// </summary>
/// <param name="client">The initialized Amazon S3 client object used
/// to call GetObjectMetadataAsync.</param>
/// <param name="bucketName">The name of the Amazon S3 bucket containing
the
/// object for which we want to retrieve metadata.</param>
/// <param name="keyName">The name of the object for which we wish to
/// retrieve the metadata.</param>
/// <param name="base64Key">The encryption key associated with the
/// object.</param>
public static async Task GetObjectMetadataAsync(
    IAmazonS3 client,
    string bucketName,
    string keyName,
    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);
}

/// <summary>
/// Copies an encrypted object from one Amazon S3 bucket to another.
/// </summary>

```

```
call    /// <param name="client">The initialized Amazon S3 client object used to
        /// CopyObjectAsync.</param>
        /// <param name="bucketName">The Amazon S3 bucket containing the object
        /// to copy.</param>
        /// <param name="keyName">The name of the object to copy.</param>
        /// <param name="copyTargetKeyName">The Amazon S3 bucket to which the
object    object
        /// will be copied.</param>
        /// <param name="aesEncryption">The encryption type to use.</param>
        /// <param name="base64Key">The encryption key to use.</param>
        public static async Task CopyObjectAsync(
            IAmazonS3 client,
            string bucketName,
            string keyName,
            string copyTargetKeyName,
            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);
        }
    }
```


- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for .NET.
 - [CopyObject](#)
 - [GetObject](#)
 - [GetObjectMetadata](#)

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Conceitos básicos de etiquetas para objetos do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como começar a usar etiquetas de objetos do Amazon S3.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to work with tags in Amazon Simple Storage
/// Service (Amazon S3) objects.
/// </summary>
public class ObjectTag
```

```
{
    public static async Task Main()
    {
        string bucketName = "doc-example-bucket";
        string keyName = "newobject.txt";
        string filePath = @"*** file path ***";

        // Specify your bucket region (an example region is shown).
        RegionEndpoint bucketRegion = RegionEndpoint.USWest2;

        var client = new AmazonS3Client(bucketRegion);
        await PutObjectsWithTagsAsync(client, bucketName, keyName, filePath);
    }

    /// <summary>
    /// This method uploads an object with tags. It then shows the tag
    /// values, changes the tags, and shows the new tags.
    /// </summary>
    /// <param name="client">The Initialized Amazon S3 client object used
    /// to call the methods to create and change an objects tags.</param>
    /// <param name="bucketName">A string representing the name of the
    /// bucket where the object will be stored.</param>
    /// <param name="keyName">A string representing the key name of the
    /// object to be tagged.</param>
    /// <param name="filePath">The directory location and file name of the
    /// object to be uploaded to the Amazon S3 bucket.</param>
    public static async Task PutObjectsWithTagsAsync(IAmazonS3 client, string
bucketName, string keyName, string filePath)
    {
        try
        {
            // Create 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);

        // Now retrieve the new object's tags.
        GetObjectTaggingRequest getTagsRequest = new
GetObjectTaggingRequest()
        {
            BucketName = bucketName,
            Key = keyName,
        };

        GetObjectTaggingResponse objectTags = await
client.GetObjectTaggingAsync(getTagsRequest);

        // Display the tag values.
        objectTags.Tagging
            .ForEach(t => Console.WriteLine($"Key: {t.Key}, Value:
{t.Value}"));

        Tagging newTagSet = new Tagging()
        {
            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);

        // Retrieve the tags again and show the values.
        GetObjectTaggingRequest getTagsRequest2 = new
GetObjectTaggingRequest()
        {
            BucketName = bucketName,
```

```
        Key = keyName,
    };
    GetObjectTaggingResponse objectTags2 = await
client.GetObjectTaggingAsync(getTagsRequest2);

    objectTags2.Tagging
        .ForEach(t => Console.WriteLine($"Key: {t.Key}, Value:
{t.Value}"));
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine(
            $"Error: '{ex.Message}'");
    }
}
}
```

- Para obter detalhes da API, consulte [GetObjectTagging](#) na Referência da API do AWS SDK for .NET.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Obter a configuração de retenção legal de um objeto do Amazon S3 usando um SDK da AWS

O exemplo de código a seguir mostra como obter a configuração de retenção jurídica de um bucket do S3.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).


```
/// <summary>
/// Get the legal hold details for an S3 object.
/// </summary>
/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The object key.</param>
/// <returns>The object legal hold details.</returns>
public async Task<ObjectLockLegalHold> GetObjectLegalHold(string bucketName,
    string objectKey)
{
    try
    {
        var request = new GetObjectLegalHoldRequest()
        {
            BucketName = bucketName,
            Key = objectKey
        };

        var response = await _amazonS3.GetObjectLegalHoldAsync(request);
        Console.WriteLine($"{objectKey} in
{bucketName}: " +
            $"{response.LegalHold.Status}");
        return response.LegalHold;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Unable to fetch legal hold: '{ex.Message}'");
        return new ObjectLockLegalHold();
    }
}
```

- Consulte detalhes da API em [GetObjectLegalHold](#) na Referência de API do AWS SDK for .NET.

Java

SDK para Java 2.x

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
// Get the legal hold details for an S3 object.
public ObjectLockLegalHold getObjectLegalHold(String bucketName, String
objectKey) {
    try {
        GetObjectLegalHoldRequest legalHoldRequest =
GetObjectLegalHoldRequest.builder()
            .bucket(bucketName)
            .key(objectKey)
            .build();

        GetObjectLegalHoldResponse response =
getClient().getObjectLegalHold(legalHoldRequest);
        System.out.println("Object legal hold for " + objectKey + " in " +
bucketName +
            ":\n\tStatus: " + response.legalHold().status());
        return response.legalHold();

    } catch (S3Exception ex) {
        System.out.println("\tUnable to fetch legal hold: '" +
ex.getMessage() + "'");
    }

    return null;
}
```

- Consulte detalhes da API em [GetObjectLegalHold](#) na Referência de API do AWS SDK for Java 2.x.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Trabalhar com os recursos de bloqueio de objetos do Amazon S3 usando um SDK da AWS

Os exemplos de código a seguir mostram como trabalhar com os recursos do Bloqueio de Objetos do S3.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Execute um cenário interativo demonstrando os recursos de bloqueio de objetos do Amazon S3.

```
using Amazon.S3;
using Amazon.S3.Model;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging;
using Microsoft.Extensions.Logging.Console;
using Microsoft.Extensions.Logging.Debug;

namespace S3ObjectLockScenario;

public static class S3ObjectLockWorkflow
{
    /*
     Before running this .NET code example, set up your development environment,
     including your credentials.
    */
}
```

This .NET example performs the following tasks:

1. Create test Amazon Simple Storage Service (S3) buckets with different lock policies.
2. Upload sample objects to each bucket.
3. Set some Legal Hold and Retention Periods on objects and buckets.
4. Investigate lock policies by viewing settings or attempting to delete or overwrite objects.
5. Clean up objects and buckets.

```
*/
```

```
public static S3ActionsWrapper _s3ActionsWrapper = null!;  
public static IConfiguration _configuration = null!;  
private static string _resourcePrefix = null!;  
private static string noLockBucketName = null!;  
private static string lockEnabledBucketName = null!;  
private static string retentionAfterCreationBucketName = null!;  
private static List<string> bucketNames = new List<string>();  
private static List<string> fileNames = new List<string>();  
  
public static async Task Main(string[] args)  
{  
    // Set up dependency injection for the Amazon service.  
    using var host = Host.CreateDefaultBuilder(args)  
        .ConfigureLogging(logging =>  
            logging.AddFilter("System", LogLevel.Debug)  
                .AddFilter<DebugLoggerProvider>("Microsoft",  
LogLevel.Information)  
                .AddFilter<ConsoleLoggerProvider>("Microsoft",  
LogLevel.Trace))  
        .ConfigureServices((_, services) =>  
            services.AddAWSService<IAmazonS3>()  
                .AddTransient<S3ActionsWrapper>()  
        )  
        .Build();  
  
    _configuration = new ConfigurationBuilder()  
        .SetBasePath(Directory.GetCurrentDirectory())  
        .AddJsonFile("settings.json") // Load settings from .json file.  
        .AddJsonFile("settings.local.json",  
            true) // Optionally, load local settings.  
        .Build();  
  
    ConfigurationSetup();  
}
```



```
        ServicesSetup(host);

        try
        {
            Console.WriteLine(new string('-', 80));
            Console.WriteLine("Welcome to the Amazon Simple Storage Service (S3)
Object Locking Workflow Scenario.");
            Console.WriteLine(new string('-', 80));
            await Setup(true);

            await DemoActionChoices();

            Console.WriteLine(new string('-', 80));
            Console.WriteLine("Cleaning up resources.");
            Console.WriteLine(new string('-', 80));
            await Cleanup(true);

            Console.WriteLine(new string('-', 80));
            Console.WriteLine("Amazon S3 Object Locking Workflow is complete.");
            Console.WriteLine(new string('-', 80));
        }
        catch (Exception ex)
        {
            Console.WriteLine(new string('-', 80));
            Console.WriteLine($"There was a problem: {ex.Message}");
            await Cleanup(true);
            Console.WriteLine(new string('-', 80));
        }
    }

    /// <summary>
    /// Populate the services for use within the console application.
    /// </summary>
    /// <param name="host">The services host.</param>
    private static void ServicesSetup(IHost host)
    {
        _s3ActionsWrapper = host.Services.GetRequiredService<S3ActionsWrapper>();
    }

    /// <summary>
    /// Any setup operations needed.
    /// </summary>
    public static void ConfigurationSetup()
    {
```

```
        _resourcePrefix = _configuration["resourcePrefix"] ?? "dotnet-example";

        noLockBucketName = _resourcePrefix + "-no-lock";
        lockEnabledBucketName = _resourcePrefix + "-lock-enabled";
        retentionAfterCreationBucketName = _resourcePrefix + "-retention-after-creation";

        bucketNames.Add(noLockBucketName);
        bucketNames.Add(lockEnabledBucketName);
        bucketNames.Add(retentionAfterCreationBucketName);
    }

    /// <summary>
    /// Deploy necessary resources for the scenario.
    /// </summary>
    /// <param name="interactive">True to run as interactive.</param>
    /// <returns>True if successful.</returns>
    public static async Task<bool> Setup(bool interactive)
    {
        Console.WriteLine(
            "\nFor this workflow, we will use the AWS SDK for .NET to create
            several S3\n" +
            "buckets and files to demonstrate working with S3 locking features.
            \n");

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Press Enter when you are ready to start.");
        if (interactive)
            Console.ReadLine();

        Console.WriteLine("\nS3 buckets can be created either with or without
        object lock enabled.");
        await _s3ActionsWrapper.CreateBucketWithObjectLock(noLockBucketName,
        false);
        await _s3ActionsWrapper.CreateBucketWithObjectLock(lockEnabledBucketName,
        true);
        await
        _s3ActionsWrapper.CreateBucketWithObjectLock(retentionAfterCreationBucketName,
        false);

        Console.WriteLine("Press Enter to continue.");
        if (interactive)
            Console.ReadLine();
    }
}
```

```
        Console.WriteLine("\nA bucket can be configured to use object locking
with a default retention period.");
        await
_s3ActionsWrapper.ModifyBucketDefaultRetention(retentionAfterCreationBucketName,
true,
        ObjectLockRetentionMode.Governance, DateTime.UtcNow.AddDays(1));

        Console.WriteLine("Press Enter to continue.");
        if (interactive)
            Console.ReadLine();

        Console.WriteLine("\nObject lock policies can also be added to existing
buckets.");
        await _s3ActionsWrapper.EnableObjectLockOnBucket(lockEnabledBucketName);

        Console.WriteLine("Press Enter to continue.");
        if (interactive)
            Console.ReadLine();

        // Upload some files to the buckets.
        Console.WriteLine("\nNow let's add some test files:");
        var fileName = _configuration["exampleFileName"] ?? "exampleFile.txt";
        int fileCount = 2;
        // Create the file if it does not already exist.
        if (!File.Exists(fileName))
        {
            await using StreamWriter sw = File.CreateText(fileName);
            await sw.WriteLineAsync(
                "This is a sample file for uploading to a bucket.");
        }

        foreach (var bucketName in bucketNames)
        {
            for (int i = 0; i < fileCount; i++)
            {
                var numberedFileName = Path.GetFileNameWithoutExtension(fileName)
+ i + Path.GetExtension(fileName);
                fileNames.Add(numberedFileName);
                await _s3ActionsWrapper.UploadFileAsync(bucketName,
numberedFileName, fileName);
            }
        }
        Console.WriteLine("Press Enter to continue.");
        if (interactive)
```

```
        Console.ReadLine();

        if (!interactive)
            return true;
        Console.WriteLine("\nNow we can set some object lock policies on
individual files:");
        foreach (var bucketName in bucketNames)
        {
            for (int i = 0; i < fileNames.Count; i++)
            {
                // No modifications to the objects in the first bucket.
                if (bucketName != bucketNames[0])
                {
                    var exampleFileName = fileNames[i];
                    switch (i)
                    {
                        case 0:
                            {
                                var question =
                                    $"{exampleFileName} in {bucketName}? (y/n)";
                                if (GetYesNoResponse(question))
                                {
                                    // Set a legal hold.
                                    await
                                    _s3ActionsWrapper.ModifyObjectLegalHold(bucketName, exampleFileName,
                                    ObjectLockLegalHoldStatus.On);

                                }
                                break;
                            }
                        case 1:
                            {
                                var question =
                                    $"{exampleFileName} in {bucketName}? (y/n)" +
                                    "\nReminder: Only a user with the
s3:BypassGovernanceRetention permission will be able to delete this file or its
bucket until the retention period has expired.";
                                if (GetYesNoResponse(question))
                                {
                                    // Set a Governance mode retention period for
                                    1 day.
```

```
                await
_s3ActionsWrapper.ModifyObjectRetentionPeriod(
                    bucketName, exampleFileName,
                    ObjectLockRetentionMode.Governance,
                    DateTime.UtcNow.AddDays(1));
            }
            break;
        }
    }
}
}
}
}
Console.WriteLine(new string('-', 80));
return true;
}

// <summary>
/// List all of the current buckets and objects.
/// </summary>
/// <param name="interactive">True to run as interactive.</param>
/// <returns>The list of buckets and objects.</returns>
public static async Task<List<S3ObjectVersion>> ListBucketsAndObjects(bool
interactive)
{
    var allObjects = new List<S3ObjectVersion>();
    foreach (var bucketName in bucketNames)
    {
        var objectsInBucket = await
_s3ActionsWrapper.ListBucketObjectsAndVersions(bucketName);
        foreach (var objectKey in objectsInBucket.Versions)
        {
            allObjects.Add(objectKey);
        }
    }

    if (interactive)
    {
        Console.WriteLine("\nCurrent buckets and objects:\n");
        int i = 0;
        foreach (var bucketObject in allObjects)
        {
            i++;
            Console.WriteLine(
```

```

        $"{i}: {bucketObject.Key} \n\tBucket:
{bucketObject.BucketName}\n\tVersion: {bucketObject.VersionId}");
    }
}

return allObjects;
}

/// <summary>
/// Present the user with the demo action choices.
/// </summary>
/// <returns>Async task.</returns>
public static async Task<bool> DemoActionChoices()
{
    var choices = new string[]{
        "List all files in buckets.",
        "Attempt to delete a file.",
        "Attempt to delete a file with retention period bypass.",
        "Attempt to overwrite a file.",
        "View the object and bucket retention settings for a file.",
        "View the legal hold settings for a file.",
        "Finish the workflow."};

    var choice = 0;
    // Keep asking the user until they choose to move on.
    while (choice != 6)
    {
        Console.WriteLine(new string('-', 80));
        choice = GetChoiceResponse(
            "\nExplore the S3 locking features by selecting one of the
following choices:"
            , choices);
        Console.WriteLine(new string('-', 80));
        switch (choice)
        {
            case 0:
            {
                await ListBucketsAndObjects(true);
                break;
            }
            case 1:
            {
                Console.WriteLine("\nEnter the number of the object to
delete:");

```

```
        var allFiles = await ListBucketsAndObjects(true);
        var fileChoice = GetChoiceResponse(null,
allFiles.Select(f => f.Key).ToArray());
        await
_s3ActionsWrapper.DeleteObjectFromBucket(allFiles[fileChoice].BucketName,
allFiles[fileChoice].Key, false, allFiles[fileChoice].VersionId);
        break;
    }
    case 2:
    {
        Console.WriteLine("\nEnter the number of the object to
delete:");

        var allFiles = await ListBucketsAndObjects(true);
        var fileChoice = GetChoiceResponse(null,
allFiles.Select(f => f.Key).ToArray());
        await
_s3ActionsWrapper.DeleteObjectFromBucket(allFiles[fileChoice].BucketName,
allFiles[fileChoice].Key, true, allFiles[fileChoice].VersionId);
        break;
    }
    case 3:
    {
        var allFiles = await ListBucketsAndObjects(true);
        Console.WriteLine("\nEnter the number of the object to
overwrite:");

        var fileChoice = GetChoiceResponse(null,
allFiles.Select(f => f.Key).ToArray());
        // Create the file if it does not already exist.
        if (!File.Exists(allFiles[fileChoice].Key))
        {
            await using StreamWriter sw =
File.CreateText(allFiles[fileChoice].Key);
            await sw.WriteLineAsync(
                "This is a sample file for uploading to a
bucket.");
        }
        await
_s3ActionsWrapper.UploadFileAsync(allFiles[fileChoice].BucketName,
allFiles[fileChoice].Key, allFiles[fileChoice].Key);
        break;
    }
    case 4:
    {
        var allFiles = await ListBucketsAndObjects(true);
```

```

        Console.WriteLine("\nEnter the number of the object and
bucket to view:");
        var fileChoice = GetChoiceResponse(null,
allFiles.Select(f => f.Key).ToArray());
        await
_s3ActionsWrapper.GetObjectRetention(allFiles[fileChoice].BucketName,
allFiles[fileChoice].Key);
        await
_s3ActionsWrapper.GetBucketObjectLockConfiguration(allFiles[fileChoice].BucketName);
        break;
    }
    case 5:
    {
        var allFiles = await ListBucketsAndObjects(true);
        Console.WriteLine("\nEnter the number of the object to
view:");
        var fileChoice = GetChoiceResponse(null,
allFiles.Select(f => f.Key).ToArray());
        await
_s3ActionsWrapper.GetObjectLegalHold(allFiles[fileChoice].BucketName,
allFiles[fileChoice].Key);
        break;
    }
    }
}
return true;
}

// <summary>
/// Clean up the resources from the scenario.
/// </summary>
/// <param name="interactive">True to run as interactive.</param>
/// <returns>True if successful.</returns>
public static async Task<bool> Cleanup(bool interactive)
{
    Console.WriteLine(new string('-', 80));

    if (!interactive || GetYesNoResponse("Do you want to clean up all files
and buckets? (y/n) "))
    {
        // Remove all locks and delete all buckets and objects.
        var allFiles = await ListBucketsAndObjects(false);
        foreach (var fileInfo in allFiles)
        {

```



```
        // Check for a legal hold.
        var legalHold = await
        _s3ActionsWrapper.GetObjectLegalHold(fileInfo.BucketName, fileInfo.Key);
        if (legalHold?.Status?.Value == ObjectLockLegalHoldStatus.On)
        {
            await
            _s3ActionsWrapper.ModifyObjectLegalHold(fileInfo.BucketName, fileInfo.Key,
            ObjectLockLegalHoldStatus.Off);
        }

        // Check for a retention period.
        var retention = await
        _s3ActionsWrapper.GetObjectRetention(fileInfo.BucketName, fileInfo.Key);
        var hasRetentionPeriod = retention?.Mode ==
        ObjectLockRetentionMode.Governance && retention.RetainUntilDate >
        DateTime.UtcNow.Date;
        await
        _s3ActionsWrapper.DeleteObjectFromBucket(fileInfo.BucketName, fileInfo.Key,
        hasRetentionPeriod, fileInfo.VersionId);
    }

    foreach (var bucketName in bucketNames)
    {
        await _s3ActionsWrapper.DeleteBucketByName(bucketName);
    }
}
else
{
    Console.WriteLine(
        "Ok, we'll leave the resources intact.\n" +
        "Don't forget to delete them when you're done with them or you
might incur unexpected charges."
    );
}

Console.WriteLine(new string('-', 80));
return true;
}

/// <summary>
/// Helper method to get a yes or no response from the user.
/// </summary>
```

```
    /// <param name="question">The question string to print on the console.</  
param>  
    /// <returns>True if the user responds with a yes.</returns>  
    private static bool GetYesNoResponse(string question)  
    {  
        Console.WriteLine(question);  
        var ynResponse = Console.ReadLine();  
        var response = ynResponse != null && ynResponse.Equals("y",  
StringComparison.InvariantCultureIgnoreCase);  
        return response;  
    }  
  
    /// <summary>  
    /// Helper method to get a choice response from the user.  
    /// </summary>  
    /// <param name="question">The question string to print on the console.</  
param>  
    /// <param name="choices">The choices to print on the console.</param>  
    /// <returns>The index of the selected choice</returns>  
    private static int GetChoiceResponse(string? question, string[] choices)  
    {  
        if (question != null)  
        {  
            Console.WriteLine(question);  
  
            for (int i = 0; i < choices.Length; i++)  
            {  
                Console.WriteLine($"{i + 1}. {choices[i]}");  
            }  
        }  
  
        var choiceNumber = 0;  
        while (choiceNumber < 1 || choiceNumber > choices.Length)  
        {  
            var choice = Console.ReadLine();  
            Int32.TryParse(choice, out choiceNumber);  
        }  
  
        return choiceNumber - 1;  
    }  
}
```

Uma classe de wrapper para funções do S3.

```
using System.Net;
using Amazon.S3;
using Amazon.S3.Model;
using Microsoft.Extensions.Configuration;

namespace S3ObjectLockScenario;

/// <summary>
/// Encapsulate the Amazon S3 operations.
/// </summary>
public class S3ActionsWrapper
{
    private readonly IAmazonS3 _amazonS3;

    /// <summary>
    /// Constructor for the S3ActionsWrapper.
    /// </summary>
    /// <param name="amazonS3">The injected S3 client.</param>
    public S3ActionsWrapper(IAmazonS3 amazonS3, IConfiguration configuration)
    {
        _amazonS3 = amazonS3;
    }

    /// <summary>
    /// Create a new Amazon S3 bucket with object lock actions.
    /// </summary>
    /// <param name="bucketName">The name of the bucket to create.</param>
    /// <param name="enableObjectLock">True to enable object lock on the
    bucket.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> CreateBucketWithObjectLock(string bucketName, bool
    enableObjectLock)
    {
        Console.WriteLine($"{\tCreating bucket {bucketName} with object lock
    {enableObjectLock}."");
        try
        {
            var request = new PutBucketRequest
            {
                BucketName = bucketName,
                UseClientRegion = true,
```

```
        ObjectLockEnabledForBucket = enableObjectLock,
    };

    var response = await _amazonS3.PutBucketAsync(request);

    return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
}
catch (AmazonS3Exception ex)
{
    Console.WriteLine($"Error creating bucket: '{ex.Message}'");
    return false;
}
}

/// <summary>
/// Enable object lock on an existing bucket.
/// </summary>
/// <param name="bucketName">The name of the bucket to modify.</param>
/// <returns>True if successful.</returns>
public async Task<bool> EnableObjectLockOnBucket(string bucketName)
{
    try
    {
        // First, enable Versioning on the bucket.
        await _amazonS3.PutBucketVersioningAsync(new
PutBucketVersioningRequest()
        {
            BucketName = bucketName,
            VersioningConfig = new S3BucketVersioningConfig()
            {
                EnableMfaDelete = false,
                Status = VersionStatus.Enabled
            }
        });

        var request = new PutObjectLockConfigurationRequest()
        {
            BucketName = bucketName,
            ObjectLockConfiguration = new ObjectLockConfiguration()
            {
                ObjectLockEnabled = new ObjectLockEnabled("Enabled"),
            },
        };
    }
}
```

```
        var response = await
_amazonS3.PutObjectLockConfigurationAsync(request);
        Console.WriteLine($"\\tAdded an object lock policy to bucket
{bucketName}.");
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error modifying object lock: '{ex.Message}'");
        return false;
    }
}

/// <summary>
/// Set or modify a retention period on an object in an S3 bucket.
/// </summary>
/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The key of the object.</param>
/// <param name="retention">The retention mode.</param>
/// <param name="retainUntilDate">The date retention expires.</param>
/// <returns>True if successful.</returns>
public async Task<bool> ModifyObjectRetentionPeriod(string bucketName,
    string objectKey, ObjectLockRetentionMode retention, DateTime
retainUntilDate)
{
    try
    {
        var request = new PutObjectRetentionRequest()
        {
            BucketName = bucketName,
            Key = objectKey,
            Retention = new ObjectLockRetention()
            {
                Mode = retention,
                RetainUntilDate = retainUntilDate
            }
        };

        var response = await _amazonS3.PutObjectRetentionAsync(request);
        Console.WriteLine($"\\tSet retention for {objectKey} in {bucketName}
until {retainUntilDate:d}.");
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
```

```
        {
            Console.WriteLine($"\\tError modifying retention period:
'{ex.Message}'");
            return false;
        }
    }

    /// <summary>
    /// Set or modify a retention period on an S3 bucket.
    /// </summary>
    /// <param name="bucketName">The bucket to modify.</param>
    /// <param name="retention">The retention mode.</param>
    /// <param name="retainUntilDate">The date for retention until.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> ModifyBucketDefaultRetention(string bucketName, bool
enableObjectLock, ObjectLockRetentionMode retention, DateTime retainUntilDate)
    {
        var enabledString = enableObjectLock ? "Enabled" : "Disabled";
        var timeDifference = retainUntilDate.Subtract(DateTime.Now);
        try
        {
            // First, enable Versioning on the bucket.
            await _amazonS3.PutBucketVersioningAsync(new
PutBucketVersioningRequest()
            {
                BucketName = bucketName,
                VersioningConfig = new S3BucketVersioningConfig()
                {
                    EnableMfaDelete = false,
                    Status = VersionStatus.Enabled
                }
            });

            var request = new PutObjectLockConfigurationRequest()
            {
                BucketName = bucketName,
                ObjectLockConfiguration = new ObjectLockConfiguration()
                {
                    ObjectLockEnabled = new ObjectLockEnabled(enabledString),
                    Rule = new ObjectLockRule()
                    {
                        DefaultRetention = new DefaultRetention()
                        {
                            Mode = retention,

```

```
        Days = timeDifference.Days // Can be specified in
days or years but not both.
    }
}
};

var response = await
_amazonS3.PutObjectLockConfigurationAsync(request);
Console.WriteLine($"\\tAdded a default retention to bucket
{bucketName}.");
return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
}
catch (AmazonS3Exception ex)
{
    Console.WriteLine($"\\tError modifying object lock: '{ex.Message}'");
    return false;
}
}

/// <summary>
/// Get the retention period for an S3 object.
/// </summary>
/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The object key.</param>
/// <returns>The object retention details.</returns>
public async Task<ObjectLockRetention> GetObjectRetention(string bucketName,
string objectKey)
{
    try
    {
        var request = new GetObjectRetentionRequest()
        {
            BucketName = bucketName,
            Key = objectKey
        };

        var response = await _amazonS3.GetObjectRetentionAsync(request);
        Console.WriteLine($"\\tObject retention for {objectKey} in
{bucketName}: " +
            $"\\n\\t{response.Retention.Mode} until
{response.Retention.RetainUntilDate:d}.");
        return response.Retention;
    }
}
```

```
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\\tUnable to fetch object lock retention:
'{{ex.Message}}'");
            return new ObjectLockRetention();
        }
    }

    /// <summary>
    /// Set or modify a legal hold on an object in an S3 bucket.
    /// </summary>
    /// <param name="bucketName">The bucket of the object.</param>
    /// <param name="objectKey">The key of the object.</param>
    /// <param name="holdStatus">The On or Off status for the legal hold.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> ModifyObjectLegalHold(string bucketName,
        string objectKey, ObjectLockLegalHoldStatus holdStatus)
    {
        try
        {
            var request = new PutObjectLegalHoldRequest()
            {
                BucketName = bucketName,
                Key = objectKey,
                LegalHold = new ObjectLockLegalHold()
                {
                    Status = holdStatus
                }
            };

            var response = await _amazonS3.PutObjectLegalHoldAsync(request);
            Console.WriteLine($"\\tModified legal hold for {{objectKey}} in
{{bucketName}}.");
            return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\\tError modifying legal hold: '{{ex.Message}}'");
            return false;
        }
    }

    /// <summary>
    /// Get the legal hold details for an S3 object.
```



```
/// </summary>
/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The object key.</param>
/// <returns>The object legal hold details.</returns>
public async Task<ObjectLockLegalHold> GetObjectLegalHold(string bucketName,
    string objectKey)
{
    try
    {
        var request = new GetObjectLegalHoldRequest()
        {
            BucketName = bucketName,
            Key = objectKey
        };

        var response = await _amazonS3.GetObjectLegalHoldAsync(request);
        Console.WriteLine($"{objectKey} in
{bucketName}: " +
            $"{response.LegalHold.Status}");
        return response.LegalHold;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"{ex.Message}");
        return new ObjectLockLegalHold();
    }
}

/// <summary>
/// Get the object lock configuration details for an S3 bucket.
/// </summary>
/// <param name="bucketName">The bucket to get details.</param>
/// <returns>The bucket's object lock configuration details.</returns>
public async Task<ObjectLockConfiguration>
GetBucketObjectLockConfiguration(string bucketName)
{
    try
    {
        var request = new GetObjectLockConfigurationRequest()
        {
            BucketName = bucketName
        };
    }
}
```

```
        var response = await
_amazonS3.GetObjectLockConfigurationAsync(request);
        Console.WriteLine($"\\tBucket object lock config for {bucketName} in
{bucketName}: " +
            $"\\n\\tEnabled:
{response.ObjectLockConfiguration.ObjectLockEnabled}" +
            $"\\n\\tRule:
{response.ObjectLockConfiguration.Rule?.DefaultRetention}");

        return response.ObjectLockConfiguration;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"\\tUnable to fetch object lock config:
'{ex.Message}");
        return new ObjectLockConfiguration();
    }
}

/// <summary>
/// Upload a file from the local computer to an Amazon S3 bucket.
/// </summary>
/// <param name="bucketName">The Amazon S3 bucket to use.</param>
/// <param name="objectName">The object to upload.</param>
/// <param name="filePath">The path, including file name, of the object to
upload.</param>
/// <returns>True if success.</returns>
public async Task<bool> UploadFileAsync(string bucketName, string objectName,
string filePath)
{
    var request = new PutObjectRequest
    {
        BucketName = bucketName,
        Key = objectName,
        FilePath = filePath,
        ChecksumAlgorithm = ChecksumAlgorithm.SHA256
    };

    var response = await _amazonS3.PutObjectAsync(request);
    if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
    {
        Console.WriteLine($"\\tSuccessfully uploaded {objectName} to
{bucketName}.");
        return true;
    }
}
```

```
    }
    else
    {
        Console.WriteLine($"\\tCould not upload {objectName} to
{bucketName}.");
        return false;
    }
}

/// <summary>
/// List bucket objects and versions.
/// </summary>
/// <param name="bucketName">The Amazon S3 bucket to use.</param>
/// <returns>The list of objects and versions.</returns>
public async Task<ListVersionsResponse> ListBucketObjectsAndVersions(string
bucketName)
{
    var request = new ListVersionsRequest()
    {
        BucketName = bucketName
    };

    var response = await _amazonS3.ListVersionsAsync(request);
    return response;
}

/// <summary>
/// Delete an object from a specific bucket.
/// </summary>
/// <param name="bucketName">The Amazon S3 bucket to use.</param>
/// <param name="objectKey">The key of the object to delete.</param>
/// <param name="hasRetention">True if the object has retention settings.</
param>
/// <param name="versionId">Optional versionId.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DeleteObjectFromBucket(string bucketName, string
objectKey, bool hasRetention, string? versionId = null)
{
    try
    {
        var request = new DeleteObjectRequest()
        {
            BucketName = bucketName,
            Key = objectKey,
```

```
        VersionId = versionId,
    };
    if (hasRetention)
    {
        // Set the BypassGovernanceRetention header
        // if the file has retention settings.
        request.BypassGovernanceRetention = true;
    }
    await _amazonS3.DeleteObjectAsync(request);
    Console.WriteLine(
        $"Deleted {objectKey} in {bucketName}.");
    return true;
}
catch (AmazonS3Exception ex)
{
    Console.WriteLine($"Unable to delete object {objectKey} in bucket
{bucketName}: " + ex.Message);
    return false;
}
}

/// <summary>
/// Delete a specific bucket.
/// </summary>
/// <param name="bucketName">The Amazon S3 bucket to use.</param>
/// <param name="objectKey">The key of the object to delete.</param>
/// <param name="versionId">Optional versionId.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DeleteBucketByName(string bucketName)
{
    try
    {
        var request = new DeleteBucketRequest() { BucketName = bucketName, };
        var response = await _amazonS3.DeleteBucketAsync(request);
        Console.WriteLine($"Delete for {bucketName} complete.");
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Unable to delete bucket {bucketName}: " +
ex.Message);
        return false;
    }
}
```

```
}  
  
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for .NET.
 - [GetObjectLegalHold](#)
 - [GetObjectLockConfiguration](#)
 - [GetObjectRetention](#)
 - [PutObjectLegalHold](#)
 - [PutObjectLockConfiguration](#)
 - [PutObjectRetention](#)

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Execute um cenário interativo demonstrando os recursos de bloqueio de objetos do Amazon S3.

```
import software.amazon.awssdk.services.s3.model.ObjectLockLegalHold;  
import software.amazon.awssdk.services.s3.model.ObjectLockRetention;  
import java.io.BufferedWriter;  
import java.io.IOException;  
import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
import java.util.stream.Collectors;  
  
/*
```

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/setup.html>

This Java example performs the following tasks:

1. Create test Amazon Simple Storage Service (S3) buckets with different lock policies.
2. Upload sample objects to each bucket.
3. Set some Legal Hold and Retention Periods on objects and buckets.
4. Investigate lock policies by viewing settings or attempting to delete or overwrite objects.
5. Clean up objects and buckets.

```
*/
public class S3ObjectLockWorkflow {

    public static final String DASHES = new String(new char[80]).replace("\0",
    "-");
    static String bucketName;
    static S3LockActions s3LockActions;
    private static final List<String> bucketNames = new ArrayList<>();
    private static final List<String> fileNames = new ArrayList<>();

    public static void main(String[] args) {
        // Get the current date and time to ensure bucket name is unique.
        LocalDateTime currentTime = LocalDateTime.now();

        // Format the date and time as a string.
        DateTimeFormatter formatter =
        DateTimeFormatter.ofPattern("yyyyMMddHHmmss");
        String timeStamp = currentTime.format(formatter);

        s3LockActions = new S3LockActions();
        bucketName = "bucket"+timeStamp;
        Scanner scanner = new Scanner(System.in);

        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon Simple Storage Service (S3)
        Object Locking Workflow Scenario.");
        System.out.println("Press Enter to continue...");
        scanner.nextLine();
        configurationSetup();
        System.out.println(DASHES);
    }
}
```

```
System.out.println(DASHES);
setup();
System.out.println("Setup is complete. Press Enter to continue...");
scanner.nextLine();
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("Lets present the user with choices.");
System.out.println("Press Enter to continue...");
scanner.nextLine();
demoActionChoices() ;
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("Would you like to clean up the resources? (y/n)");
String delAns = scanner.nextLine().trim();
if (delAns.equalsIgnoreCase("y")) {
    cleanup();
    System.out.println("Clean up is complete.");
}

System.out.println("Press Enter to continue...");
scanner.nextLine();
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("Amazon S3 Object Locking Workflow is complete.");
System.out.println(DASHES);
}

// Present the user with the demo action choices.
public static void demoActionChoices() {
    String[] choices = {
        "List all files in buckets.",
        "Attempt to delete a file.",
        "Attempt to delete a file with retention period bypass.",
        "Attempt to overwrite a file.",
        "View the object and bucket retention settings for a file.",
        "View the legal hold settings for a file.",
        "Finish the workflow."
    };
};

int choice = 0;
```

```
while (true) {
    System.out.println(DASHES);
    choice = getChoiceResponse("Explore the S3 locking features by
selecting one of the following choices:", choices);
    System.out.println(DASHES);
    System.out.println("You selected "+choices[choice]);
    switch (choice) {
        case 0 -> {
            s3LockActions.listBucketsAndObjects(bucketNames, true);
        }

        case 1 -> {
            System.out.println("Enter the number of the object to
delete:");

            List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, true);
            List<String> fileKeys = allFiles.stream().map(f ->
f.getKeyName()).collect(Collectors.toList());
            String[] fileKeysArray = fileKeys.toArray(new String[0]);
            int fileChoice = getChoiceResponse(null, fileKeysArray);
            String objectKey = fileKeys.get(fileChoice);
            String bucketName = allFiles.get(fileChoice).getBucketName();
            String version = allFiles.get(fileChoice).getVersion();
            s3LockActions.deleteObjectFromBucket(bucketName, objectKey,
false, version);
        }

        case 2 -> {
            System.out.println("Enter the number of the object to
delete:");

            List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, true);
            List<String> fileKeys = allFiles.stream().map(f ->
f.getKeyName()).collect(Collectors.toList());
            String[] fileKeysArray = fileKeys.toArray(new String[0]);
            int fileChoice = getChoiceResponse(null, fileKeysArray);
            String objectKey = fileKeys.get(fileChoice);
            String bucketName = allFiles.get(fileChoice).getBucketName();
            String version = allFiles.get(fileChoice).getVersion();
            s3LockActions.deleteObjectFromBucket(bucketName, objectKey,
true, version);
        }

        case 3 -> {
```



```
        System.out.println("Enter the number of the object to
overwrite:");
        List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, true);
        List<String> fileKeys = allFiles.stream().map(f ->
f.getKeyName()).collect(Collectors.toList());
        String[] fileKeysArray = fileKeys.toArray(new String[0]);
        int fileChoice = getChoiceResponse(null, fileKeysArray);
        String objectKey = fileKeys.get(fileChoice);
        String bucketName = allFiles.get(fileChoice).getBucketName();

        // Attempt to overwrite the file.
        try (BufferedWriter writer = new BufferedWriter(new
java.io.FileWriter(objectKey))) {
            writer.write("This is a modified text.");

        } catch (IOException e) {
            e.printStackTrace();
        }
        s3LockActions.uploadFile(bucketName, objectKey, objectKey);
    }

    case 4 -> {
        System.out.println("Enter the number of the object to
overwrite:");
        List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, true);
        List<String> fileKeys = allFiles.stream().map(f ->
f.getKeyName()).collect(Collectors.toList());
        String[] fileKeysArray = fileKeys.toArray(new String[0]);
        int fileChoice = getChoiceResponse(null, fileKeysArray);
        String objectKey = fileKeys.get(fileChoice);
        String bucketName = allFiles.get(fileChoice).getBucketName();
        s3LockActions.getObjectRetention(bucketName, objectKey);
    }

    case 5 -> {
        System.out.println("Enter the number of the object to
view:");
        List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, true);
        List<String> fileKeys = allFiles.stream().map(f ->
f.getKeyName()).collect(Collectors.toList());
        String[] fileKeysArray = fileKeys.toArray(new String[0]);
```

```
        int fileChoice = getChoiceResponse(null, fileKeysArray);
        String objectKey = fileKeys.get(fileChoice);
        String bucketName = allFiles.get(fileChoice).getBucketName();
        s3LockActions.getObjectLegalHold(bucketName, objectKey);
        s3LockActions.getBucketObjectLockConfiguration(bucketName);
    }

    case 6 -> {
        System.out.println("Exiting the workflow...");
        return;
    }

    default -> {
        System.out.println("Invalid choice. Please select again.");
    }
}

}

// Clean up the resources from the scenario.
private static void cleanup() {
    List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, false);
    for (S3InfoObject fileInfo : allFiles) {
        String bucketName = fileInfo.getBucketName();
        String key = fileInfo.getKeyName();
        String version = fileInfo.getVersion();
        if (bucketName.contains("lock-enabled") ||
(bucketName.contains("retention-after-creation"))) {
            ObjectLockLegalHold legalHold =
s3LockActions.getObjectLegalHold(bucketName, key);
            if (legalHold != null) {
                String holdStatus = legalHold.status().name();
                System.out.println(holdStatus);
                if (holdStatus.compareTo("ON") == 0) {
                    s3LockActions.modifyObjectLegalHold(bucketName, key,
false);
                }
            }
        }
        // Check for a retention period.
        ObjectLockRetention retention =
s3LockActions.getObjectRetention(bucketName, key);
        boolean hasRetentionPeriod ;
        hasRetentionPeriod = retention != null;
    }
}
```

```
        s3LockActions.deleteObjectFromBucket(bucketName,
key,hasRetentionPeriod, version);

        } else {
            System.out.println(bucketName +" objects do not have a legal
lock");
            s3LockActions.deleteObjectFromBucket(bucketName, key,false,
version);
        }
    }

    // Delete the buckets.
    System.out.println("Delete "+bucketName);
    for (String bucket : bucketNames){
        s3LockActions.deleteBucketByName(bucket);
    }
}

private static void setup() {
    Scanner scanner = new Scanner(System.in);
    System.out.println("""
        For this workflow, we will use the AWS SDK for Java to create
several S3
        buckets and files to demonstrate working with S3 locking
features.
        """);

    System.out.println("S3 buckets can be created either with or without
object lock enabled.");
    System.out.println("Press Enter to continue...");
    scanner.nextLine();

    // Create three S3 buckets.
    s3LockActions.createBucketWithLockOptions(false, bucketNames.get(0));
    s3LockActions.createBucketWithLockOptions(true, bucketNames.get(1));
    s3LockActions.createBucketWithLockOptions(false, bucketNames.get(2));
    System.out.println("Press Enter to continue.");
    scanner.nextLine();

    System.out.println("Bucket "+bucketNames.get(2) +" will be configured to
use object locking with a default retention period.");
    s3LockActions.modifyBucketDefaultRetention(bucketNames.get(2));
    System.out.println("Press Enter to continue.");
    scanner.nextLine();
}
```

```
        System.out.println("Object lock policies can also be added to existing
buckets. For this example, we will use "+bucketNames.get(1));
        s3LockActions.enableObjectLockOnBucket(bucketNames.get(1));
        System.out.println("Press Enter to continue.");
        scanner.nextLine();

        // Upload some files to the buckets.
        System.out.println("Now let's add some test files:");
        String fileName = "exampleFile.txt";
        int fileCount = 2;
        try (BufferedWriter writer = new BufferedWriter(new
java.io.FileWriter(fileName))) {
            writer.write("This is a sample file for uploading to a bucket.");

        } catch (IOException e) {
            e.printStackTrace();
        }

        for (String bucketName : bucketNames){
            for (int i = 0; i < fileCount; i++) {
                // Get the file name without extension.
                String fileNameWithoutExtension =
java.nio.file.Paths.get(fileName).getFileName().toString();
                int extensionIndex = fileNameWithoutExtension.lastIndexOf('.');
                if (extensionIndex > 0) {
                    fileNameWithoutExtension =
fileNameWithoutExtension.substring(0, extensionIndex);
                }

                // Create the numbered file names.
                String numberedFileName = fileNameWithoutExtension + i +
getFileExtension(fileName);
                fileNames.add(numberedFileName);
                s3LockActions.uploadFile(bucketName, numberedFileName, fileName);
            }
        }

        String question = null;
        System.out.print("Press Enter to continue...");
        scanner.nextLine();
        System.out.println("Now we can set some object lock policies on
individual files:");
        for (String bucketName : bucketNames) {
```

```

        for (int i = 0; i < fileNames.size(); i++){

            // No modifications to the objects in the first bucket.
            if (!bucketName.equals(bucketNames.get(0))) {
                String exampleFileName = fileNames.get(i);
                switch (i) {
                    case 0 -> {
                        question = "Would you like to add a legal hold to " +
exampleFileName + " in " + bucketName + " (y/n)?";
                        System.out.println(question);
                        String ans = scanner.nextLine().trim();
                        if (ans.equalsIgnoreCase("y")) {
                            System.out.println("**** You have selected to put
a legal hold " + exampleFileName);

                                // Set a legal hold.
                                s3LockActions.modifyObjectLegalHold(bucketName,
exampleFileName, true);
                            }
                        }
                    case 1 -> {
                        """"
                            Would you like to add a 1 day Governance
retention period to %s in %s (y/n)?
                            Reminder: Only a user with the
s3:BypassGovernanceRetention permission will be able to delete this file or its
bucket until the retention period has expired.
                            """".formatted(exampleFileName, bucketName);
                        System.out.println(question);
                        String ans2 = scanner.nextLine().trim();
                        if (ans2.equalsIgnoreCase("y")) {

s3LockActions.modifyObjectRetentionPeriod(bucketName, exampleFileName);
                            }
                        }
                    }
                }
            }
        }

        // Get file extension.
        private static String getFileExtension(String fileName) {
            int dotIndex = fileName.lastIndexOf('.');

```

```
        if (dotIndex > 0) {
            return fileName.substring(dotIndex);
        }
        return "";
    }

    public static void configurationSetup() {
        String noLockBucketName = bucketName + "-no-lock";
        String lockEnabledBucketName = bucketName + "-lock-enabled";
        String retentionAfterCreationBucketName = bucketName + "-retention-after-creation";
        bucketNames.add(noLockBucketName);
        bucketNames.add(lockEnabledBucketName);
        bucketNames.add(retentionAfterCreationBucketName);
    }

    public static int getChoiceResponse(String question, String[] choices) {
        Scanner scanner = new Scanner(System.in);
        if (question != null) {
            System.out.println(question);
            for (int i = 0; i < choices.length; i++) {
                System.out.println("\t" + (i + 1) + ". " + choices[i]);
            }
        }

        int choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > choices.length) {
            String choice = scanner.nextLine();
            try {
                choiceNumber = Integer.parseInt(choice);
            } catch (NumberFormatException e) {
                System.out.println("Invalid choice. Please enter a valid number.");
            }
        }

        return choiceNumber - 1;
    }
}
```

Uma classe de wrapper para funções do S3.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.BucketVersioningStatus;
import software.amazon.awssdk.services.s3.model.ChecksumAlgorithm;
import software.amazon.awssdk.services.s3.model.CreateBucketRequest;
import software.amazon.awssdk.services.s3.model.DefaultRetention;
import software.amazon.awssdk.services.s3.model.DeleteBucketRequest;
import software.amazon.awssdk.services.s3.model.DeleteObjectRequest;
import software.amazon.awssdk.services.s3.model.GetObjectLegalHoldRequest;
import software.amazon.awssdk.services.s3.model.GetObjectLegalHoldResponse;
import
    software.amazon.awssdk.services.s3.model.GetObjectLockConfigurationRequest;
import
    software.amazon.awssdk.services.s3.model.GetObjectLockConfigurationResponse;
import software.amazon.awssdk.services.s3.model.GetObjectRetentionRequest;
import software.amazon.awssdk.services.s3.model.GetObjectRetentionResponse;
import software.amazon.awssdk.services.s3.model.HeadBucketRequest;
import software.amazon.awssdk.services.s3.model.ListObjectVersionsRequest;
import software.amazon.awssdk.services.s3.model.ListObjectVersionsResponse;
import software.amazon.awssdk.services.s3.model.MFADelete;
import software.amazon.awssdk.services.s3.model.ObjectLockConfiguration;
import software.amazon.awssdk.services.s3.model.ObjectLockEnabled;
import software.amazon.awssdk.services.s3.model.ObjectLockLegalHold;
import software.amazon.awssdk.services.s3.model.ObjectLockLegalHoldStatus;
import software.amazon.awssdk.services.s3.model.ObjectLockRetention;
import software.amazon.awssdk.services.s3.model.ObjectLockRetentionMode;
import software.amazon.awssdk.services.s3.model.ObjectLockRule;
import software.amazon.awssdk.services.s3.model.PutBucketVersioningRequest;
import software.amazon.awssdk.services.s3.model.PutObjectLegalHoldRequest;
import
    software.amazon.awssdk.services.s3.model.PutObjectLockConfigurationRequest;
import software.amazon.awssdk.services.s3.model.PutObjectRequest;
import software.amazon.awssdk.services.s3.model.PutObjectResponse;
import software.amazon.awssdk.services.s3.model.PutObjectRetentionRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.VersioningConfiguration;
import software.amazon.awssdk.services.s3.waiters.S3Waiter;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.time.Instant;
import java.time.ZoneId;
import java.time.ZonedDateTime;
import java.time.format.DateTimeFormatter;
```

```
import java.time.temporal.ChronoUnit;
import java.util.List;
import java.util.concurrent.atomic.AtomicInteger;
import java.util.stream.Collectors;

// Contains application logic for the Amazon S3 operations used in this workflow.
public class S3LockActions {

    private static S3Client getClient() {
        return S3Client.builder()
            .region(Region.US_EAST_1)
            .build();
    }

    // Set or modify a retention period on an object in an S3 bucket.
    public void modifyObjectRetentionPeriod(String bucketName, String objectKey)
    {
        // Calculate the instant one day from now.
        Instant futureInstant = Instant.now().plus(1, ChronoUnit.DAYS);

        // Convert the Instant to a ZonedDateTime object with a specific time
        zone.
        ZonedDateTime zonedDateTime =
        futureInstant.atZone(ZoneId.systemDefault());

        // Define a formatter for human-readable output.
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd
        HH:mm:ss");

        // Format the ZonedDateTime object to a human-readable date string.
        String humanReadableDate = formatter.format(zonedDateTime);

        // Print the formatted date string.
        System.out.println("Formatted Date: " + humanReadableDate);
        ObjectLockRetention retention = ObjectLockRetention.builder()
            .mode(ObjectLockRetentionMode.GOVERNANCE)
            .retainUntilDate(futureInstant)
            .build();

        PutObjectRetentionRequest retentionRequest =
        PutObjectRetentionRequest.builder()
            .bucket(bucketName)
            .key(objectKey)
            .retention(retention)
```



```
        .build();

        getClient().putObjectRetention(retentionRequest);
        System.out.println("Set retention for "+objectKey +" in " +bucketName +"
until "+ humanReadableDate +".");
    }

    // Get the legal hold details for an S3 object.
    public ObjectLockLegalHold getObjectLegalHold(String bucketName, String
objectKey) {
        try {
            GetObjectLegalHoldRequest legalHoldRequest =
GetObjectLegalHoldRequest.builder()
                .bucket(bucketName)
                .key(objectKey)
                .build();

            GetObjectLegalHoldResponse response =
getClient().getObjectLegalHold(legalHoldRequest);
            System.out.println("Object legal hold for " + objectKey + " in " +
bucketName +
                ":\n\tStatus: " + response.legalHold().status());
            return response.legalHold();

        } catch (S3Exception ex) {
            System.out.println("\tUnable to fetch legal hold: '" +
ex.getMessage() + "'");
        }

        return null;
    }

    // Create a new Amazon S3 bucket with object lock options.
    public void createBucketWithLockOptions(boolean enableObjectLock, String
bucketName) {
        S3Waiter s3Waiter = getClient().waiter();
        CreateBucketRequest bucketRequest = CreateBucketRequest.builder()
            .bucket(bucketName)
            .objectLockEnabledForBucket(enableObjectLock)
            .build();

        getClient().createBucket(bucketRequest);
        HeadBucketRequest bucketRequestWait = HeadBucketRequest.builder()
            .bucket(bucketName)
```

```
        .build());

        // Wait until the bucket is created and print out the response.
        s3Waiter.waitUntilBucketExists(bucketRequestWait);
        System.out.println(bucketName + " is ready");
    }

    public List<S3InfoObject> listBucketsAndObjects(List<String> bucketNames,
        Boolean interactive) {
        AtomicInteger counter = new AtomicInteger(0); // Initialize counter.
        return bucketNames.stream()
            .flatMap(bucketName ->
                listBucketObjectsAndVersions(bucketName).versions().stream()
                    .map(version -> {
                        S3InfoObject s3InfoObject = new S3InfoObject();
                        s3InfoObject.setBucketName(bucketName);
                        s3InfoObject.setVersion(version.versionId());
                        s3InfoObject.setKeyName(version.key());
                        return s3InfoObject;
                    })))
            .peek(s3InfoObject -> {
                int i = counter.incrementAndGet(); // Increment and get the
                updated value.
                if (interactive) {
                    System.out.println(i + ": " + s3InfoObject.getKeyName());
                    System.out.printf("%5s Bucket name: %s\n", "",
                        s3InfoObject.getBucketName());
                    System.out.printf("%5s Version: %s\n", "",
                        s3InfoObject.getVersion());
                }
            })
            .collect(Collectors.toList());
    }

    public ListObjectVersionsResponse listBucketObjectsAndVersions(String
        bucketName) {
        ListObjectVersionsRequest versionsRequest =
        ListObjectVersionsRequest.builder()
            .bucket(bucketName)
            .build();

        return getClient().listObjectVersions(versionsRequest);
    }
}
```

```
// Set or modify a retention period on an S3 bucket.
public void modifyBucketDefaultRetention(String bucketName) {
    VersioningConfiguration versioningConfiguration =
VersioningConfiguration.builder()
        .mfaDelete(MFADelete.DISABLED)
        .status(BucketVersioningStatus.ENABLED)
        .build();

    PutBucketVersioningRequest versioningRequest =
PutBucketVersioningRequest.builder()
        .bucket(bucketName)
        .versioningConfiguration(versioningConfiguration)
        .build();

    getClient().putBucketVersioning(versioningRequest);
    DefaultRetention retention = DefaultRetention.builder()
        .days(1)
        .mode(ObjectLockRetentionMode.GOVERNANCE)
        .build();

    ObjectLockRule lockRule = ObjectLockRule.builder()
        .defaultRetention(retention)
        .build();

    ObjectLockConfiguration objectLockConfiguration =
ObjectLockConfiguration.builder()
        .objectLockEnabled(ObjectLockEnabled.ENABLED)
        .rule(lockRule)
        .build();

    PutObjectLockConfigurationRequest putObjectLockConfigurationRequest =
PutObjectLockConfigurationRequest.builder()
        .bucket(bucketName)
        .objectLockConfiguration(objectLockConfiguration)
        .build();

    getClient().putObjectLockConfiguration(putObjectLockConfigurationRequest) ;
    System.out.println("Added a default retention to bucket "+bucketName
+".");
}

// Enable object lock on an existing bucket.
public void enableObjectLockOnBucket(String bucketName) {
```

```
    try {
        VersioningConfiguration versioningConfiguration =
VersioningConfiguration.builder()
            .status(BucketVersioningStatus.ENABLED)
            .build();

        PutBucketVersioningRequest putBucketVersioningRequest =
PutBucketVersioningRequest.builder()
            .bucket(bucketName)
            .versioningConfiguration(versioningConfiguration)
            .build();

        // Enable versioning on the bucket.
        getClient().putBucketVersioning(putBucketVersioningRequest);
        PutObjectLockConfigurationRequest request =
PutObjectLockConfigurationRequest.builder()
            .bucket(bucketName)
            .objectLockConfiguration(ObjectLockConfiguration.builder()
                .objectLockEnabled(ObjectLockEnabled.ENABLED)
                .build())
            .build();

        getClient().putObjectLockConfiguration(request);
        System.out.println("Successfully enabled object lock on
"+bucketName);

    } catch (S3Exception ex) {
        System.out.println("Error modifying object lock: '" + ex.getMessage()
+ "'");
    }
}

public void uploadFile(String bucketName, String objectName, String filePath)
{
    Path file = Paths.get(filePath);
    PutObjectRequest request = PutObjectRequest.builder()
        .bucket(bucketName)
        .key(objectName)
        .checksumAlgorithm(ChecksumAlgorithm.SHA256)
        .build();

    PutObjectResponse response = getClient().putObject(request, file);
    if (response != null) {
```

```
        System.out.println("\tSuccessfully uploaded " + objectName + " to " +
bucketName + ".");
    } else {
        System.out.println("\tCould not upload " + objectName + " to " +
bucketName + ".");
    }
}

// Set or modify a legal hold on an object in an S3 bucket.
public void modifyObjectLegalHold(String bucketName, String objectKey,
boolean legalHoldOn) {
    ObjectLockLegalHold legalHold ;
    if (legalHoldOn) {
        legalHold = ObjectLockLegalHold.builder()
            .status(ObjectLockLegalHoldStatus.ON)
            .build();
    } else {
        legalHold = ObjectLockLegalHold.builder()
            .status(ObjectLockLegalHoldStatus.OFF)
            .build();
    }

    PutObjectLegalHoldRequest legalHoldRequest =
PutObjectLegalHoldRequest.builder()
        .bucket(bucketName)
        .key(objectKey)
        .legalHold(legalHold)
        .build();

    getClient().putObjectLegalHold(legalHoldRequest) ;
    System.out.println("Modified legal hold for "+ objectKey +" in
"+bucketName +".");
}

// Delete an object from a specific bucket.
public void deleteObjectFromBucket(String bucketName, String objectKey,
boolean hasRetention, String versionId) {
    try {
        DeleteObjectRequest objectRequest;
        if (hasRetention) {
            objectRequest = DeleteObjectRequest.builder()
                .bucket(bucketName)
                .key(objectKey)
                .versionId(versionId)
```

```
        .bypassGovernanceRetention(true)
        .build();
    } else {
        objectRequest = DeleteObjectRequest.builder()
            .bucket(bucketName)
            .key(objectKey)
            .versionId(versionId)
            .build();
    }

    getClient().deleteObject(objectRequest) ;
    System.out.println("The object was successfully deleted");

} catch (S3Exception e) {
    System.err.println(e.awsErrorDetails().errorMessage());
}
}

// Get the retention period for an S3 object.
public ObjectLockRetention getObjectRetention(String bucketName, String key){
    try {
        GetObjectRetentionRequest retentionRequest =
GetObjectRetentionRequest.builder()
            .bucket(bucketName)
            .key(key)
            .build();

        GetObjectRetentionResponse response =
getClient().getObjectRetention(retentionRequest);
        System.out.println("tObject retention for "+key +"
in "+ bucketName +": " + response.retention().mode() +" until "+
response.retention().retainUntilDate() +".");
        return response.retention();

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        return null;
    }
}

public void deleteBucketByName(String bucketName) {
    try {
        DeleteBucketRequest request = DeleteBucketRequest.builder()
            .bucket(bucketName)
```

```
        .build();

        getClient().deleteBucket(request);
        System.out.println(bucketName + " was deleted.");

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
    }
}

// Get the object lock configuration details for an S3 bucket.
public void getBucketObjectLockConfiguration(String bucketName) {
    GetObjectLockConfigurationRequest objectLockConfigurationRequest =
GetObjectLockConfigurationRequest.builder()
        .bucket(bucketName)
        .build();

    GetObjectLockConfigurationResponse response =
getClient().getObjectLockConfiguration(objectLockConfigurationRequest);
    System.out.println("Bucket object lock config for "+bucketName +": ");
    System.out.println("\tEnabled:
"+response.getObjectLockConfiguration().objectLockEnabled());
    System.out.println("\tRule: "+
response.getObjectLockConfiguration().rule().defaultRetention());
}
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for Java 2.x.
 - [GetObjectLegalHold](#)
 - [GetObjectLockConfiguration](#)
 - [GetObjectRetention](#)
 - [PutObjectLegalHold](#)
 - [PutObjectLockConfiguration](#)
 - [PutObjectRetention](#)

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Gerenciar listas de controle de acesso (ACLs) para buckets do Amazon S3 usando um AWS SDK

Os exemplos de código a seguir mostram como gerenciar listas de controle de acesso (ACLs) para buckets do Amazon S3.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to manage Amazon Simple Storage Service
/// (Amazon S3) access control lists (ACLs) to control Amazon S3 bucket
/// access.
/// </summary>
public class ManageACLs
{
    public static async Task Main()
    {
        string bucketName = "doc-example-bucket1";
        string newBucketName = "doc-example-bucket2";
        string keyName = "sample-object.txt";
        string emailAddress = "someone@example.com";

        // If the AWS Region where your bucket is located is different from
        // the Region defined for the default user, pass the Amazon S3
        bucket's
        // name to the client constructor. It should look like this:
        // RegionEndpoint bucketRegion = RegionEndpoint.USEast1;
```



```

        IAmazonS3 client = new AmazonS3Client();

        await TestBucketObjectACLsAsync(client, bucketName, newBucketName,
keyName, emailAddress);
    }

    /// <summary>
    /// Creates a new Amazon S3 bucket with a canned ACL, then retrieves the
ACL
    /// information and then adds a new ACL to one of the objects in the
    /// Amazon S3 bucket.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// methods to create a bucket, get an ACL, and add a different ACL to
    /// one of the objects.</param>
    /// <param name="bucketName">A string representing the original Amazon S3
    /// bucket name.</param>
    /// <param name="newBucketName">A string representing the name of the
    /// new bucket that will be created.</param>
    /// <param name="keyName">A string representing the key name of an Amazon
S3
    /// object for which we will change the ACL.</param>
    /// <param name="emailAddress">A string representing the email address
    /// belonging to the person to whom access to the Amazon S3 bucket will
be
    /// granted.</param>
    public static async Task TestBucketObjectACLsAsync(
        IAmazonS3 client,
        string bucketName,
        string newBucketName,
        string keyName,
        string emailAddress)
    {
        try
        {
            // Create a new Amazon S3 bucket and specify canned ACL.
            var success = await CreateBucketWithCannedACLAsync(client,
newBucketName);

            // Get the ACL on a bucket.
            await GetBucketACLAsync(client, bucketName);

            // Add (replace) the ACL on an object in a bucket.

```

```

        await AddACLToExistingObjectAsync(client, bucketName, keyName,
        emailAddress);
    }
    catch (AmazonS3Exception amazonS3Exception)
    {
        Console.WriteLine($"Exception: {amazonS3Exception.Message}");
    }
}

/// <summary>
/// Creates a new Amazon S3 bucket with a canned ACL attached.
/// </summary>
/// <param name="client">The initialized client object used to call
/// PutBucketAsync.</param>
/// <param name="newBucketName">A string representing the name of the
/// new Amazon S3 bucket.</param>
/// <returns>Returns a boolean value indicating success or failure.</
returns>
public static async Task<bool> CreateBucketWithCannedACLAsync(IAmazonS3
client, string newBucketName)
{
    var request = new PutBucketRequest()
    {
        BucketName = newBucketName,
        BucketRegion = S3Region.EUWest1,

        // Add a canned ACL.
        CannedACL = S3CannedACL.LogDeliveryWrite,
    };

    var response = await client.PutBucketAsync(request);
    return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
}

/// <summary>
/// Retrieves the ACL associated with the Amazon S3 bucket name in the
/// bucketName parameter.
/// </summary>
/// <param name="client">The initialized client object used to call
/// PutBucketAsync.</param>
/// <param name="bucketName">The Amazon S3 bucket for which we want to
get the
/// ACL list.</param>

```

```

    /// <returns>Returns an S3AccessControlList returned from the call to
    /// GetACLAsync.</returns>
    public static async Task<S3AccessControlList> GetBucketACLAsync(IAmazonS3
client, string bucketName)
    {
        GetACLResponse response = await client.GetACLAsync(new GetACLRequest
        {
            BucketName = bucketName,
        });

        return response.AccessControlList;
    }

    /// <summary>
    /// Adds a new ACL to an existing object in the Amazon S3 bucket.
    /// </summary>
    /// <param name="client">The initialized client object used to call
    /// PutBucketAsync.</param>
    /// <param name="bucketName">A string representing the name of the Amazon
S3
param>
    /// bucket containing the object to which we want to apply a new ACL.</
param>
    /// <param name="keyName">A string representing the name of the object
    /// to which we want to apply the new ACL.</param>
    /// <param name="emailAddress">The email address of the person to whom
    /// we will be applying to whom access will be granted.</param>
    public static async Task AddACLToExistingObjectAsync(IAmazonS3 client,
string bucketName, string keyName, string emailAddress)
    {
        // Retrieve the ACL for an object.
        GetACLResponse aclResponse = await client.GetACLAsync(new
GetACLRequest
        {
            BucketName = bucketName,
            Key = keyName,
        });

        S3AccessControlList acl = aclResponse.AccessControlList;

        // Retrieve the owner.
        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).
var fullControlGrant = new S3Grant
{
    Grantee = new S3Grantee { CanonicalUser = acl.Owner.Id },
};
acl.AddGrant(fullControlGrant.Grantee, S3Permission.FULL_CONTROL);

// Specify email to identify grantee for granting permissions.
var grantUsingEmail = new S3Grant
{
    Grantee = new S3Grantee { EmailAddress = emailAddress },
    Permission = S3Permission.WRITE_ACP,
};

// Specify log delivery group as grantee.
var grantLogDeliveryGroup = new S3Grant
{
    Grantee = new S3Grantee { URI = "http://acs.amazonaws.com/groups/
s3/LogDelivery" },
    Permission = S3Permission.WRITE,
};

// Create a new ACL.
var newAcl = new S3AccessControlList
{
    Grants = new List<S3Grant> { grantUsingEmail,
grantLogDeliveryGroup },
    Owner = owner,
};

// Set the new ACL. We're throwing away the response here.
_ = await client.PutACLAsync(new PutACLRequest
{
    BucketName = bucketName,
    Key = keyName,
    AccessControlList = newAcl,
});
}

}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for .NET.
 - [GetBucketAcl](#)
 - [GetObjectAcl](#)
 - [PutBucketAcl](#)
 - [PutObjectAcl](#)

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). 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.
- Criar 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 AWS SDK e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Analisar URIs do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como usar analisar URIs do Amazon S3 para extrair componentes importantes como o nome do bucket e a chave do objeto.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Analise um URI do Amazon S3 usando a classe [S3Uri](#).

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.S3Uri;
import software.amazon.awssdk.services.s3.S3Utilities;

import java.net.URI;
import java.util.List;
import java.util.Map;

/**
```

```
*
* @param s3Client - An S3Client through which you acquire an S3Uri
instance.
* @param s3objectUrl - A complex URL (String) that is used to demonstrate
S3Uri
* capabilities.
*/
public static void parseS3UriExample(S3Client s3Client, String s3objectUrl) {
    logger.info(s3objectUrl);
    // Console output:
    // 'https://s3.us-west-1.amazonaws.com/myBucket/resources/doc.txt?
versionId=abc123&partNumber=77&partNumber=88'.

    // Create an S3Utilities object using the configuration of the s3Client.
    S3Utilities s3Utilities = s3Client.utilities();

    // From a String URL create a URI object to pass to the parseUri()
method.
    URI uri = URI.create(s3objectUrl);
    S3Uri s3Uri = s3Utilities.parseUri(uri);

    // If the URI contains no value for the Region, bucket or key, the SDK
returns
    // an empty Optional.
    // The SDK returns decoded URI values.

    Region region = s3Uri.region().orElse(null);
    log("region", region);
    // Console output: 'region: us-west-1'.

    String bucket = s3Uri.bucket().orElse(null);
    log("bucket", bucket);
    // Console output: 'bucket: myBucket'.

    String key = s3Uri.key().orElse(null);
    log("key", key);
    // Console output: 'key: resources/doc.txt'.

    Boolean isPathStyle = s3Uri.isPathStyle();
    log("isPathStyle", isPathStyle);
    // Console output: 'isPathStyle: true'.

    // If the URI contains no query parameters, the SDK returns an empty map.
    Map<String, List<String>> queryParams = s3Uri.rawQueryParameters();
```

```

    log("rawQueryParameters", queryParams);
    // Console output: 'rawQueryParameters: {versionId=[abc123],
partNumber=[77,
    // 88]}'.

    // Retrieve the first or all values for a query parameter as shown in the
    // following code.
    String versionId =
s3Uri.firstMatchingRawQueryParameter("versionId").orElse(null);
    log("firstMatchingRawQueryParameter-versionId", versionId);
    // Console output: 'firstMatchingRawQueryParameter-versionId: abc123'.

    String partNumber =
s3Uri.firstMatchingRawQueryParameter("partNumber").orElse(null);
    log("firstMatchingRawQueryParameter-partNumber", partNumber);
    // Console output: 'firstMatchingRawQueryParameter-partNumber: 77'.

    List<String> partNumbers =
s3Uri.firstMatchingRawQueryParameters("partNumber");
    log("firstMatchingRawQueryParameter", partNumbers);
    // Console output: 'firstMatchingRawQueryParameter: [77, 88]'.

    /*
    * Object keys and query parameters with reserved or unsafe characters,
must be
    * URL-encoded.
    * For example replace whitespace " " with "%20".
    * Valid:
    * "https://s3.us-west-1.amazonaws.com/myBucket/object%20key?query=
%5Bbrackets%5D"
    * Invalid:
    * "https://s3.us-west-1.amazonaws.com/myBucket/object key?
query=[brackets]"
    *
    * Virtual-hosted-style URIs with bucket names that contain a dot, ".",
the dot
    * must not be URL-encoded.
    * Valid: "https://my.Bucket.s3.us-west-1.amazonaws.com/key"
    * Invalid: "https://my%2EBucket.s3.us-west-1.amazonaws.com/key"
    */
}

private static void log(String s3UriElement, Object element) {
    if (element == null) {

```



```
        logger.info("{}: {}", s3UriElement, "null");
    } else {
        logger.info("{}: {}", s3UriElement, element);
    }
}
```

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Executar uma cópia multipart de um objeto do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como executar uma cópia multipart de um objeto do Amazon S3.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to perform a multi-part copy from one Amazon
/// Simple Storage Service (Amazon S3) bucket to another.
/// </summary>
public class MPUapiCopyObj
{
    private const string SourceBucket = "doc-example-bucket1";
```

```
private const string TargetBucket = "doc-example-bucket2";
private const string SourceObjectKey = "example.mov";
private const string TargetObjectKey = "copied_video_file.mov";

/// <summary>
/// This method starts the multi-part upload.
/// </summary>
public static async Task Main()
{
    var s3Client = new AmazonS3Client();
    Console.WriteLine("Copying object...");
    await MPUCopyObjectAsync(s3Client);
}

/// <summary>
/// This method uses the passed client object to perform a multipart
/// copy operation.
/// </summary>
/// <param name="client">An Amazon S3 client object that will be used
/// to perform the copy.</param>
public static async Task MPUCopyObjectAsync(AmazonS3Client client)
{
    // Create a list to store the copy part responses.
    var copyResponses = new List<CopyPartResponse>();

    // Setup information required to initiate the multipart upload.
    var initiateRequest = new InitiateMultipartUploadRequest
    {
        BucketName = TargetBucket,
        Key = TargetObjectKey,
    };

    // Initiate the upload.
    InitiateMultipartUploadResponse initResponse =
        await client.InitiateMultipartUploadAsync(initiateRequest);

    // Save the upload ID.
    string uploadId = initResponse.UploadId;

    try
    {
        // Get the size of the object.
        var metadataRequest = new GetObjectMetadataRequest
        {
```

```
        BucketName = SourceBucket,
        Key = SourceObjectKey,
    };

    GetObjectMetadataResponse metadataResponse =
        await client.GetObjectMetadataAsync(metadataRequest);
    var objectSize = metadataResponse.ContentLength; // Length in
bytes.

    // Copy the parts.
    var partSize = 5 * (long)Math.Pow(2, 20); // Part size is 5 MB.

    long bytePosition = 0;
    for (int i = 1; bytePosition < objectSize; i++)
    {
        var 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 client.CopyPartAsync(copyRequest));

        bytePosition += partSize;
    }

    // Set up to complete the copy.
    var completeRequest = new CompleteMultipartUploadRequest
    {
        BucketName = TargetBucket,
        Key = TargetObjectKey,
        UploadId = initResponse.UploadId,
    };
    completeRequest.AddPartETags(copyResponses);

    // Complete the copy.
    CompleteMultipartUploadResponse completeUploadResponse =
```

```
        await client.CompleteMultipartUploadAsync(completeRequest);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine($"Error encountered on server.
Message: '{e.Message}' when writing an object");
    }
    catch (Exception e)
    {
        Console.WriteLine($"Unknown encountered on server.
Message: '{e.Message}' when writing an object");
    }
}
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for .NET.
 - [CompleteMultipartUpload](#)
 - [CreateMultipartUpload](#)
 - [GetObjectMetadata](#)
 - [UploadPartCopy](#)

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Realizar um carregamento fragmentado em um objeto do Amazon S3 usando um SDK da AWS

O exemplo de código a seguir mostra como executar um carregamento fracionado de um objeto do Amazon S3.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Os exemplos de código usam as importações a seguir.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.exception.SdkException;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.services.s3.S3AsyncClient;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.CompletedMultipartUpload;
import software.amazon.awssdk.services.s3.model.CompletedPart;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.PutObjectResponse;
import software.amazon.awssdk.services.s3.model.UploadPartRequest;
import software.amazon.awssdk.services.s3.model.UploadPartResponse;
import software.amazon.awssdk.services.s3.waiters.S3Waiter;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.FileUpload;
import software.amazon.awssdk.transfer.s3.model.UploadFileRequest;

import java.io.IOException;
import java.io.RandomAccessFile;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.ByteBuffer;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.List;
import java.util.Objects;
import java.util.UUID;
import java.util.concurrent.CompletableFuture;
```

Use o [Gerenciador de Transferências do S3](#) na parte superior do [cliente do S3 baseado no AWS CRT](#) para realizar de forma transparente um carregamento fracionado quando o tamanho do conteúdo exceder um limite. O limite padrão é 8 MB.

```
public void multipartUploadWithTransferManager(String filePath) {
    S3TransferManager transferManager = S3TransferManager.create();
    UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
        .putObjectRequest(b -> b
            .bucket(bucketName)
            .key(key))
        .source(Paths.get(filePath))
        .build();
    FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);
    fileUpload.completionFuture().join();
    transferManager.close();
}
```

Use a [API S3Client](#) para realizar um carregamento fracionado.

```
public void multipartUploadWithS3Client(String filePath) {

    // Initiate the multipart upload.
    CreateMultipartUploadResponse createMultipartUploadResponse =
s3Client.createMultipartUpload(b -> b
        .bucket(bucketName)
        .key(key));
    String uploadId = createMultipartUploadResponse.uploadId();

    // Upload the parts of the file.
    int partNumber = 1;
    List<CompletedPart> completedParts = new ArrayList<>();
    ByteBuffer bb = ByteBuffer.allocate(1024 * 1024 * 5); // 5 MB byte buffer

    try (RandomAccessFile file = new RandomAccessFile(filePath, "r")) {
        long fileSize = file.length();
        int position = 0;
        while (position < fileSize) {
            file.seek(position);
            int read = file.getChannel().read(bb);

            bb.flip(); // Swap position and limit before reading from the
buffer.

```

```
UploadPartRequest uploadPartRequest = UploadPartRequest.builder()
    .bucket(bucketName)
    .key(key)
    .uploadId(uploadId)
    .partNumber(partNumber)
    .build();

UploadPartResponse partResponse = s3Client.uploadPart(
    uploadPartRequest,
    RequestBody.fromByteBuffer(bb));

CompletedPart part = CompletedPart.builder()
    .partNumber(partNumber)
    .eTag(partResponse.eTag())
    .build();
completedParts.add(part);

bb.clear();
position += read;
partNumber++;
}
} catch (IOException e) {
    logger.error(e.getMessage());
}

// Complete the multipart upload.
s3Client.completeMultipartUpload(b -> b
    .bucket(bucketName)
    .key(key)
    .uploadId(uploadId)
    .multipartUpload(CompletedMultipartUpload.builder().parts(completedParts).build()));
}
```

Use a [API S3AsyncClient](#) com o carregamento fracionado habilitado para realizar um carregamento fracionado.

```
public void multipartUploadWithS3AsyncClient(String filePath) {
    // Enable multipart support.
    S3AsyncClient s3AsyncClient = S3AsyncClient.builder()
        .multipartEnabled(true)
```

```
        .build();

        CompletableFuture<PutObjectResponse> response = s3AsyncClient.putObject(b
-> b
            .bucket(bucketName)
            .key(key),
            Paths.get(filePath));

        response.join();
        logger.info("File uploaded in multiple 8 MiB parts using
S3AsyncClient.");
    }
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for Java 2.x.
 - [CompleteMultipartUpload](#)
 - [CreateMultipartUpload](#)
 - [UploadPart](#)

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Monitorar o upload ou download de um objeto do Amazon S3 usando um SDK da AWS

O exemplo de código a seguir mostra como usar o Amazon S3 para carregar ou baixar arquivos grandes.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Acompanhe o andamento do carregamento de um arquivo.

```
public void trackUploadFile(S3TransferManager transferManager, String
bucketName,
                        String key, URI filePathURI) {
    UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
        .putObjectRequest(b -> b.bucket(bucketName).key(key))
        .addTransferListener(LoggingTransferListener.create()) // Add
listener.
        .source(Paths.get(filePathURI))
        .build();
```

```
FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);
```

```
fileUpload.completionFuture().join();
```

```
/*
```

The SDK provides a `LoggingTransferListener` implementation of the `TransferListener` interface.

You can also implement the interface to provide your own logic.

Configure `log4j2` with settings such as the following.

```
<Configuration status="WARN">
    <Appenders>
        <Console name="AlignedConsoleAppender"
target="SYSTEM_OUT">
            <PatternLayout pattern="%m%n"/>
        </Console>
    </Appenders>

    <Loggers>
        <logger
name="software.amazon.awssdk.transfer.s3.progress.LoggingTransferListener"
level="INFO" additivity="false">
            <AppenderRef ref="AlignedConsoleAppender"/>
        </logger>
    </Loggers>
</Configuration>
```

`Log4j2` logs the progress. The following is example output for a 21.3 MB file upload.

```
Transfer initiated...
|                               | 0.0%
|====                          | 21.1%
|=====                        | 60.5%
```

```

|=====| 100.0%
Transfer complete!
*/
}

```

Acompanhe o andamento do download de um arquivo.

```

public void trackDownloadFile(S3TransferManager transferManager, String
bucketName,
                                String key, String downloadedFilePath) {
    DownloadFileRequest downloadFileRequest = DownloadFileRequest.builder()
        .getObjectRequest(b -> b.bucket(bucketName).key(key))
        .addTransferListener(LoggingTransferListener.create()) // Add
listener.
        .destination(Paths.get(downloadedFilePath))
        .build();

    FileDownload downloadFile =
transferManager.downloadFile(downloadFileRequest);

    CompletedFileDownload downloadResult =
downloadFile.completionFuture().join();
    /*
        The SDK provides a LoggingTransferListener implementation of the
TransferListener interface.
        You can also implement the interface to provide your own logic.

        Configure log4J2 with settings such as the following.
        <Configuration status="WARN">
            <Appenders>
                <Console name="AlignedConsoleAppender"
target="SYSTEM_OUT">
                    <PatternLayout pattern="%m%n"/>
                </Console>
            </Appenders>

            <Loggers>
                <logger
name="software.amazon.awssdk.transfer.s3.progress.LoggingTransferListener"
level="INFO" additivity="false">
                    <AppenderRef ref="AlignedConsoleAppender"/>
                </logger>

```

```
        </Loggers>
    </Configuration>

    Log4J2 logs the progress. The following is example output for a 21.3
    MB file download.
        Transfer initiated...
        |=====| 39.4%
        |=====| 78.8%
        |=====| 100.0%
        Transfer complete!

    */
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for Java 2.x.
 - [GetObject](#)
 - [PutObject](#)

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de abordagens para testes de unidade e integração com SDK da AWS

O exemplo de código a seguir mostra técnicas de práticas recomendadas ao escrever testes de unidade e de integração usando SDK da AWS.

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Cargo.toml para exemplos de testes.

```
[package]
name = "testing-examples"
version = "0.1.0"
authors = [
  "John Disanti <jdisanti@amazon.com>",
  "Doug Schwartz <dougsch@amazon.com>",
]
edition = "2021"

# snippet-start:[testing.rust.Cargo.toml]
[dependencies]
async-trait = "0.1.51"
aws-config = { version = "1.0.1", features = ["behavior-version-latest"] }
aws-credential-types = { version = "1.0.1", features = [ "hardcoded-credentials", ] }
aws-sdk-s3 = { version = "1.4.0" }
aws-smithy-types = { version = "1.0.1" }
aws-smithy-runtime = { version = "1.0.1", features = ["test-util"] }
aws-smithy-runtime-api = { version = "1.0.1", features = ["test-util"] }
aws-types = { version = "1.0.1" }
clap = { version = "~4.4", features = ["derive"] }
http = "0.2.9"
mockall = "0.11.4"
serde_json = "1"
tokio = { version = "1.20.1", features = ["full"] }
tracing-subscriber = { version = "0.3.15", features = ["env-filter"] }
# snippet-end:[testing.rust.Cargo.toml]

[[bin]]
name = "main"
path = "src/main.rs"
```

Exemplo de teste de unidade usando automock e um wrapper de serviço.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

// snippet-start:[testing.rust.wrapper]
// snippet-start:[testing.rust.wrapper-uses]
use aws_sdk_s3 as s3;
#[allow(unused_imports)]
use mockall::automock;
```

```
use s3::operation::list_objects_v2::{ListObjectsV2Error, ListObjectsV2Output};
// snippet-end:[testing.rust.wrapper-uses]

// snippet-start:[testing.rust.wrapper-which-impl]
#[cfg(test)]
pub use MockS3Impl as S3;
#[cfg(not(test))]
pub use S3Impl as S3;
// snippet-end:[testing.rust.wrapper-which-impl]

// snippet-start:[testing.rust.wrapper-impl]
#[allow(dead_code)]
pub struct S3Impl {
    inner: s3::Client,
}

#[cfg_attr(test, automock)]
impl S3Impl {
    #[allow(dead_code)]
    pub fn new(inner: s3::Client) -> Self {
        Self { inner }
    }

    #[allow(dead_code)]
    pub async fn list_objects(
        &self,
        bucket: &str,
        prefix: &str,
        continuation_token: Option<String>,
    ) -> Result<ListObjectsV2Output, s3::error::SdkError<ListObjectsV2Error>> {
        self.inner
            .list_objects_v2()
            .bucket(bucket)
            .prefix(prefix)
            .set_continuation_token(continuation_token)
            .send()
            .await
    }
}
// snippet-end:[testing.rust.wrapper-impl]

// snippet-start:[testing.rust.wrapper-func]
#[allow(dead_code)]
```

```

pub async fn determine_prefix_file_size(
    // Now we take a reference to our trait object instead of the S3 client
    // s3_list: ListObjectsService,
    s3_list: S3,
    bucket: &str,
    prefix: &str,
) -> Result<usize, s3::Error> {
    let mut next_token: Option<String> = None;
    let mut total_size_bytes = 0;
    loop {
        let result = s3_list
            .list_objects(bucket, prefix, next_token.take())
            .await?;

        // Add up the file sizes we got back
        for object in result.contents() {
            total_size_bytes += object.size().unwrap_or(0) as usize;
        }

        // Handle pagination, and break the loop if there are no more pages
        next_token = result.next_continuation_token.clone();
        if next_token.is_none() {
            break;
        }
    }
    Ok(total_size_bytes)
}
// snippet-end:[testing.rust.wrapper-func]
// snippet-end:[testing.rust.wrapper]

// snippet-start:[testing.rust.wrapper-test-mod]
#[cfg(test)]
mod test {
    // snippet-start:[testing.rust.wrapper-tests]
    use super::*;
    use mockall::predicate::eq;

    // snippet-start:[testing.rust.wrapper-test-single]
    #[tokio::test]
    async fn test_single_page() {
        let mut mock = MockS3Impl::default();
        mock.expect_list_objects()
            .with(eq("test-bucket"), eq("test-prefix"), eq(None))
            .return_once(|_, _, _| {

```

```
        Ok(ListObjectsV2Output::builder()
            .set_contents(Some(vec![
                // Mock content for ListObjectsV2 response
                s3::types::Object::builder().size(5).build(),
                s3::types::Object::builder().size(2).build(),
            ]))
            .build())
    });

// Run the code we want to test with it
let size = determine_prefix_file_size(mock, "test-bucket", "test-prefix")
    .await
    .unwrap();

// Verify we got the correct total size back
assert_eq!(7, size);
}
// snippet-end:[testing.rust.wrapper-test-single]

// snippet-start:[testing.rust.wrapper-test-multiple]
#[tokio::test]
async fn test_multiple_pages() {
    // Create the Mock instance with two pages of objects now
    let mut mock = MockS3Impl::default();
    mock.expect_list_objects()
        .with(eq("test-bucket"), eq("test-prefix"), eq(None))
        .return_once(|_, _, _| {
            Ok(ListObjectsV2Output::builder()
                .set_contents(Some(vec![
                    // Mock content for ListObjectsV2 response
                    s3::types::Object::builder().size(5).build(),
                    s3::types::Object::builder().size(2).build(),
                ]))
                .set_next_continuation_token(Some("next".to_string()))
                .build())
        });
    mock.expect_list_objects()
        .with(
            eq("test-bucket"),
            eq("test-prefix"),
            eq(Some("next".to_string()))
        )
        .return_once(|_, _, _| {
            Ok(ListObjectsV2Output::builder()
                .set_contents(Some(vec![
                    // Mock content for ListObjectsV2 response
                    s3::types::Object::builder().size(5).build(),
                    s3::types::Object::builder().size(2).build(),
                ]))
                .set_next_continuation_token(Some("next".to_string()))
                .build())
        });
}
```

```

        .set_contents(Some(vec![
            // Mock content for ListObjectsV2 response
            s3::types::Object::builder().size(3).build(),
            s3::types::Object::builder().size(9).build(),
        ]))
        .build()
    });

    // Run the code we want to test with it
    let size = determine_prefix_file_size(mock, "test-bucket", "test-prefix")
        .await
        .unwrap();

    assert_eq!(19, size);
}
// snippet-end:[testing.rust.wrapper-test-multiple]
// snippet-end:[testing.rust.wrapper-tests]
}
// snippet-end:[testing.rust.wrapper-test-mod]

```

Exemplo de teste de integração usando StaticReplayClient.

```

// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

// snippet-start:[testing.rust.replay-uses]
use aws_sdk_s3 as s3;
// snippet-end:[testing.rust.replay-uses]

#[allow(dead_code)]
// snippet-start:[testing.rust.replay]
pub async fn determine_prefix_file_size(
    // Now we take a reference to our trait object instead of the S3 client
    // s3_list: ListObjectsService,
    s3: s3::Client,
    bucket: &str,
    prefix: &str,
) -> Result<usize, s3::Error> {
    let mut next_token: Option<String> = None;
    let mut total_size_bytes = 0;
    loop {
        let result = s3

```



```
        .list_objects_v2()
        .prefix(prefix)
        .bucket(bucket)
        .set_continuation_token(next_token.take())
        .send()
        .await?;

    // Add up the file sizes we got back
    for object in result.contents() {
        total_size_bytes += object.size().unwrap_or(0) as usize;
    }

    // Handle pagination, and break the loop if there are no more pages
    next_token = result.next_continuation_token.clone();
    if next_token.is_none() {
        break;
    }
}
Ok(total_size_bytes)
}
// snippet-end:[testing.rust.replay]

#[allow(dead_code)]
// snippet-start:[testing.rust.replay-tests]
// snippet-start:[testing.rust.replay-make-credentials]
fn make_s3_test_credentials() -> s3::config::Credentials {
    s3::config::Credentials::new(
        "ATESTCLIENT",
        "astestsecretkey",
        Some("atestsessiontoken".to_string()),
        None,
        "",
    )
}
// snippet-end:[testing.rust.replay-make-credentials]

// snippet-start:[testing.rust.replay-test-module]
#[cfg(test)]
mod test {
    // snippet-start:[testing.rust.replay-test-single]
    use super::*;
    use aws_config::BehaviorVersion;
    use aws_sdk_s3 as s3;
```

```

    use aws_smithy_runtime::client::http::test_util::{ReplayEvent,
StaticReplayClient};
    use aws_smithy_types::body::SdkBody;

    #[tokio::test]
    async fn test_single_page() {
        let page_1 = ReplayEvent::new(
            http::Request::builder()
                .method("GET")
                .uri("https://test-bucket.s3.us-east-1.amazonaws.com/?list-
type=2&prefix=test-prefix")
                .body(SdkBody::empty())
                .unwrap(),
            http::Response::builder()
                .status(200)
                .body(SdkBody::from(include_str!("./testing/
response_1.xml")))
                .unwrap(),
        );
        let replay_client = StaticReplayClient::new(vec![page_1]);
        let client: s3::Client = s3::Client::from_conf(
            s3::Config::builder()
                .behavior_version(BehaviorVersion::latest())
                .credentials_provider(make_s3_test_credentials())
                .region(s3::config::Region::new("us-east-1"))
                .http_client(replay_client.clone())
                .build(),
        );

        // Run the code we want to test with it
        let size = determine_prefix_file_size(client, "test-bucket", "test-
prefix")
            .await
            .unwrap();

        // Verify we got the correct total size back
        assert_eq!(7, size);
        replay_client.assert_requests_match(&[]);
    }
    // snippet-end:[testing.rust.replay-test-single]

    // snippet-start:[testing.rust.replay-test-multiple]
    #[tokio::test]
    async fn test_multiple_pages() {

```

```

// snippet-start:[testing.rust.replay-create-replay]
let page_1 = ReplayEvent::new(
    http::Request::builder()
        .method("GET")
        .uri("https://test-bucket.s3.us-east-1.amazonaws.com/?list-
type=2&prefix=test-prefix")
        .body(SdkBody::empty())
        .unwrap(),
    http::Response::builder()
        .status(200)
        .body(SdkBody::from(include_str!("./testing/
response_multi_1.xml")))
        .unwrap(),
);
let page_2 = ReplayEvent::new(
    http::Request::builder()
        .method("GET")
        .uri("https://test-bucket.s3.us-east-1.amazonaws.com/?list-
type=2&prefix=test-prefix&continuation-token=next")
        .body(SdkBody::empty())
        .unwrap(),
    http::Response::builder()
        .status(200)
        .body(SdkBody::from(include_str!("./testing/
response_multi_2.xml")))
        .unwrap(),
);
let replay_client = StaticReplayClient::new(vec![page_1, page_2]);
// snippet-end:[testing.rust.replay-create-replay]
// snippet-start:[testing.rust.replay-create-client]
let client: s3::Client = s3::Client::from_conf(
    s3::Config::builder()
        .behavior_version(BehaviorVersion::latest())
        .credentials_provider(make_s3_test_credentials())
        .region(s3::config::Region::new("us-east-1"))
        .http_client(replay_client.clone())
        .build(),
);
// snippet-end:[testing.rust.replay-create-client]

// Run the code we want to test with it
// snippet-start:[testing.rust.replay-test-and-verify]
let size = determine_prefix_file_size(client, "test-bucket", "test-
prefix")

```

```
        .await
        .unwrap();

    assert_eq!(19, size);

    replay_client.assert_requests_match(&[]);
    // snippet-end:[testing.rust.replay-test-and-verify]
}
// snippet-end:[testing.rust.replay-test-multiple]
}
// snippet-end:[testing.rust.replay-tests]
// snippet-end:[testing.rust.replay-test-module]
```

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Fazer upload recursivo de um diretório local em um bucket do Amazon Simple Storage Service (Amazon S3)

O exemplo de código a seguir mostra como fazer upload recursivo de um diretório local em um bucket do Amazon Simple Storage Service (Amazon S3).

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Use [S3TransferManager](#) para [fazer upload de um diretório local](#). Veja o [arquivo completo e teste](#).

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.s3.model.ObjectIdentifier;
```

```
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedDirectoryUpload;
import software.amazon.awssdk.transfer.s3.model.DirectoryUpload;
import software.amazon.awssdk.transfer.s3.model.UploadDirectoryRequest;

import java.net.URI;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.file.Paths;
import java.util.UUID;

    public Integer uploadDirectory(S3TransferManager transferManager,
        URI sourceDirectory, String bucketName) {
        DirectoryUpload directoryUpload =
transferManager.uploadDirectory(UploadDirectoryRequest.builder()
        .source(Paths.get(sourceDirectory))
        .bucket(bucketName)
        .build());

        CompletedDirectoryUpload completedDirectoryUpload =
directoryUpload.completionFuture().join();
        completedDirectoryUpload.failedTransfers()
            .forEach(fail -> logger.warn("Object [{}] failed to transfer",
fail.toString()));
        return completedDirectoryUpload.failedTransfers().size();
    }
```

- Para obter detalhes da API, consulte [UploadDirectory](#) na Referência da API do AWS SDK for Java 2.x.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.


Fazer upload ou download de arquivos grandes de e para o Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como fazer upload e download de arquivos grandes de e para o Amazon S3.

Para obter mais informações, consulte [Carregar um objeto usando carregamento fracionado](#).

.NET

AWS SDK for .NET

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Chame funções que transferem arquivos de e para um bucket do S3 usando o Amazon S3 TransferUtility.

```
global using System.Text;
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.
using Microsoft.Extensions.Configuration;

IAmazonS3 client = new AmazonS3Client();
var transferUtil = new TransferUtility(client);
IConfiguration _configuration;

_configuration = new ConfigurationBuilder()
    .SetBasePath(Directory.GetCurrentDirectory())
    .AddJsonFile("settings.json") // Load test settings from JSON file.
    .AddJsonFile("settings.local.json",
        true) // Optionally load local settings.
    .Build();

// Edit the values in settings.json to use an S3 bucket and files that
// exist on your AWS account and on the local computer where you
// run this scenario.
var bucketName = _configuration["BucketName"];
```

```
var localPath =
    $"{Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData)}\
    \TransferFolder";

DisplayInstructions();

PressEnter();

Console.WriteLine();

// Upload a single file to an S3 bucket.
DisplayTitle("Upload a single file");

var fileToUpload = _configuration["FileToUpload"];
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.
DisplayTitle("Upload all files from a local directory");
Console.WriteLine("Upload all the files in a local folder to an S3 bucket.");
const string keyPrefix = "UploadFolder";
var uploadPath = $"{localPath}\\UploadFolder";

Console.WriteLine($"Uploading the files in {uploadPath} to {bucketName}");
DisplayTitle($"{uploadPath} files");
DisplayLocalFiles(uploadPath);
Console.WriteLine();

PressEnter();

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.
DisplayTitle("Download a single file");
Console.WriteLine("Now we will download a single file from an S3 bucket.");

var keyName = _configuration["FileToDownload"];

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.
DisplayTitle("Download the contents of an S3 bucket");
var s3Path = _configuration["S3Path"];
var downloadPath = $"{localPath}\\{s3Path}";

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();

    // Displays the title for a section of the scenario.
    static void DisplayTitle(string titleText)
    {
        var sepBar = new string('-', Console.WindowWidth);

        Console.WriteLine(sepBar);
        Console.WriteLine(CenterText(titleText));
        Console.WriteLine(sepBar);
    }

    // Displays a description of the actions to be performed by the scenario.
    static void DisplayInstructions()
    {
        var sepBar = new string('-', Console.WindowWidth);

        DisplayTitle("Amazon S3 Transfer Utility Basics");
        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 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}");
    }

    // Pauses the scenario.
    static void PressEnter()
    {
        Console.WriteLine("Press <Enter> to continue.");
        _ = Console.ReadLine();
        Console.WriteLine("\n");
    }

    // Returns the string textToCenter, padded on the left with spaces
    // that center the text on the console display.
```

```
static string CenterText(string textToCenter)
{
    var centeredText = new StringBuilder();
    var screenWidth = Console.WindowWidth;
    centeredText.Append(new string(' ', (int)(screenWidth -
textToCenter.Length) / 2));
    centeredText.Append(textToCenter);
    return centeredText.ToString();
}

// Displays a list of file names included in the specified path.
static void DisplayLocalFiles(string localPath)
{
    var fileList = Directory.GetFiles(localPath);
    if (fileList.Length > 0)
    {
        foreach (var fileName in fileList)
        {
            Console.WriteLine(fileName);
        }
    }
}

// Displays a list of the files in the specified S3 bucket and prefix.
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.

```
/// <summary>
/// Uploads a single file from the local computer to an S3 bucket.
/// </summary>
/// <param name="transferUtil">The transfer initialized TransferUtility
/// object.</param>
/// <param name="bucketName">The name of the S3 bucket where the file
/// will be stored.</param>
/// <param name="fileName">The name of the file to upload.</param>
/// <param name="localPath">The local path where the file is stored.</
param>
/// <returns>A boolean value indicating the success of the action.</
returns>
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.

```
/// <summary>
/// Uploads all the files in a local directory to a directory in an S3
/// bucket.
/// </summary>
/// <param name="transferUtil">The transfer initialized TransferUtility
/// object.</param>
/// <param name="bucketName">The name of the S3 bucket where the files
/// will be stored.</param>
/// <param name="keyPrefix">The key prefix is the S3 directory where
/// the files will be stored.</param>
/// <param name="localPath">The local directory that contains the files
/// to be uploaded.</param>
/// <returns>A Boolean value representing the success of the action.</
returns>
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.

```

/// <summary>
/// Download a single file from an S3 bucket to the local computer.
/// </summary>
/// <param name="transferUtil">The transfer initialized TransferUtility
/// object.</param>
/// <param name="bucketName">The name of the S3 bucket containing the
/// file to download.</param>
/// <param name="keyName">The name of the file to download.</param>
/// <param name="localPath">The path on the local computer where the
/// downloaded file will be saved.</param>
/// <returns>A Boolean value indicating the results of the action.</
returns>
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}",
    });

    return (File.Exists($"{localPath}\\{keyName}"));
}

```

Baixe o conteúdo de um bucket do S3.

```

/// <summary>
/// Downloads the contents of a directory in an S3 bucket to a
/// directory on the local computer.
/// </summary>
/// <param name="transferUtil">The transfer initialized TransferUtility
/// object.</param>
/// <param name="bucketName">The bucket containing the files to
download.</param>
/// <param name="s3Path">The S3 directory where the files are located.</
param>
/// <param name="localPath">The local path to which the files will be
/// saved.</param>
/// <returns>A Boolean value representing the success of the action.</
returns>
public static async Task<bool> DownloadS3DirectoryAsync(
    TransferUtility transferUtil,
    string bucketName,
    string s3Path,
    string localPath)
{
    int fileCount = 0;

    // If the directory doesn't exist, it will be created.

```

```
        if (Directory.Exists(s3Path))
        {
            var files = Directory.GetFiles(localPath);
            fileCount = files.Length;
        }

        await transferUtil.DownloadDirectoryAsync(new
TransferUtilityDownloadDirectoryRequest
        {
            BucketName = bucketName,
            LocalDirectory = localPath,
            S3Directory = s3Path,
        });

        if (Directory.Exists(localPath))
        {
            var files = Directory.GetFiles(localPath);
            if (files.Length > fileCount)
            {
                return true;
            }

            // No change in the number of files. Assume
            // the download failed.
            return false;
        }

        // The local directory doesn't exist. No files
        // were downloaded.
        return false;
    }
}
```

Acompanhe o progresso de um upload usando TransferUtility.

```
using System;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Transfer;

/// <summary>
/// This example shows how to track the progress of a multipart upload
```

```
/// using the Amazon Simple Storage Service (Amazon S3) TransferUtility to
/// upload to an Amazon S3 bucket.
/// </summary>
public class TrackMPUUsingHighLevelAPI
{
    public static async Task Main()
    {
        string bucketName = "doc-example-bucket";
        string keyName = "sample_pic.png";
        string path = "filepath/directory/";
        string filePath = $"{path}{keyName}";

        // If the AWS Region defined for your default user is different
        // from the Region where your Amazon S3 bucket is located,
        // pass the Region name to the Amazon S3 client object's constructor.
        // For example: RegionEndpoint.USWest2 or RegionEndpoint.USEast2.
        IAmazonS3 client = new AmazonS3Client();

        await TrackMPUAsync(client, bucketName, filePath, keyName);
    }

    /// <summary>
    /// Starts an Amazon S3 multipart upload and assigns an event handler to
    /// track the progress of the upload.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
    /// perform the multipart upload.</param>
    /// <param name="bucketName">The name of the bucket to which to upload
    /// the file.</param>
    /// <param name="filePath">The path, including the file name of the
    /// file to be uploaded to the Amazon S3 bucket.</param>
    /// <param name="keyName">The file name to be used in the
    /// destination Amazon S3 bucket.</param>
    public static async Task TrackMPUAsync(
        IAmazonS3 client,
        string bucketName,
        string filePath,
        string keyName)
    {
        try
        {
            var fileTransferUtility = new TransferUtility(client);

            // 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 ex)
{
    Console.WriteLine($"Error:: {ex.Message}");
}
}

/// <summary>
/// Event handler to check the progress of the multipart upload.
/// </summary>
/// <param name="sender">The object that raised the event.</param>
/// <param name="e">The object that contains multipart upload
/// information.</param>
public static void UploadRequest_UploadPartProgressEvent(object sender,
UploadProgressArgs e)
{
    // Process event.
    Console.WriteLine($"{e.TransferredBytes}/{e.TotalBytes}");
}
}
```

Faça upload de um objeto com criptografia.

```
using System;
using System.Collections.Generic;
using System.IO;
```

```
using System.Security.Cryptography;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// Uses the Amazon Simple Storage Service (Amazon S3) low level API to
/// perform a multipart upload to an Amazon S3 bucket.
/// </summary>
public class SSECLowLevelMPUcopyObject
{
    public static async Task Main()
    {
        string existingBucketName = "doc-example-bucket";
        string sourceKeyName = "sample_file.txt";
        string targetKeyName = "sample_file_copy.txt";
        string filePath = $"sample\\{targetKeyName}";

        // If the AWS Region defined for your default user is different
        // from the Region where your Amazon S3 bucket is located,
        // pass the Region name to the Amazon S3 client object's constructor.
        // For example: RegionEndpoint.USEast1.
        IAmazonS3 client = new AmazonS3Client();

        // Create the encryption key.
        var base64Key = CreateEncryptionKey();

        await CreateSampleObjUsingClientEncryptionKeyAsync(
            client,
            existingBucketName,
            sourceKeyName,
            filePath,
            base64Key);
    }

    /// <summary>
    /// Creates the encryption key to use with the multipart upload.
    /// </summary>
    /// <returns>A string containing the base64-encoded key for encrypting
    /// the multipart upload.</returns>
    public static string CreateEncryptionKey()
    {
        Aes aesEncryption = Aes.Create();
        aesEncryption.KeySize = 256;
    }
}
```

```
        aesEncryption.GenerateKey();
        string base64Key = Convert.ToBase64String(aesEncryption.Key);
        return base64Key;
    }

    /// <summary>
    /// Creates and uploads an object using a multipart upload.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 object used to
    /// initialize and perform the multipart upload.</param>
    /// <param name="existingBucketName">The name of the bucket to which
    /// the object will be uploaded.</param>
    /// <param name="sourceKeyName">The source object name.</param>
    /// <param name="filePath">The location of the source object.</param>
    /// <param name="base64Key">The encryption key to use with the upload.</
param>
    public static async Task CreateSampleObjUsingClientEncryptionKeyAsync(
        IAmazonS3 client,
        string existingBucketName,
        string sourceKeyName,
        string filePath,
        string base64Key)
    {
        List<UploadPartResponse> uploadResponses = new
List<UploadPartResponse>();

        InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
        {
            BucketName = existingBucketName,
            Key = sourceKeyName,
            ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
            ServerSideEncryptionCustomerProvidedKey = base64Key,
        };

        InitiateMultipartUploadResponse initResponse =
            await client.InitiateMultipartUploadAsync(initiateRequest);

        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
client.UploadPartAsync(uploadRequest));

    filePosition += partSize;
}

CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = sourceKeyName,
    UploadId = initResponse.UploadId,
};
completeRequest.AddPartETags(uploadResponses);


CompleteMultipartUploadResponse completeUploadResponse =
    await client.CompleteMultipartUploadAsync(completeRequest);
}
catch (Exception exception)
{
    Console.WriteLine($"Exception occurred: {exception.Message}");

    // If there was an error, abort the multipart upload.
    AbortMultipartUploadRequest abortMPURquest = new
AbortMultipartUploadRequest
{
```

```
        BucketName = existingBucketName,  
        Key = sourceKeyName,  
        UploadId = initResponse.UploadId,  
    };  
  
    await client.AbortMultipartUploadAsync(abortMPURequest);  
    }  
}
```

Go

SDK para Go V2

 Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Faça upload de um objeto grande usando um gerenciador de uploads para dividir os dados em partes e fazer upload delas simultaneamente.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)  
// actions  
// used in the examples.  
// It contains S3Client, an Amazon S3 service client that is used to perform  
// bucket  
// and object actions.  
type BucketBasics struct {  
    S3Client *s3.Client  
}  
  
// UploadLargeObject uses an upload manager to upload data to an object in a  
// bucket.
```

```
// The upload manager breaks large data into parts and uploads the parts
concurrently.
func (basics BucketBasics) UploadLargeObject(bucketName string, objectKey string,
largeObject []byte) error {
    largeBuffer := bytes.NewReader(largeObject)
    var partMiBs int64 = 10
    uploader := manager.NewUploader(basics.S3Client, func(u *manager.Uploader) {
        u.PartSize = partMiBs * 1024 * 1024
    })
    _, err := uploader.Upload(context.TODO(), &s3.PutObjectInput{
        Bucket: aws.String(bucketName),
        Key:     aws.String(objectKey),
        Body:   largeBuffer,
    })
    if err != nil {
        log.Printf("Couldn't upload large object to %v:%v. Here's why: %v\n",
            bucketName, objectKey, err)
    }

    return err
}
```

Faça download de um objeto grande usando um gerenciador de downloads para obter os dados em partes e baixá-las simultaneamente.

```
// DownloadLargeObject uses a download manager to download an object from a
bucket.
// The download manager gets the data in parts and writes them to a buffer until
all of
// the data has been downloaded.
func (basics BucketBasics) DownloadLargeObject(bucketName string, objectKey
string) ([]byte, error) {
    var partMiBs int64 = 10
    downloader := manager.NewDownloader(basics.S3Client, func(d *manager.Downloader)
{
        d.PartSize = partMiBs * 1024 * 1024
    })
    buffer := manager.NewWriteAtBuffer([]byte{})
    _, err := downloader.Download(context.TODO(), buffer, &s3.GetObjectInput{
        Bucket: aws.String(bucketName),
```

```
    Key:    aws.String(objectKey),
  })
  if err != nil {
    log.Printf("Couldn't download large object from %v:%v. Here's why: %v\n",
      bucketName, objectKey, err)
  }
  return buffer.Bytes(), err
}
```

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Chame funções que transferem arquivos de e para um bucket do S3 usando o `S3TransferManager`.

```
public Integer downloadObjectsToDirectory(S3TransferManager transferManager,
    URI destinationPathURI, String bucketName) {
    DirectoryDownload directoryDownload =
transferManager.downloadDirectory(DownloadDirectoryRequest.builder()
        .destination(Paths.get(destinationPathURI))
        .bucket(bucketName)
        .build());
    CompletedDirectoryDownload completedDirectoryDownload =
directoryDownload.completionFuture().join();

    completedDirectoryDownload.failedTransfers()
        .forEach(fail -> logger.warn("Object [{}] failed to transfer",
fail.toString()));
    return completedDirectoryDownload.failedTransfers().size();
}
```

Carregue um diretório local inteiro.

```
public Integer uploadDirectory(S3TransferManager transferManager,
    URI sourceDirectory, String bucketName) {
    DirectoryUpload directoryUpload =
transferManager.uploadDirectory(UploadDirectoryRequest.builder()
        .source(Paths.get(sourceDirectory))
        .bucket(bucketName)
        .build());

    CompletedDirectoryUpload completedDirectoryUpload =
directoryUpload.completionFuture().join();
    completedDirectoryUpload.failedTransfers()
        .forEach(fail -> logger.warn("Object [{}] failed to transfer",
fail.toString()));
    return completedDirectoryUpload.failedTransfers().size();
}
```

Carregue um único arquivo.

```
public String uploadFile(S3TransferManager transferManager, String
bucketName,
    String key, URI filePathURI) {
    UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
        .putObjectRequest(b -> b.bucket(bucketName).key(key))
        .source(Paths.get(filePathURI))
        .build();

    FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);

    CompletedFileUpload uploadResult = fileUpload.completionFuture().join();
    return uploadResult.response().eTag();
}
```


JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Faça upload de um arquivo grande.

```
import {
  CreateMultipartUploadCommand,
  UploadPartCommand,
  CompleteMultipartUploadCommand,
  AbortMultipartUploadCommand,
  S3Client,
} from "@aws-sdk/client-s3";

const twentyFiveMB = 25 * 1024 * 1024;

export const createString = (size = twentyFiveMB) => {
  return "x".repeat(size);
};

export const main = async () => {
  const s3Client = new S3Client({});
  const bucketName = "test-bucket";
  const key = "multipart.txt";
  const str = createString();
  const buffer = Buffer.from(str, "utf8");

  let uploadId;

  try {
    const multipartUpload = await s3Client.send(
      new CreateMultipartUploadCommand({
        Bucket: bucketName,
        Key: key,
      }),
    );
  }
};
```

```
uploadId = multipartUpload.UploadId;

const uploadPromises = [];
// Multipart uploads require a minimum size of 5 MB per part.
const partSize = Math.ceil(buffer.length / 5);

// Upload each part.
for (let i = 0; i < 5; i++) {
  const start = i * partSize;
  const end = start + partSize;
  uploadPromises.push(
    s3Client
      .send(
        new UploadPartCommand({
          Bucket: bucketName,
          Key: key,
          UploadId: uploadId,
          Body: buffer.subarray(start, end),
          PartNumber: i + 1,
        })
      )
      .then((d) => {
        console.log("Part", i + 1, "uploaded");
        return d;
      })
  );
}

const uploadResults = await Promise.all(uploadPromises);

return await s3Client.send(
  new CompleteMultipartUploadCommand({
    Bucket: bucketName,
    Key: key,
    UploadId: uploadId,
    MultipartUpload: {
      Parts: uploadResults.map(({ ETag }, i) => ({
        ETag,
        PartNumber: i + 1,
      })),
    },
  })
);
```

```
// Verify the output by downloading the file from the Amazon Simple Storage
Service (Amazon S3) console.
// Because the output is a 25 MB string, text editors might struggle to open
the file.
} catch (err) {
  console.error(err);

  if (uploadId) {
    const abortCommand = new AbortMultipartUploadCommand({
      Bucket: bucketName,
      Key: key,
      UploadId: uploadId,
    });

    await s3Client.send(abortCommand);
  }
}
};
```

Baixe um arquivo grande.

```
import { GetObjectCommand, S3Client } from "@aws-sdk/client-s3";
import { createWriteStream } from "fs";

const s3Client = new S3Client({});
const oneMB = 1024 * 1024;

export const getObjectRange = ({ bucket, key, start, end }) => {
  const command = new GetObjectCommand({
    Bucket: bucket,
    Key: key,
    Range: `bytes=${start}-${end}`,
  });

  return s3Client.send(command);
};

/**
 * @param {string | undefined} contentRange
 */
export const getRangeAndLength = (contentRange) => {
  const [range, length] = contentRange.split("/");
```

```
const [start, end] = range.split("-");
return {
  start: parseInt(start),
  end: parseInt(end),
  length: parseInt(length),
};
};

export const isComplete = ({ end, length }) => end === length - 1;

// When downloading a large file, you might want to break it down into
// smaller pieces. Amazon S3 accepts a Range header to specify the start
// and end of the byte range to be downloaded.
const downloadInChunks = async ({ bucket, key }) => {
  const writeStream = createWriteStream(
    fileURLToPath(new URL(`./${key}`, import.meta.url)),
  ).on("error", (err) => console.error(err));

  let rangeAndLength = { start: -1, end: -1, length: -1 };

  while (!isComplete(rangeAndLength)) {
    const { end } = rangeAndLength;
    const nextRange = { start: end + 1, end: end + oneMB };

    console.log(`Downloading bytes ${nextRange.start} to ${nextRange.end}`);

    const { ContentRange, Body } = await getObjectRange({
      bucket,
      key,
      ...nextRange,
    });

    writeStream.write(await Body.transformToByteArray());
    rangeAndLength = getRangeAndLength(ContentRange);
  }
};

export const main = async () => {
  await downloadInChunks({
    bucket: "my-cool-bucket",
    key: "my-cool-object.txt",
  });
};
```

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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_chunksize_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"{'':4}{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."
        )
```

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

```
use std::fs::File;
use std::io::prelude::*;
use std::path::Path;

use aws_config::meta::region::RegionProviderChain;
use aws_sdk_s3::error::DisplayErrorContext;
use aws_sdk_s3::operation::{
    create_multipart_upload::CreateMultipartUploadOutput,
    get_object::GetObjectOutput,
};
use aws_sdk_s3::types::{CompletedMultipartUpload, CompletedPart};
use aws_sdk_s3::{config::Region, Client as S3Client};
use aws_smithy_types::byte_stream::{ByteStream, Length};
```

```
use rand::distributions::Alphanumeric;
use rand::{thread_rng, Rng};
use s3_service::error::Error;
use std::process;
use uuid::Uuid;

//In bytes, minimum chunk size of 5MB. Increase CHUNK_SIZE to send larger chunks.
const CHUNK_SIZE: u64 = 1024 * 1024 * 5;
const MAX_CHUNKS: u64 = 10000;

#[tokio::main]
pub async fn main() {
    if let Err(err) = run_example().await {
        eprintln!("Error: {}", DisplayErrorContext(err));
        process::exit(1);
    }
}

async fn run_example() -> Result<(), Error> {
    let shared_config = aws_config::load_from_env().await;
    let client = S3Client::new(&shared_config);

    let bucket_name = format!("doc-example-bucket-{}", Uuid::new_v4());
    let region_provider = RegionProviderChain::first_try(Region::new("us-
west-2"));
    let region = region_provider.region().await.unwrap();
    s3_service::create_bucket(&client, &bucket_name, region.as_ref()).await?;

    let key = "sample.txt".to_string();
    let multipart_upload_res: CreateMultipartUploadOutput = client
        .create_multipart_upload()
        .bucket(&bucket_name)
        .key(&key)
        .send()
        .await
        .unwrap();
    let upload_id = multipart_upload_res.upload_id().unwrap();

    //Create a file of random characters for the upload.
    let mut file = File::create(&key).expect("Could not create sample file.");
    // Loop until the file is 5 chunks.
    while file.metadata().unwrap().len() <= CHUNK_SIZE * 4 {
        let rand_string: String = thread_rng()
            .sample_iter(&Alphanumeric)
```

```
        .take(256)
        .map(char::from)
        .collect();
    let return_string: String = "\n".to_string();
    file.write_all(rand_string.as_ref())
        .expect("Error writing to file.");
    file.write_all(return_string.as_ref())
        .expect("Error writing to file.");
}

let path = Path::new(&key);
let file_size = tokio::fs::metadata(path)
    .await
    .expect("it exists I swear")
    .len();

let mut chunk_count = (file_size / CHUNK_SIZE) + 1;
let mut size_of_last_chunk = file_size % CHUNK_SIZE;
if size_of_last_chunk == 0 {
    size_of_last_chunk = CHUNK_SIZE;
    chunk_count -= 1;
}

if file_size == 0 {
    panic!("Bad file size.");
}
if chunk_count > MAX_CHUNKS {
    panic!("Too many chunks! Try increasing your chunk size.")
}

let mut upload_parts: Vec<CompletedPart> = Vec::new();

for chunk_index in 0..chunk_count {
    let this_chunk = if chunk_count - 1 == chunk_index {
        size_of_last_chunk
    } else {
        CHUNK_SIZE
    };
    let stream = ByteStream::read_from()
        .path(path)
        .offset(chunk_index * CHUNK_SIZE)
        .length(Length::Exact(this_chunk))
        .build()
        .await
```

```
        .unwrap());
    //Chunk index needs to start at 0, but part numbers start at 1.
    let part_number = (chunk_index as i32) + 1;
    let upload_part_res = client
        .upload_part()
        .key(&key)
        .bucket(&bucket_name)
        .upload_id(upload_id)
        .body(stream)
        .part_number(part_number)
        .send()
        .await?;
    upload_parts.push(
        CompletedPart::builder()
            .e_tag(upload_part_res.e_tag.unwrap_or_default())
            .part_number(part_number)
            .build(),
    );
}
let completed_multipart_upload: CompletedMultipartUpload =
CompletedMultipartUpload::builder()
    .set_parts(Some(upload_parts))
    .build();

let _complete_multipart_upload_res = client
    .complete_multipart_upload()
    .bucket(&bucket_name)
    .key(&key)
    .multipart_upload(completed_multipart_upload)
    .upload_id(upload_id)
    .send()
    .await
    .unwrap();

let data: GetObjectOutput = s3_service::download_object(&client,
&bucket_name, &key).await?;
let data_length: u64 = data
    .content_length()
    .unwrap_or_default()
    .try_into()
    .unwrap();
if file.metadata().unwrap().len() == data_length {
    println!("Data lengths match.");
} else {
```

```
        println!("The data was not the same size!");
    }

    s3_service::delete_objects(&client, &bucket_name)
        .await
        .expect("Error emptying bucket.");
    s3_service::delete_bucket(&client, &bucket_name)
        .await
        .expect("Error deleting bucket.");

    Ok(())
}
```

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Faça upload de um fluxo de tamanho desconhecido em um objeto do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como fazer upload de um fluxo de tamanho desconhecido em um objeto do Amazon S3.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Use o [Cliente do S3 baseado em CRT da AWS](#).

```
import com.example.s3.util.AsyncExampleUtils;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.async.AsyncRequestBody;
```

```
import software.amazon.awssdk.core.async.BlockingInputStreamAsyncRequestBody;
import software.amazon.awssdk.core.exception.SdkException;
import software.amazon.awssdk.services.s3.S3AsyncClient;
import software.amazon.awssdk.services.s3.model.PutObjectResponse;

import java.io.ByteArrayInputStream;
import java.util.UUID;
import java.util.concurrent.CompletableFuture;

/**
 * @param s3CrtAsyncClient - To upload content from a stream of unknown
 size, use the AWS CRT-based S3 client. For more information, see
 * https://docs.aws.amazon.com/sdk-for-java/latest/
 developer-guide/crt-based-s3-client.html.
 * @param bucketName - The name of the bucket.
 * @param key - The name of the object.
 * @return software.amazon.awssdk.services.s3.model.PutObjectResponse -
 Returns metadata pertaining to the put object operation.
 */
public PutObjectResponse putObjectFromStream(S3AsyncClient s3CrtAsyncClient,
String bucketName, String key) {

    BlockingInputStreamAsyncRequestBody body =
        AsyncRequestBody.forBlockingInputStream(null); // 'null'
 indicates a stream will be provided later.

    CompletableFuture<PutObjectResponse> responseFuture =
        s3CrtAsyncClient.putObject(r -> r.bucket(bucketName).key(key),
body);

    // AsyncExampleUtils.randomString() returns a random string up to 100
 characters.
    String randomString = AsyncExampleUtils.randomString();
    logger.info("random string to upload: {}: length={}", randomString,
randomString.length());

    // Provide the stream of data to be uploaded.
    body.writeInputStream(new ByteArrayInputStream(randomString.getBytes()));

    PutObjectResponse response = responseFuture.join(); // Wait for the
 response.
    logger.info("Object {} uploaded to bucket {}.", key, bucketName);
    return response;
}
```

```
}
```

Use o [Gerenciador de transferências do Amazon S3](#).

```
import com.example.s3.util.AsyncExampleUtils;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.async.AsyncRequestBody;
import software.amazon.awssdk.core.async.BlockingInputStreamAsyncRequestBody;
import software.amazon.awssdk.core.exception.SdkException;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedUpload;
import software.amazon.awssdk.transfer.s3.model.Upload;

import java.io.ByteArrayInputStream;
import java.util.UUID;

/**
 * @param transferManager - To upload content from a stream of unknown size,
 * use the S3TransferManager based on the AWS CRT-based S3 client.
 *
 * For more information, see https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/transfer-manager.html.
 * @param bucketName - The name of the bucket.
 * @param key - The name of the object.
 * @return - software.amazon.awssdk.transfer.s3.model.CompletedUpload - The
 * result of the completed upload.
 */
public CompletedUpload uploadStream(S3TransferManager transferManager, String
bucketName, String key) {

    BlockingInputStreamAsyncRequestBody body =
        AsyncRequestBody.forBlockingInputStream(null); // 'null'
    indicates a stream will be provided later.

    Upload upload = transferManager.upload(builder -> builder
        .requestBody(body)
        .putObjectRequest(req -> req.bucket(bucketName).key(key))
        .build());

    // AsyncExampleUtils.randomString() returns a random string up to 100
    characters.
    String randomString = AsyncExampleUtils.randomString();
```



```
        logger.info("random string to upload: {}: length={}", randomString,
            randomString.length());

        // Provide the stream of data to be uploaded.
        body.writeInputStream(new ByteArrayInputStream(randomString.getBytes()));

        return upload.completionFuture().join();
    }
}
```

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar somas de verificação para trabalhar com um objeto do Amazon S3 usando um AWS SDK

O exemplo de código a seguir mostra como usar somas de verificação para trabalhar com um objeto do Amazon S3.

Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Os exemplos de código usam um subconjunto das importações a seguir.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.exception.SdkException;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.ChecksumAlgorithm;
```

```
import software.amazon.awssdk.services.s3.model.ChecksumMode;
import software.amazon.awssdk.services.s3.model.CompletedMultipartUpload;
import software.amazon.awssdk.services.s3.model.CompletedPart;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.UploadPartRequest;
import software.amazon.awssdk.services.s3.model.UploadPartResponse;
import software.amazon.awssdk.services.s3.waiters.S3Waiter;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.FileUpload;
import software.amazon.awssdk.transfer.s3.model.UploadFileRequest;

import java.io.FileInputStream;
import java.io.IOException;
import java.io.RandomAccessFile;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.ByteBuffer;
import java.nio.file.Paths;
import java.security.DigestInputStream;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.Base64;
import java.util.List;
import java.util.Objects;
import java.util.UUID;
```

Especifique um algoritmo de soma de verificação para o método `putObject` ao [criar PutObjectRequest](#).

```
public void putObjectWithChecksum() {
    s3Client.putObject(b -> b
        .bucket(bucketName)
        .key(key)
        .checksumAlgorithm(ChecksumAlgorithm.CRC32),
        RequestBody.fromString("This is a test"));
}
```

Verifique a soma de verificação do método `getObject` ao [criar o GetObjectRequest](#).

```
public GetObjectResponse getObjectWithChecksum() {
    return s3Client.getObject(b -> b
        .bucket(bucketName)
        .key(key)
        .checksumMode(ChecksumMode.ENABLED))
        .response();
}
```

Calcule previamente uma soma de verificação para o método `putObject` ao [criar o PutObjectRequest](#).

```
public void putObjectWithPrecalculatedChecksum(String filePath) {
    String checksum = calculateChecksum(filePath, "SHA-256");

    s3Client.putObject((b -> b
        .bucket(bucketName)
        .key(key)
        .checksumSHA256(checksum)),
        RequestBody.fromFile(Paths.get(filePath)));
}
```

Use o [Gerenciador de Transferências do S3](#) na parte superior do [cliente do S3 baseado no AWS CRT](#) para realizar de forma transparente um carregamento fracionado quando o tamanho do conteúdo exceder um limite. O limite padrão é 8 MB.

Você pode especificar um algoritmo de soma de verificação para o SDK usar. Por padrão, o SDK usa o algoritmo CRC32.

```
public void multipartUploadWithChecksumTm(String filePath) {
    S3TransferManager transferManager = S3TransferManager.create();
    UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
        .putObjectRequest(b -> b
            .bucket(bucketName)
            .key(key)
            .checksumAlgorithm(ChecksumAlgorithm.SHA1))
        .source(Paths.get(filePath))
        .build();
    FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);
    fileUpload.completionFuture().join();
}
```

```
transferManager.close();  
}
```

Use a [API S3Client](#) ou (API S3AsyncClient) para realizar um carregamento fracionado. Se você especificar uma soma de verificação adicional, deverá indicar o algoritmo a ser usado no início do carregamento. Você também deve especificar o algoritmo para a solicitação de cada parte e fornecer a soma de verificação calculada para cada parte após o carregamento.

```
public void multipartUploadWithChecksumS3Client(String filePath) {  
    ChecksumAlgorithm algorithm = ChecksumAlgorithm.CRC32;  
  
    // Initiate the multipart upload.  
    CreateMultipartUploadResponse createMultipartUploadResponse =  
s3Client.createMultipartUpload(b -> b  
        .bucket(bucketName)  
        .key(key)  
        .checksumAlgorithm(algorithm)); // Checksum specified on  
initiation.  
    String uploadId = createMultipartUploadResponse.uploadId();  
  
    // Upload the parts of the file.  
    int partNumber = 1;  
    List<CompletedPart> completedParts = new ArrayList<>();  
    ByteBuffer bb = ByteBuffer.allocate(1024 * 1024 * 5); // 5 MB byte buffer  
  
    try (RandomAccessFile file = new RandomAccessFile(filePath, "r")) {  
        long fileSize = file.length();  
        long position = 0;  
        while (position < fileSize) {  
            file.seek(position);  
            long read = file.getChannel().read(bb);  
  
            bb.flip(); // Swap position and limit before reading from the  
buffer.  
  
            UploadPartRequest uploadPartRequest = UploadPartRequest.builder()  
                .bucket(bucketName)  
                .key(key)  
                .uploadId(uploadId)  
                .checksumAlgorithm(algorithm) // Checksum specified on  
each part.  
                .partNumber(partNumber)  
                .build();
```

```
UploadPartResponse partResponse = s3Client.uploadPart(
    uploadPartRequest,
    RequestBody.fromByteBuffer(bb));

CompletedPart part = CompletedPart.builder()
    .partNumber(partNumber)
    .checksumCRC32(partResponse.checksumCRC32()) // Provide
the calculated checksum.
    .eTag(partResponse.eTag())
    .build();
completedParts.add(part);

bb.clear();
position += read;
partNumber++;
}
} catch (IOException e) {
    System.err.println(e.getMessage());
}

// Complete the multipart upload.
s3Client.completeMultipartUpload(b -> b
    .bucket(bucketName)
    .key(key)
    .uploadId(uploadId)

.multipartUpload(CompletedMultipartUpload.builder().parts(completedParts).build()));
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for Java 2.x.
 - [CompleteMultipartUpload](#)
 - [CreateMultipartUpload](#)
 - [UploadPart](#)

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). 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 exemplo de código a seguir mostra como:

- Criar um bucket do S3 versionado.
- Obter todas as versões de um objeto.
- Reverter um objeto para uma versão anterior.
- Excluir e restaurar um objeto versionado.
- Exclua, permanentemente, todas as versões de um objeto.

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

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 AWS SDK para Python (Boto3).
 - [CreateBucket](#)
 - [DeleteObject](#)
 - [ListObjectVersions](#)
 - [PutBucketLifecycleConfiguration](#)

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos sem servidor para o Amazon S3 usando AWS SDKs

Os exemplos de código a seguir mostram como usar o Amazon S3 com SDKs da AWS.

Exemplos

- [Invocar uma função do Lambda em um acionador do Amazon S3](#)

Invocar uma função do Lambda em um acionador do Amazon S3

Os exemplos de código a seguir mostram como implementar uma função do Lambda que recebe um evento acionado pelo upload de um objeto em um bucket do S3. A função recupera o nome do bucket do S3 e a chave do objeto do parâmetro de evento e chama a API do Amazon S3 para recuperar e registrar o tipo de conteúdo do objeto.

.NET

AWS SDK for .NET

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório dos [Exemplos sem servidor](#).

Consumir um evento do S3 com o Lambda usando .NET.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
using System.Threading.Tasks;
using Amazon.Lambda.Core;
using Amazon.S3;
using System;
using Amazon.Lambda.S3Events;
using System.Web;

// Assembly attribute to enable the Lambda function's JSON input to be converted
// into a .NET class.
[assembly: LambdaSerializer(typeof(Amazon.Lambda.Serialization.SystemTextJson.DefaultLambdaJsonSerializer))]

namespace S3Integration
{
    public class Function
    {
        private static AmazonS3Client _s3Client;
        public Function() : this(null)
        {
        }

        internal Function(AmazonS3Client s3Client)
        {
            _s3Client = s3Client ?? new AmazonS3Client();
        }

        public async Task<string> Handler(S3Event evt, ILambdaContext context)
        {
            try
```

```
    {
        if (evt.Records.Count <= 0)
        {
            context.Logger.LogLine("Empty S3 Event received");
            return string.Empty;
        }

        var bucket = evt.Records[0].S3.Bucket.Name;
        var key = HttpUtility.UrlDecode(evt.Records[0].S3.Object.Key);

        context.Logger.LogLine($"Request is for {bucket} and {key}");

        var objectResult = await _s3Client.GetObjectAsync(bucket, key);

        context.Logger.LogLine($"Returning {objectResult.Key}");

        return objectResult.Key;
    }
    catch (Exception e)
    {
        context.Logger.LogLine($"Error processing request -
{e.Message}");

        return string.Empty;
    }
}
}
```

Go

SDK para Go V2

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório dos [Exemplos sem servidor](#).

Consumir um evento do S3 com o Lambda usando Go.

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

```
// SPDX-License-Identifier: Apache-2.0
package main

import (
    "context"
    "log"

    "github.com/aws/aws-lambda-go/events"
    "github.com/aws/aws-lambda-go/lambda"
    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/s3"
)

func handler(ctx context.Context, s3Event events.S3Event) error {
    sdkConfig, err := config.LoadDefaultConfig(ctx)
    if err != nil {
        log.Printf("failed to load default config: %s", err)
        return err
    }
    s3Client := s3.NewFromConfig(sdkConfig)

    for _, record := range s3Event.Records {
        bucket := record.S3.Bucket.Name
        key := record.S3.Object.URLDecodedKey
        headOutput, err := s3Client.HeadObject(ctx, &s3.HeadObjectInput{
            Bucket: &bucket,
            Key:    &key,
        })
        if err != nil {
            log.Printf("error getting head of object %s/%s: %s", bucket, key, err)
            return err
        }
        log.Printf("successfully retrieved %s/%s of type %s", bucket, key,
            *headOutput.ContentType)
    }

    return nil
}

func main() {
    lambda.Start(handler)
}
```


Java

SDK para Java 2.x

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório dos [Exemplos sem servidor](#).

Consumir um evento do S3 com o Lambda usando Java.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
package example;

import software.amazon.awssdk.services.s3.model.HeadObjectRequest;
import software.amazon.awssdk.services.s3.model.HeadObjectResponse;
import software.amazon.awssdk.services.s3.S3Client;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;
import com.amazonaws.services.lambda.runtime.events.S3Event;
import
    com.amazonaws.services.lambda.runtime.events.models.s3.S3EventNotification.S3EventNotifi

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class Handler implements RequestHandler<S3Event, String> {
    private static final Logger logger = LoggerFactory.getLogger(Handler.class);
    @Override
    public String handleRequest(S3Event s3event, Context context) {
        try {
            S3EventNotificationRecord record = s3event.getRecords().get(0);
            String srcBucket = record.getS3().getBucket().getName();
            String srcKey = record.getS3().getObject().getUrlDecodedKey();

            S3Client s3Client = S3Client.builder().build();
            HeadObjectResponse headObject = getHeadObject(s3Client, srcBucket,
srcKey);
```

```
        logger.info("Successfully retrieved " + srcBucket + "/" + srcKey + " of
type " + headObject.contentType());

        return "Ok";
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}

private HeadObjectResponse getHeadObject(S3Client s3Client, String bucket,
String key) {
    HeadObjectRequest headObjectRequest = HeadObjectRequest.builder()
        .bucket(bucket)
        .key(key)
        .build();
    return s3Client.headObject(headObjectRequest);
}
}
```

JavaScript

SDK para JavaScript (v3)

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório dos [Exemplos sem servidor](#).

Consumir um evento do S3 com o Lambda usando JavaScript.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { S3Client, HeadObjectCommand } from "@aws-sdk/client-s3";

const client = new S3Client();

exports.handler = async (event, context) => {

    // Get the object from the event and show its content type
    const bucket = event.Records[0].s3.bucket.name;
```

```
const key = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, ' '));

try {
  const { ContentType } = await client.send(new HeadObjectCommand({
    Bucket: bucket,
    Key: key,
  }));

  console.log('CONTENT TYPE:', ContentType);
  return ContentType;

} catch (err) {
  console.log(err);
  const message = `Error getting object ${key} from bucket ${bucket}. Make sure they exist and your bucket is in the same region as this function.`;
  console.log(message);
  throw new Error(message);
}
};
```

Consumir um evento do S3 com o Lambda usando TypeScript.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { S3Event } from 'aws-lambda';
import { S3Client, HeadObjectCommand } from '@aws-sdk/client-s3';

const s3 = new S3Client({ region: process.env.AWS_REGION });

export const handler = async (event: S3Event): Promise<string | undefined> => {
  // Get the object from the event and show its content type
  const bucket = event.Records[0].s3.bucket.name;
  const key = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, ' '));
  const params = {
    Bucket: bucket,
    Key: key,
  };
  try {
    const { ContentType } = await s3.send(new HeadObjectCommand(params));
    console.log('CONTENT TYPE:', ContentType);
  }
};
```

```
    return ContentType;
  } catch (err) {
    console.log(err);
    const message = `Error getting object ${key} from bucket ${bucket}. Make sure
they exist and your bucket is in the same region as this function.`;
    console.log(message);
    throw new Error(message);
  }
};
```

PHP

SDK para PHP

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório dos [Exemplos sem servidor](#).

Como consumir um evento do S3 com o Lambda usando PHP.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
<?php

use Bref\Context\Context;
use Bref\Event\S3\S3Event;
use Bref\Event\S3\S3Handler;
use Bref\Logger\StderrLogger;

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

class Handler extends S3Handler
{
    private StderrLogger $logger;
    public function __construct(StderrLogger $logger)
    {
        $this->logger = $logger;
    }
}
```

```
public function handleS3(S3Event $event, Context $context) : void
{
    $this->logger->info("Processing S3 records");

    // Get the object from the event and show its content type
    $records = $event->getRecords();

    foreach ($records as $record)
    {
        $bucket = $record->getBucket()->getName();
        $key = urldecode($record->getObject()->getKey());

        try {
            $fileSize = urldecode($record->getObject()->getSize());
            echo "File Size: " . $fileSize . "\n";
            // TODO: Implement your custom processing logic here
        } catch (Exception $e) {
            echo $e->getMessage() . "\n";
            echo 'Error getting object ' . $key . ' from bucket ' .
            $bucket . '. Make sure they exist and your bucket is in the same region as this
            function.' . "\n";
            throw $e;
        }
    }
}

$logger = new StderrLogger();
return new Handler($logger);
```

Python

SDK para Python (Boto3).

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório dos [Exemplos sem servidor](#).

Consumir um evento do S3 com o Lambda usando Python.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
import json
import urllib.parse
import boto3

print('Loading function')

s3 = boto3.client('s3')

def lambda_handler(event, context):
    #print("Received event: " + json.dumps(event, indent=2))

    # Get the object from the event and show its content type
    bucket = event['Records'][0]['s3']['bucket']['name']
    key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'],
    encoding='utf-8')
    try:
        response = s3.get_object(Bucket=bucket, Key=key)
        print("CONTENT TYPE: " + response['ContentType'])
        return response['ContentType']
    except Exception as e:
        print(e)
        print('Error getting object {} from bucket {}. Make sure they exist and
        your bucket is in the same region as this function.'.format(key, bucket))
        raise e
```

Ruby

SDK para Ruby

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório dos [Exemplos sem servidor](#).

Como consumir um evento do S3 com o Lambda usando Ruby.

```
require 'json'
require 'uri'
require 'aws-sdk'

puts 'Loading function'

def lambda_handler(event:, context:)
  s3 = Aws::S3::Client.new(region: 'region') # Your AWS region
  # puts "Received event: #{JSON.dump(event)}"

  # Get the object from the event and show its content type
  bucket = event['Records'][0]['s3']['bucket']['name']
  key = URI.decode_www_form_component(event['Records'][0]['s3']['object']['key'],
  Encoding::UTF_8)
  begin
    response = s3.get_object(bucket: bucket, key: key)
    puts "CONTENT TYPE: #{response.content_type}"
    return response.content_type
  rescue StandardError => e
    puts e.message
    puts "Error getting object #{key} from bucket #{bucket}. Make sure they exist
    and your bucket is in the same region as this function."
    raise e
  end
end
```

Rust

SDK para Rust

Note

Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório dos [Exemplos sem servidor](#).

Consumir um evento do S3 com o Lambda usando Rust.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
```

```
use aws_lambda_events::event::s3::S3Event;
use aws_sdk_s3::{Client};
use lambda_runtime::{run, service_fn, Error, LambdaEvent};

/// Main function
#[tokio::main]
async fn main() -> Result<(), Error> {
    tracing_subscriber::fmt()
        .with_max_level(tracing::Level::INFO)
        .with_target(false)
        .without_time()
        .init();

    // Initialize the AWS SDK for Rust
    let config = aws_config::load_from_env().await;
    let s3_client = Client::new(&config);

    let res = run(service_fn(|request: LambdaEvent<S3Event>| {
        function_handler(&s3_client, request)
    })).await;

    res
}

async fn function_handler(
    s3_client: &Client,
    evt: LambdaEvent<S3Event>
) -> Result<(), Error> {
    tracing::info!(records = ?evt.payload.records.len(), "Received request from
    SQS");

    if evt.payload.records.len() == 0 {
        tracing::info!("Empty S3 event received");
    }

    let bucket = evt.payload.records[0].s3.bucket.name.as_ref().expect("Bucket
    name to exist");
    let key = evt.payload.records[0].s3.object.key.as_ref().expect("Object key to
    exist");

    tracing::info!("Request is for {} and object {}", bucket, key);

    let s3_get_object_result = s3_client
```



```
.get_object()
.bucket(bucket)
.key(key)
.send()
.await;

match s3_get_object_result {
  Ok(_) => tracing::info!("S3 Get Object success, the s3GetObjectResult
contains a 'body' property of type ByteStream"),
  Err(_) => tracing::info!("Failure with S3 Get Object request")
}

Ok(())
}
```

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). 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, em que é possível encontrar instruções sobre como configurar e executar a aplicação.

Exemplos

- [Criar uma aplicação Amazon Transcribe](#)
- [Converter texto em fala e de volta em texto usando um AWS SDK](#)
- [Criar uma aplicação de gerenciamento de ativos de fotos que permita que os usuários gerenciem fotos usando rótulos](#)
- [Criar uma aplicação de exploração do Amazon Textract](#)
- [Detectar EPI em imagens com o Amazon Rekognition usando um AWS SDK](#)
- [Detectar entidades em texto extraído de uma imagem usando um AWS SDK](#)
- [Detectar faces em uma imagem usando um AWS SDK](#)
- [Detectar objetos em imagens com o Amazon Rekognition usando um AWS SDK](#)
- [Detectar pessoas e objetos em um vídeo com o Amazon Rekognition usando um AWS SDK](#)
- [Salvar o EXIF e outras informações de imagem usando um AWS SDK](#)

- [Como transformar dados para sua aplicação com o S3 Object Lambda](#)

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 utilizados neste exemplo

- Identidade do Amazon Cognito
- Amazon S3
- Amazon Transcribe

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). 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 exemplo de código a seguir mostra como:

- Usar o Amazon Polly para sintetizar um arquivo de entrada de texto simples (UTF-8) para um arquivo de áudio.

- Fazer upload do arquivo de áudio para um bucket do Amazon S3.
- Usar o Amazon Transcribe para converter o arquivo de áudio em texto.
- Exibir o texto.

Rust

SDK para Rust

Use o Amazon Polly para sintetizar um arquivo de texto simples (UTF-8) para um arquivo de áudio, fazer upload do arquivo de áudio para um bucket do Amazon S3, 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 utilizados neste exemplo

- Amazon Polly
- Amazon S3
- Amazon Transcribe

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Criar uma aplicação de gerenciamento de ativos de fotos que permita que os usuários gerenciem fotos usando rótulos

O exemplo de código a seguir mostra como criar uma aplicação com tecnologia sem servidor que permite que os usuários gerenciem fotos usando rótulos.

.NET

AWS SDK for .NET

Mostra como desenvolver uma aplicação de gerenciamento de ativos fotográficos que detecta rótulos em imagens usando o Amazon Rekognition e os armazena para recuperação posterior.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Para uma análise detalhada da origem desse exemplo, veja a publicação na [Comunidade da AWS](#).

Serviços utilizados neste exemplo

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

C++

SDK para C++

Mostra como desenvolver uma aplicação de gerenciamento de ativos fotográficos que detecta rótulos em imagens usando o Amazon Rekognition e os armazena para recuperação posterior.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Para uma análise detalhada da origem desse exemplo, veja a publicação na [Comunidade da AWS](#).

Serviços utilizados neste exemplo

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Java

SDK para Java 2.x

Mostra como desenvolver uma aplicação de gerenciamento de ativos fotográficos que detecta rótulos em imagens usando o Amazon Rekognition e os armazena para recuperação posterior.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Para uma análise detalhada da origem desse exemplo, veja a publicação na [Comunidade da AWS](#).

Serviços utilizados neste exemplo

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

JavaScript

SDK para JavaScript (v3)

Mostra como desenvolver uma aplicação de gerenciamento de ativos fotográficos que detecta rótulos em imagens usando o Amazon Rekognition e os armazena para recuperação posterior.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Para uma análise detalhada da origem desse exemplo, veja a publicação na [Comunidade da AWS](#).

Serviços utilizados neste exemplo

- API Gateway

- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Kotlin

SDK for Kotlin

Mostra como desenvolver uma aplicação de gerenciamento de ativos fotográficos que detecta rótulos em imagens usando o Amazon Rekognition e os armazena para recuperação posterior.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Para uma análise detalhada da origem desse exemplo, veja a publicação na [Comunidade da AWS](#).

Serviços utilizados neste exemplo

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

PHP

SDK para PHP

Mostra como desenvolver uma aplicação de gerenciamento de ativos fotográficos que detecta rótulos em imagens usando o Amazon Rekognition e os armazena para recuperação posterior.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Para uma análise detalhada da origem desse exemplo, veja a publicação na [Comunidade da AWS](#).

Serviços utilizados neste exemplo

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Rust

SDK para Rust

Mostra como desenvolver uma aplicação de gerenciamento de ativos fotográficos que detecta rótulos em imagens usando o Amazon Rekognition e os armazena para recuperação posterior.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Para uma análise detalhada da origem desse exemplo, veja a publicação na [Comunidade da AWS](#).

Serviços utilizados neste exemplo

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Para obter uma lista completa dos Guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). 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

- Identidade do Amazon Cognito
- 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 um aplicativo 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](#). 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 AWS SDK e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). 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 caderno 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 caderno 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 utilizados 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](#). 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 exemplo de código a seguir mostra como:

- Salvar uma imagem em um bucket do Amazon S3.
- Usar o Amazon Rekognition para detectar detalhes faciais, como faixa etária, gênero e emoções (sorriso, etc.).

- Exibir esses detalhes.

Rust

SDK para Rust

Salve a imagem em um bucket do Amazon S3 com um prefixo uploads, use o Amazon Rekognition para detectar detalhes faciais, como faixa etária, gênero 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 AWS SDK e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). 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 for Kotlin

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 AWS SDK e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). 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 AWS SDK e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). 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 exemplo de código a seguir mostra como:

- Obter informações de EXIF de um arquivo JPG, JPEG ou PNG.
- Fazer upload do arquivo de imagem para um bucket do Amazon S3.
- Usar o Amazon Rekognition para identificar os três principais atributos (rótulos) no arquivo.
- Adicione as informações de EXIF e rótulo a uma tabela do Amazon DynamoDB na região.

Rust

SDK para Rust

Obtenha informações de EXIF de um arquivo JPG, JPEG ou PNG, faça upload do arquivo de imagem para um bucket do Amazon S3, 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 AWS SDK e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Como transformar dados para sua aplicação com o S3 Object Lambda

O exemplo de código a seguir mostra como transformar dados para sua aplicação com o S3 Object Lambda.

.NET

AWS SDK for .NET

Mostra como adicionar código personalizado a solicitações GET padrão do S3 para modificar o objeto solicitado e recuperado do S3 e possibilitar que o objeto atenda às necessidades do cliente ou aplicação solicitante.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no [GitHub](#).

Serviços utilizados neste exemplo

- Lambda
- Amazon S3

Para obter uma lista completa dos Guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usar este serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Solução de problemas

Esta seção descreve como solucionar problemas de recursos do Amazon S3 e explica como obter os IDs de solicitação necessários quando você entrar em contato com o AWS Support.

Tópicos

- [Solucionar erros de acesso negado \(403 proibido\) no Amazon S3](#)
- [Solução de problemas com operações em lote](#)
- [Solução de problemas do CORS](#)
- [Solução de problemas do ciclo de vida do Amazon S3](#)
- [Solução de problemas de replicação](#)
- [Solução de problemas do registro em log de acessos ao servidor](#)
- [Solucionar problemas de versionamento](#)
- [Obter IDs de solicitação do Amazon S3 para o AWS Support](#)

Solucionar erros de acesso negado (403 proibido) no Amazon S3

Important

Em 13 de maio de 2024, começamos a implantar uma alteração para eliminar cobranças por solicitações não autorizadas que não foram iniciadas pelo proprietário do bucket. Depois que a implantação dessa alteração for concluída, os proprietários do bucket nunca incorrerão em cobranças de solicitação ou largura de banda para solicitações que retornem erros AccessDenied (HTTP 403 Forbidden) quando forem iniciadas de fora de sua conta da AWS individual ou de organização da AWS. Consulte uma lista completa de códigos de status HTTP 3XX e 4XX que não serão cobrados em [Cobrança pelas respostas de erro do Amazon S3](#). Essa alteração de cobrança não exige atualizações nas aplicações e é destinada a todos os buckets do S3. Quando a implantação dessa alteração for concluída em todas as Regiões da AWS, atualizaremos nossa documentação.

Os tópicos a seguir abordam as causas mais comuns dos erros de acesso negado (403 proibido) no Amazon S3.

Note

Para Access Denied (HTTP 403 Forbidden), o S3 não cobra do proprietário do bucket quando a solicitação é iniciada fora da conta da AWS individual dele ou da organização da AWS à qual ele pertence.

Tópicos

- [Políticas de bucket e políticas do IAM](#)
- [Configurações de ACL do Amazon S3](#)
- [Configurações do bloqueio de acesso público do S3](#)
- [Configurações de criptografia do Amazon S3](#)
- [Configurações do bloqueio de objetos do S3](#)
- [Política de endpoint da VPC](#)
- [Políticas do AWS Organizations](#)
- [Configurações do ponto de acesso](#)

Note

Se você estiver tentando solucionar um problema de permissões, comece com a seção [Políticas de bucket e políticas do IAM](#) e siga as orientações em [Dicas para verificar permissões](#).

Políticas de bucket e políticas do IAM

Operações no nível do bucket

Se não houver uma política de bucket em vigor, o bucket permitirá implicitamente solicitações de qualquer identidade do AWS Identity and Access Management (IAM) na conta proprietária do bucket. O bucket também nega implicitamente solicitações de qualquer outra identidade do IAM de qualquer outra conta e solicitações anônimas (não assinadas). No entanto, se não houver uma política de usuário do IAM em vigor, o solicitante (a menos que seja o usuário raiz) será implicitamente impedido de fazer qualquer solicitação. Para obter mais informações sobre essa lógica de avaliação, consulte [Determinar se uma solicitação é negada ou permitida em uma conta](#) no Guia do usuário do IAM.

Operações no nível do objeto

Se o objeto pertencer à conta proprietária do bucket, a política de bucket e a política de usuário do IAM funcionarão da mesma forma para operações no nível do objeto e para operações no nível do bucket. Por exemplo, se não houver uma política de bucket em vigor, o bucket permitirá implicitamente solicitações de objeto de qualquer identidade do IAM na conta proprietária do bucket. O bucket também nega implicitamente solicitações de objeto de qualquer outra identidade do IAM de qualquer outra conta e solicitações anônimas (não assinadas). No entanto, se não houver uma política de usuário do IAM em vigor, o solicitante (a menos que seja o usuário raiz) será implicitamente impedido de fazer qualquer solicitação de objeto.

Se o objeto pertencer a uma conta externa, o acesso ao objeto só poderá ser concedido por meio de listas de controle de acesso (ACLs) do objeto. A política de bucket e a política de usuário do IAM ainda podem ser usadas para negar solicitações de objeto.

Portanto, para garantir que a política de bucket ou a política de usuário do IAM não esteja causando um erro de acesso negado (403 proibido), verifique se os seguintes requisitos são atendidos:

- Para acesso à mesma conta, não deve haver uma declaração Deny explícita contra o solicitante ao qual você está tentando conceder permissões, seja na política de bucket ou na política de usuário do IAM. Se você quiser conceder permissões usando somente a política de bucket e a política de usuário do IAM, deve haver pelo menos uma declaração Allow explícita em uma dessas políticas.
- Para acesso entre contas, não deve haver uma declaração Deny explícita contra o solicitante ao qual você está tentando conceder permissões, seja na política de bucket ou na política de usuário do IAM. Se você quiser conceder permissões entre contas usando somente a política de bucket e a política de usuário do IAM, ambas as políticas do solicitante devem incluir uma declaração Allow explícita.

Note

As declarações Allow em uma política de bucket se aplicam somente a objetos que [pertencem à mesma conta proprietária do bucket](#). No entanto, as declarações Deny em uma política de bucket se aplicam a todos os objetos, independentemente da propriedade dele.

Para revisar ou editar a política de bucket

Note

Para visualizar ou editar uma política de bucket, é necessário ter a permissão `s3:GetBucketPolicy`.

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 para o qual você deseja visualizar ou editar uma política de bucket.
4. Escolha a aba Permissões.
5. Em Bucket policy (Política de bucket), escolha Edit (Editar). A página Edit bucket policy (Editar política de bucket) é exibida.

Para revisar ou editar a política de bucket usando a AWS Command Line Interface (AWS CLI), use o comando [get-bucket-policy](#).

Note

Se você for bloqueado de um bucket devido a uma política de bucket incorreta, [faça login no AWS Management Console usando as credenciais de usuário raiz](#). Para recuperar o acesso ao bucket, exclua a política de bucket usando as credenciais de usuário raiz.

Dicas para verificar permissões

Para verificar se o solicitante tem as permissões adequadas para realizar uma operação do Amazon S3, tente o seguinte:

- Identifique o solicitante. Se for uma solicitação não assinada, será uma solicitação anônima sem uma política de usuário do IAM. Se for uma solicitação usando um URL pré-assinado, a política do usuário será a mesma do perfil ou usuário do IAM que assinou a solicitação.

- Verifique se você está usando o perfil ou usuário do IAM correto. É possível verificar o perfil ou usuário do IAM no canto superior direito do AWS Management Console ou usando o comando [aws sts get-caller-identity](#).
- Verifique as políticas do IAM relacionadas ao perfil ou usuário do IAM. É possível usar um dos seguintes métodos:
 - [Testar as políticas do IAM com o simulador de políticas do IAM](#)
 - Analise os diferentes [tipos de política do IAM](#).
- Se necessário, [edite a política de usuário do IAM](#).
- Veja os seguintes exemplos de políticas que negam ou permitem explicitamente o acesso:
 - Política de usuário do IAM de permissão explícita: [IAM: permite e nega acesso a vários serviços de forma programática e no console](#)
 - Política de bucket de permissão explícita: [Conceder permissões a várias contas para fazer upload de objetos ou definir ACLs de objetos para acesso público](#)
 - Política de usuário do IAM de negação explícita: [AWS: negar acesso à AWS com base na Região da AWS solicitada](#)
 - Política de bucket de negação explícita: [Exigir SSE-KMS para todos os objetos gravados em um bucket](#)

Configurações de ACL do Amazon S3

Ao verificar as configurações de ACL, primeiro [revise a configuração de propriedade de objetos](#) para verificar se as ACLs estão ativadas no bucket. Esteja ciente de que as permissões de ACL podem ser usadas somente para conceder permissões e não podem ser usadas para rejeitar solicitações. As ACLs também não podem ser usadas para conceder acesso a solicitantes que são rejeitados por negações explícitas nas políticas de bucket ou nas políticas de usuário do IAM.

A configuração de propriedade de objetos está definida como o proprietário do bucket aplicado

Se a configuração de proprietário do bucket aplicado estiver ativada, é improvável que as configurações de ACL causem um erro de acesso negado (403 proibido), pois essa configuração desativa todas as ACLs que se aplicam ao bucket e aos objetos. O proprietário de bucket aplicado é a configuração padrão (e recomendada) para os buckets do Amazon S3.

A configuração de propriedade de objetos está definida como o proprietário de bucket de preferência ou o gravador de objetos

As permissões de ACL ainda são válidas com a configuração do proprietário de bucket de preferência ou com a configuração do gravador de objetos. Há dois tipos de ACLs: ACLs de bucket e ACLs de objeto. Para ver as diferenças entre esses dois tipos de ACLs, consulte [Mapeamento das permissões da ACL e das permissões da política de acesso](#).


Dependendo da ação da solicitação rejeitada, [verifique as permissões de ACL para o bucket ou objeto](#):

- Se o Amazon S3 rejeitou uma LIST, um objeto PUT, GetBucketAcl ou uma solicitação PutBucketAcl, [revise as permissões de ACL para o bucket](#).

 Note

Você não pode conceder permissões de objeto GET com as configurações de ACL do bucket.

- Se o Amazon S3 rejeitou uma solicitação GET em um objeto do S3 ou uma solicitação PutObjectAcl, [revise as permissões de ACL para o objeto](#).

 Important


Se a conta da que possui o objeto for diferente da conta da que possui o bucket, o acesso ao objeto não será controlado pela política de bucket.

Solução de problemas de um erro de acesso negado (403 proibido) de uma solicitação de objeto **GET** durante a propriedade de objetos entre contas

Examine as [configurações de propriedade de objetos](#) do bucket para determinar o proprietário do objeto. Se você tiver acesso às [ACLs do objeto](#), também poderá verificar a conta do proprietário do objeto. (Para visualizar a conta do proprietário do objeto, revise a configuração da ACL do objeto no console do Amazon S3.) Como alternativa, você também pode fazer uma solicitação GetObjectAcl a fim de encontrar a [ID canônica](#) do proprietário do objeto para verificar a conta do proprietário do objeto. Por padrão, as ACLs concedem permissões explícitas para solicitações GET à conta do proprietário do objeto.

Depois de confirmar que o proprietário do objeto é diferente do proprietário do bucket, dependendo do caso de uso e nível de acesso, escolha um dos seguintes métodos para ajudar a resolver o erro Acesso negado (403 proibido):

- Desative as ACLs (recomendado): esse método se aplicará a todos os objetos e poderá ser executado pelo proprietário do bucket. Esse método concede automaticamente ao proprietário do bucket propriedade e controle total de todos os objetos do bucket. Antes de implementar esse método, verifique os [pré-requisitos para desativar as ACLs](#). Para obter informações sobre como configurar o bucket no modo proprietário do bucket aplicado (recomendado), consulte [Configurar Object Ownership em um bucket existente](#).

 Important

Para evitar um erro de acesso negado (403 proibido), migre as permissões da ACL para uma política de bucket antes de desativar as ACLs. Para obter mais informações, consulte [Exemplos de políticas de bucket para migrar de permissões de ACL](#).

- Altere o proprietário do objeto para o proprietário do bucket: esse método pode ser aplicado a objetos individuais, mas somente o proprietário do objeto (ou um usuário com as permissões apropriadas) pode alterar a propriedade de um objeto. Custos adicionais de PUT podem ser aplicados. (Para obter mais informações, consulte [Preço do Amazon S3](#).) Esse método concede ao proprietário do bucket a propriedade total do objeto, permitindo que ele controle o acesso ao objeto por meio de uma política de bucket.

Para alterar a propriedade do objeto, siga um destes procedimentos:

- Você (o proprietário do bucket) pode [copiar o objeto](#) de volta para o bucket.
- É possível alterar a configuração de propriedade de objetos do bucket para a configuração de preferência do proprietário do bucket. Se o versionamento estiver desativado, os objetos no bucket serão substituídos. Se o versionamento estiver ativado, versões duplicadas do mesmo objeto aparecerão no bucket, e o proprietário do bucket poderá [definir uma regra de ciclo de vida para expirar](#). Para obter instruções sobre como alterar a configuração de propriedade de objetos, consulte [Configurar Object Ownership em um bucket existente](#).

Note

Quando você atualiza a configuração de propriedade de objetos para a de preferência do proprietário do bucket, a configuração só é aplicada a novos objetos que são enviados ao bucket.

- É possível fazer com que o proprietário do objeto carregue o objeto novamente com a ACL `bucket-owner-full-control` do objeto predefinido.

Note

Para uploads entre contas, você também pode exigir a ACL `bucket-owner-full-control` do objeto predefinido na política de bucket. Para obter um exemplo de política de bucket, consulte [Conceder permissões entre contas para fazer upload de objetos garantindo que o proprietário do bucket tenha controle total](#).

- Mantenha o gravador do objeto como proprietário do objeto: esse método não altera o proprietário do objeto, mas permite que você conceda acesso aos objetos individualmente. Para conceder acesso a um objeto, você deve ter a permissão `PutObjectACL` para o objeto. Depois, para corrigir o erro Acesso negado (403 proibido), adicione o solicitante como [beneficiário](#) para acessar o objeto nas ACLs do objeto. Para ter mais informações, consulte [Configurar ACLs](#).

Configurações do bloqueio de acesso público do S3

Se a solicitação com falha envolver acesso público ou políticas públicas, verifique as configurações do bloqueio de acesso público do S3 na conta, no bucket ou no ponto de acesso do S3. A partir de abril de 2023, todas as configurações de bloqueio de acesso público são habilitadas por padrão para novos buckets. Para obter mais informações sobre como o Amazon S3 define “público”, consulte [O significado de "público"](#).

Quando definidas como TRUE, as configurações do bloqueio de acesso público atuam como políticas de negação explícitas que substituem as permissões por ACLs, políticas de bucket e políticas de usuário do IAM. Para determinar se as configurações de bloqueio de acesso público estão rejeitando sua solicitação, analise os seguintes cenários:

- Se a lista de controle de acesso (ACL) especificada for pública, a configuração `BlockPublicAcls` rejeitará as chamadas `PutBucketACL` e `PutObjectACL`.

- Se a solicitação incluir uma ACL pública, a configuração `BlockPublicAcls` rejeitará as chamadas `PutObject`.
- Se a configuração `BlockPublicAcls` for aplicada a uma conta e a solicitação incluir uma ACL pública, todas as chamadas `CreateBucket` que incluam ACLs públicas falharão.
- Se a permissão da solicitação for concedida somente por uma ACL pública, a configuração `IgnorePublicAcls` rejeitará a solicitação.
- Se a política de bucket especificada permitir acesso público, a configuração `BlockPublicPolicy` rejeitará as chamadas `PutBucketPolicy`.
- Se a configuração `BlockPublicPolicy` for aplicada a um ponto de acesso, todas as chamadas `PutAccessPointPolicy` e `PutBucketPolicy` que especificam uma política pública e são feitas por meio do ponto de acesso falharão.
- Se o ponto de acesso ou o bucket tiver uma política pública, a configuração `RestrictPublicBuckets` rejeitará todas as chamadas entre contas, exceto para entidades principais do AWS service (Serviço da AWS). Essa configuração também rejeita todas as chamadas anônimas (ou não assinadas).

Para revisar e atualizar as configurações de bloqueio de acesso público, consulte [Configurar o bloqueio de acesso público para seus buckets do S3](#).

Configurações de criptografia do Amazon S3

O Amazon S3 é compatível com a criptografia do lado do servidor no bucket. 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 da AWS e descriptografa-os quando você os acessa.

Por padrão, o Amazon S3 agora aplica a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) como nível básico de criptografia para cada bucket no Amazon S3. O Amazon S3 também permite que você especifique o método de criptografia do lado do servidor ao fazer upload de objetos.

Para revisar as configurações de criptografia e o status da criptografia do lado do servidor do 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 Buckets, escolha o bucket para o qual você deseja verificar as configurações de criptografia.
4. Escolha a guia Properties (Propriedades).
5. Role para baixo até a seção Criptografia padrão e veja as configurações de Tipo de criptografia.

Para verificar as configurações de criptografia usando a AWS CLI, use o comando [get-bucket-encryption](#).

Para verificar o status de 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. No painel de navegação à esquerda, escolha Buckets.
3. Na lista Buckets, escolha o nome do bucket que contém o objeto.
4. Na lista Objetos, escolha o nome do objeto ao qual você deseja adicionar ou no qual deseja alterar a criptografia.

A página de detalhes do objeto é exibida.

5. Role para baixo até a seção Configurações de criptografia do lado do servidor para ver as configurações de criptografia do lado do servidor do objeto.

Para verificar o status de criptografia do objeto usando a AWS CLI, use o comando [head-object](#).

Requisitos de permissões e criptografia

O Amazon S3 agora é compatível com três tipos de criptografia do lado do servidor.

- Criptografia do 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)

Com base nas configurações de criptografia, verifique se os seguintes requisitos de permissões são atendidos:

- SSE-S3: nenhuma permissão extra é necessária.

- SSE-KMS (com uma chave gerenciada pelo cliente): para fazer upload de objetos, é necessária a permissão `kms:GenerateDataKey` no AWS KMS key. Para baixar objetos e fazer uploads de objetos em várias partes, é necessária a permissão `kms:Decrypt` na chave do KMS.
- SSE-KMS (com um Chave gerenciada pela AWS): o solicitante deve ser da mesma conta que possui a chave do KMS `aws/s3`. O solicitante também deve ter as permissões corretas do Amazon S3 para acessar o objeto.
- SSE-C (com uma chave fornecida pelo cliente): nenhuma permissão adicional é necessária. É possível configurar a política de bucket para [exigir e restringir a criptografia do lado do servidor com chaves de criptografia fornecidas pelo cliente](#) para objetos no bucket.

Se o objeto estiver criptografado com uma chave gerenciada pelo cliente, certifique-se de que a política de chaves do KMS permita que você realize as ações `kms:GenerateDataKey` ou `kms:Decrypt`. Para obter instruções sobre como verificar a política de chaves do KMS, consulte [Visualizar uma política de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

Configurações do bloqueio de objetos do S3

Se o bucket tiver o [bloqueio de objetos do S3](#) ativado e o objeto estiver protegido por um [período de retenção](#) ou uma [retenção legal](#), o Amazon S3 retornará um erro de acesso negado (403 proibido) quando você tentar excluir o objeto.

Para verificar se o bucket tem o bloqueio de objetos ativado

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 deseja revisar.
4. Escolha a guia Properties (Propriedades).
5. Role para baixo até a seção Bloqueio de objetos. Verifique se a configuração de Bloqueio de objetos está Ativada ou Desativada.

Para determinar se o objeto está protegido por um período de retenção ou uma retenção legal, [veja as informações de bloqueio](#) do objeto.

Se o objeto estiver protegido por um período de retenção ou uma retenção legal, verifique o seguinte:

- Se a versão do objeto estiver protegida pelo modo de retenção de conformidade, não há como excluí-lo permanentemente. Uma solicitação DELETE permanente de qualquer solicitante, incluindo o usuário raiz, resultará em um erro de Acesso negado (403 proibido). Além disso, lembre-se de que quando você envia uma solicitação DELETE para um objeto protegido pelo modo de retenção de conformidade, o Amazon S3 cria um [marcador de exclusão](#) para o objeto.
- Se a versão do objeto estiver protegida com o modo de retenção de governança e você tiver a permissão `s3:BypassGovernanceRetention`, poderá ignorar a proteção e excluir permanentemente a versão. Para obter mais informações, consulte [Ignorar modo de governança](#).
- Se a versão do objeto estiver protegida por uma retenção legal, uma solicitação DELETE permanente poderá resultar em um erro de Acesso negado (403 proibido). Para excluir permanentemente a versão do objeto, é necessário remover a retenção legal da versão do objeto. Para remover uma retenção legal, você deve ter a permissão `s3:PutObjectLegalHold`. Para obter mais informações como remover uma retenção legal, consulte [Configurar a funcionalidade Bloqueio de Objetos do S3](#).

Política de endpoint da VPC

Se você estiver acessando o Amazon S3 usando um endpoint da nuvem privada virtual (VPC), verifique se a política de endpoint da VPC não está impedindo que você acesse os recursos do Amazon S3. Por padrão, a política de endpoint da VPC permite todas as solicitações ao Amazon S3. Também é possível configurar a política de endpoint da VPC para restringir determinadas solicitações. Para obter informações sobre como verificar a política de endpoint da VPC, consulte [Controlar o acesso a endpoints da VPC usando políticas de endpoint](#) no Guia do AWS PrivateLink.

Políticas do AWS Organizations

Se a sua Conta da AWS pertence a uma organização, as políticas do AWS Organizations podem impedir que você acesse os recursos do Amazon S3. Por padrão, as políticas do AWS Organizations não bloqueiam nenhuma solicitação ao Amazon S3. No entanto, verifique se as suas políticas do AWS Organizations não foram configuradas para bloquear o acesso aos buckets do S3. Para obter instruções sobre como verificar as políticas do AWS Organizations, consulte [Listar todas as políticas](#) no Guia do usuário do AWS Organizations.

Configurações do ponto de acesso

Se você receber um erro de acesso negado (403 proibido) ao fazer solicitações por meio de Pontos de Acesso Amazon S3, talvez seja necessário verificar o seguinte:

- As configurações dos pontos de acesso
- A política de usuário do IAM usada para os pontos de acesso
- A política de bucket usada para gerenciar ou configurar os pontos de acesso entre contas

Políticas e configurações de pontos de acesso

- Ao criar um ponto de acesso, é possível designar Internet ou VPC como a origem da rede. Se a origem da rede for definida somente como VPC, o Amazon S3 rejeitará todas as solicitações feitas ao ponto de acesso que não sejam originadas da VPC especificada. Para verificar a origem da rede do ponto de acesso, consulte [Criar pontos de acesso restritos a uma nuvem privada virtual](#).
- Com os pontos de acesso, você também pode definir configurações personalizadas do bloqueio de acesso público, que funcionam de forma semelhante às configurações desse recurso no nível do bucket ou da conta. Para verificar as configurações personalizadas do bloqueio de acesso público, consulte [Gerenciar o acesso público a pontos de acesso](#).
- Para fazer solicitações bem-sucedidas ao Amazon S3 usando pontos de acesso, o solicitante deve ter as permissões necessárias do IAM. Para ter mais informações, consulte [Configurar políticas do IAM para uso de pontos de acesso](#).
- Se a solicitação envolver pontos de acesso entre contas, o proprietário do bucket deve ter atualizado a política do bucket para autorizar solicitações do ponto de acesso. Para ter mais informações, consulte [Conceder permissões para pontos de acesso entre contas](#).

Se o erro Acesso negado (403 proibido) persistir depois de verificar todos os itens neste tópico, [recupere seu ID de solicitação do Amazon S3](#) e entre em contato com o AWS Support para obter mais orientações.

Solução de problemas com operações em lote

Os tópicos a seguir listam os erros comuns para ajudar você a solucionar problemas que pode encontrar durante as operações em lote.

Erros comuns

- [Relatório de trabalhos não é entregue quando há um problema com as permissões ou quando o modo de retenção do Bloqueio de objetos do S3 está habilitado](#)
- [Falha na replicação em lote do S3 com erro: a geração do manifesto não encontrou chaves que corresponderem aos critérios do filtro](#)

- [As falhas de operações em lote ocorrem após a adição de uma nova regra de replicação a uma configuração de replicação existente](#)
- [Objetos com falha nas operações em lote com o erro 400 InvalidRequest: falha na tarefa devido à falta de VersionId](#)
- [Crie uma falha de trabalho com a opção de tag de trabalho ativada](#)
- [Acesso negado para ler o manifesto](#)

Relatório de trabalhos não é entregue quando há um problema com as permissões ou quando o modo de retenção do Bloqueio de objetos do S3 está habilitado

O erro a seguir ocorrerá se não houver as permissões necessárias ou se o modo de retenção do bloqueio de objetos (modo de governança ou modo de conformidade) estiver ativado no bucket de destino.

Erro: motivos da falha. Não foi possível gravar o relatório do trabalho no bucket de relatórios. Verifique suas permissões.

É necessário configurar o perfil do IAM e a política de confiança para permitir que as operações em lote do S3 acessem objetos PUT no bucket em que o relatório será entregue. Se essas permissões necessárias estiverem ausentes, ocorrerá uma falha na entrega do relatório de trabalhos.

Quando um modo de retenção está ativado, o bucket é protegido de gravação única e várias leituras (WORM). O bloqueio de objetos com o modo de retenção habilitado no bucket de destino não é compatível; portanto, as tentativas de entrega do relatório de conclusão de trabalhos falham. Para corrigir esse problema, escolha um bucket de destino para seus relatórios de conclusão de trabalhos que não tenha o modo de retenção de bloqueio de objetos ativado.

Falha na replicação em lote do S3 com erro: a geração do manifesto não encontrou chaves que correspondessem aos critérios do filtro

Os erros a seguir ocorrem quando objetos no bucket de destino são armazenados nas classes de armazenamento S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive.

Erro: a geração do manifesto não encontrou chaves que correspondessem aos critérios do filtro.

Note

Se os critérios de filtro fornecidos não corresponderem a nenhum objeto válido no bucket de origem, esse erro poderá ser um comportamento esperado. Verifique os critérios do filtro e a classe de armazenamento do objeto de destino.

Para usar a replicação em lote nesses objetos, primeiro restaure-os na classe de armazenamento S3 Standard usando uma operação de iniciar a restauração do objeto do S3 em um trabalho de operações em lote. Para obter mais informações, consulte [Restaurar um objeto arquivado](#) e [Restaurar objetos \(operações em lote\)](#). Depois de restaurar os objetos, você pode replicá-los usando um trabalho de replicação em lote.

As falhas de operações em lote ocorrem após a adição de uma nova regra de replicação a uma configuração de replicação existente

As operações em lote tentam realizar a replicação de objetos existente para cada regra na configuração de replicação do bucket de origem. Se houver problemas com qualquer uma das regras de replicação existentes, poderão ocorrer falhas.

O relatório de conclusão do trabalho de operações em lote explica os motivos da falha do trabalho. Para obter uma lista de erros comuns, consulte [Motivos de falha da replicação do Amazon S3](#).

Objetos com falha nas operações em lote com o erro 400 InvalidRequest: falha na tarefa devido à falta de VersionId

O exemplo de erro a seguir ocorre se um trabalho de operações em lote estiver executando ações em objetos em um bucket versionado e encontrar um objeto no manifesto com um campo de ID de versão vazio.

Erro: *BUCKET_NAME, prefix/file_name*,failed,400,InvalidRequest,Falha na tarefa devido à falta de VersionId

Esse erro ocorre porque o campo ID da versão no manifesto é uma string vazia, em vez da string `null` literal.

As operações em lote falharão para aquele objeto ou objetos específicos, mas não para todo o trabalho. Esse problema ocorre se o formato do manifesto estiver configurado para usar IDs de

versão durante a operação. Os trabalhos sem versionamento não enfrentam esse problema porque operam somente na versão mais recente de cada objeto e ignoram os IDs de versão no manifesto.

Para corrigir esse problema, converta os IDs de versão vazios em strings `null`. Para obter mais informações, consulte [the section called “Converter strings de ID de versão vazias em cadeias de caracteres null”](#).

Crie uma falha de trabalho com a opção de tag de trabalho ativada

Sem a permissão `s3:PutJobTagging`, a criação de trabalhos de operações em lote com a opção de tag de trabalho ativada causa erros `403 access denied`.

Para criar trabalhos de operações em lote com a opção de tag de trabalho ativada, o usuário do AWS Identity and Access Management (IAM) que está criando o trabalho de operações em lote deve ter a permissão `s3:PutJobTagging` além da permissão `s3:CreateJob`.

Para obter mais informações sobre as permissões necessárias para as operações em lote, consulte [the section called “Conceder permissões”](#).

Acesso negado para ler o manifesto

Se as operações em lote não conseguirem ler o arquivo de manifesto ao tentar criar um trabalho de operações em lote, os seguintes erros poderão ocorrer.

AWS CLI

Motivo da falha A leitura do manifesto é proibida: `AccessDenied`

Console do Amazon S3

Aviso: não é possível obter a ETag do objeto de manifesto. Especifique um objeto diferente para continuar.

Para resolver esse problema, execute um dos seguintes procedimentos:

- Verifique se o perfil do IAM da Conta da AWS usado para criar o trabalho de operações em lote tem as permissões `s3:GetObject`. O perfil do IAM da conta deve ter as permissões `s3:GetObject` para permitir que as operações em lote leiam o arquivo de manifesto.

Para obter mais informações sobre as permissões necessárias para as operações em lote, consulte [the section called “Conceder permissões”](#).

- Verifique os metadados dos objetos do manifesto para ver se há incompatibilidade de acesso com a propriedade do objeto do S3. Para obter mais informações sobre a propriedade de objeto do S3, consulte [the section called “Controlar a propriedade do objeto”](#).
- Verifique se as chaves do AWS Key Management Service (AWS KMS) são usadas para criptografar o arquivo de manifesto.

As operações em lote são compatíveis com relatórios de inventário CSV criptografados pelo AWS KMS. No entanto, as operações em lote não são compatíveis com arquivos de manifesto CSV criptografados pelo AWS KMS. Para obter mais informações, consulte [Configurar o inventário do Amazon S3](#) e [Especificar um manifesto](#).

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.

Solução de problemas do ciclo de vida do Amazon S3

As informações a seguir podem ajudar você a solucionar problemas comuns com as regras de ciclo de vida do Amazon S3.

Tópicos

- [Executei uma operação de lista no bucket e vi objetos que achei que estavam expirados ou que tinham sido transferidos de acordo com uma regra de ciclo de vida.](#)
- [Como faço para monitorar o andamento da minha regra de ciclo de vida para verificar se ela está ativa?](#)
- [Minha quantidade de objetos do S3 ainda aumenta, mesmo depois de configurar as regras do ciclo de vida em um bucket com versionamento ativado.](#)
- [Como esvaziar meu bucket do S3 usando regras de ciclo de vida?](#)
- [Minha fatura do Amazon S3 aumentou após a transição de objetos para uma classe de armazenamento de menor custo.](#)
- [Eu atualizei minha política de bucket, mas meus objetos do S3 ainda estão sendo excluídos pelas regras de ciclo de vida expiradas.](#)
- [Posso recuperar objetos do S3 que expiraram pelas regras do ciclo de vida do S3?](#)

Executei uma operação de lista no bucket e vi objetos que achei que estavam expirados ou que tinham sido transferidos de acordo com uma regra de ciclo de vida.

As [transições de objetos](#) e as [expirações de objetos](#) do ciclo de vida do S3 são operações assíncronas. Portanto, pode haver um atraso entre o momento em que os objetos são elegíveis para expiração ou transição e o momento em que eles realmente são transferidos ou expirados. As alterações no faturamento são aplicadas assim que a regra do ciclo de vida é cumprida, mesmo que

a ação não seja concluída. A única exceção a esse comportamento é se você tiver uma regra de ciclo de vida definida para transição à 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 mudanças no faturamento, consulte [Definir a configuração do ciclo de vida em um bucket](#).

Note

O Amazon S3 não faz a transição de objetos com menos de 128 KB da classe de armazenamento S3 Standard ou S3 Standard-IA para a classe de armazenamento S3 Intelligent-Tiering, S3 Standard-IA ou S3 One Zone-IA.

Como faço para monitorar o andamento da minha regra de ciclo de vida para verificar se ela está ativa?

Para ver o andamento de (ou monitorar as alterações feitas por) qualquer regra ativa do ciclo de vida, [use o painel da Lente de Armazenamento](#). Com o painel, é possível visualizar as métricas a seguir, que monitoram a quantidade ou tamanho do objeto.

- Bytes da versão atual
- Contagem de objetos da versão atual
- Bytes de versão não atual
- Número de objetos de versões não atuais
- Contagem de objetos do marcador de exclusão
- Bytes de armazenamento do marcador de exclusão
- Bytes de uploads incompletos de várias partes
- Contagem de objetos de multipart upload incompletos

Também é possível usar os seguintes recursos para monitorar as regras de ciclo de vida:

- [Inventário do Amazon S3](#): é possível usar o inventário do S3 para gerar a lista de prefixos ou objetos para o bucket do Amazon S3 (em CSV), no formato colunar de linhas otimizado (ORC) para Apache ou no formato Apache Parquet para fins de auditoria. Dependendo do caso de uso, você também pode consultar o inventário do S3 no SQL padrão com o Amazon Athena.

- [Notificações de eventos do S3](#): é possível configurar notificações de eventos para receber notificações sobre qualquer evento de transição ou expiração do ciclo de vida.
- Logs de acesso ao servidor S3: é possível habilitar logs de acesso ao servidor para o bucket do S3 a fim de capturar ações relacionadas ao ciclo de vida, como transições de objetos para outra classe de armazenamento e expirações de objeto. Para obter mais informações, consulte [Ciclo de vida e log](#).

Minha quantidade de objetos do S3 ainda aumenta, mesmo depois de configurar as regras do ciclo de vida em um bucket com versionamento ativado.

Quando um objeto está configurado para expirar em um [bucket com versionamento ativado](#), o objeto não é completamente excluído do bucket. Em vez disso, um [marcador de exclusão](#) é criado como a versão mais recente do objeto. Os marcadores de exclusão ainda são contados como objetos. Portanto, se uma regra de ciclo de vida for criada para expirar somente nas versões atuais, a contagem de objetos no bucket do S3 realmente aumentará em vez de diminuir.

Por exemplo, digamos que um bucket do S3 tenha o versionamento ativado com 100 objetos, e uma regra de ciclo de vida esteja configurada para expirar as versões atuais do objeto após sete dias. Após o sétimo dia, a quantidade de objetos aumenta para 200 porque 100 marcadores de exclusão são criados além dos 100 objetos originais, que agora são as versões não atuais. Para obter mais informações sobre as ações da regra de configuração do ciclo de vida do S3 para buckets com versionamento ativado, consulte [Definir a configuração do ciclo de vida em um bucket](#).

Para remover objetos permanentemente, adicione uma configuração de ciclo de vida adicional para excluir as versões anteriores dos objetos, os marcadores de exclusão expirados e os uploads incompletos de várias partes. Para obter instruções sobre como criar regras de ciclo de vida, consulte [Definir a configuração do ciclo de vida em um bucket](#).

Note

- O Amazon S3 arredonda a data de transição ou expiração de um objeto para a meia-noite UTC do dia seguinte. Para obter mais informações, consulte [Regras de ciclo de vida: com base na idade de um objeto](#).
- Para objetos S3 protegidos pelo bloqueio de objetos, as versões atuais não são excluídas permanentemente. Em vez disso, um marcador de exclusão é adicionado aos objetos,

tornando-os não atuais. As versões não atuais são então preservadas e não expiram permanentemente.

Como esvaziar meu bucket do S3 usando regras de ciclo de vida?

As regras do ciclo de vida do S3 são uma ferramenta eficaz para [esvaziar um bucket do S3](#) com milhões de objetos. Para excluir um grande número de objetos do bucket do S3, use estes quatro pares de regras de ciclo de vida:

- Expire as versões atuais dos objetos e Exclua permanentemente as versões anteriores dos objetos
- Exclua marcadores de exclusão expirados e Exclua uploads incompletos de várias partes

Para ver as etapas de como criar regras de configuração de ciclo de vida, consulte [Definir a configuração do ciclo de vida em um bucket](#).

Note

Para objetos S3 protegidos pelo bloqueio de objetos, as versões atuais não são excluídas permanentemente. Em vez disso, um marcador de exclusão é adicionado aos objetos, tornando-os não atuais. As versões não atuais são então preservadas e não expiram permanentemente.

Minha fatura do Amazon S3 aumentou após a transição de objetos para uma classe de armazenamento de menor custo.

Há vários motivos pelos quais sua fatura pode aumentar após a transição de objetos para uma classe de armazenamento de menor custo:

- Sobrecarga do S3 Glacier cobra por objetos pequenos

Para cada objeto que é transferido para o S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, uma sobrecarga total de 40 KB é associada a essa atualização de armazenamento. Como parte da sobrecarga de 40 KB, 8 KB são usados para armazenar os metadados e o nome do objeto. Esses 8 KB são cobrados de acordo com as tarifas do S3 Standard. Os 32 KB restantes são usados para indexação e metadados relacionados. Esses 32 KB são cobrados de acordo com os preços do S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive.

Portanto, se você estiver armazenando muitos objetos menores, não recomendamos o uso de transições de ciclo de vida. Em vez disso, para reduzir as despesas adicionais, considere a possibilidade de agregar vários objetos menores em um número menor de objetos grandes antes de armazená-los no Amazon S3. Para obter mais informações sobre as considerações de custo, consulte [Transição para as classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive \(arquivamento de objetos\)](#).

- Cobranças de armazenamento mínimo

Algumas classes de armazenamento do S3 têm requisitos mínimos de duração de armazenamento. Os objetos que são excluídos, substituídos ou transferidos dessas classes antes que a duração mínima seja cumprida recebem uma cobrança proporcional de taxa de exclusão ou transição antecipada. Esses requisitos de duração mínima de armazenamento são os seguintes:

- S3 Standard-IA e S3 One Zone-IA: 30 dias
- S3 Glacier Flexible Retrieval e S3 Glacier Instant Retrieval: 90 dias
- S3 Glacier Deep Archive: 180 dias

Para obter mais informações sobre esses requisitos, consulte a seção Restrições de [Transição de objetos usando o ciclo de vida do S3](#). Para obter informações gerais sobre preços do S3, consulte [Preços do Amazon S3](#) e a [Calculadora de preços da AWS](#).

- Custos de transição do ciclo de vida

Cada vez que um objeto é transferido para uma classe de armazenamento diferente por uma regra de ciclo de vida, o Amazon S3 considera essa transição como uma solicitação de transição. Os custos dessas solicitações de transição são adicionais aos custos dessas classes de armazenamento. Se você pretende fazer a transição de um grande número de objetos, considere os custos de solicitação ao fazer a transição para um nível mais baixo. Para obter mais informações, consulte [Definição de preço do Amazon S3](#).

Eu atualizei minha política de bucket, mas meus objetos do S3 ainda estão sendo excluídos pelas regras de ciclo de vida expiradas.

As declarações Deny em uma política de bucket não evitam a expiração dos objetos definidos em uma regra de ciclo de vida. As ações do ciclo de vida (como transições ou expirações) não usam a operação `DeleteObject` do S3. Em vez disso, as ações do ciclo de vida do S3 são executadas usando endpoints internos do S3. (Para obter mais informações, consulte [Ciclo de vida e log](#).)

Para evitar que a regra de ciclo de vida realize qualquer ação, você deve editar, excluir ou [desativar a regra](#).

Posso recuperar objetos do S3 que expiraram pelas regras do ciclo de vida do S3?

A única maneira de recuperar objetos que expiraram pelo ciclo de vida do S3 é por meio do versionamento, que deve estar em vigor antes que os objetos se tornem elegíveis para expiração. Você não pode desfazer as operações de expiração que são executadas pelas regras do ciclo de vida. Se os objetos forem excluídos permanentemente pelas regras de ciclo de vida do S3 em vigor, você não poderá recuperá-los. Para ativar o versionamento em um bucket, consulte [the section called “Usando o versionamento do S3”](#).

Se você aplicou o versionamento ao bucket e as versões não atuais dos objetos ainda estão intactas, é possível [restaurar as versões anteriores dos objetos expirados](#). Para obter mais informações sobre o comportamento das ações e os estados de versionamento do ciclo de vida do S3, consulte a tabela de estados de Ações de ciclo de vida e estado de versionamento do bucket em [Elementos de configuração do ciclo de vida](#).

Note

Se o bucket do S3 estiver protegido pelo [AWS Backup](#) ou pela [Replicação do S3](#), você também poderá usar esses recursos para recuperar os objetos expirados.

Solução de problemas de replicação

Esta seção lista dicas de solução de problemas para a replicação do Amazon S3 e apresenta informações sobre erros de replicação em lote do S3.

Tópicos

- [Dicas de solução de problemas para a replicação do S3](#)
- [Erros de replicação em lote](#)

Dicas de solução de problemas para a replicação do S3

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. 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. Para ter visibilidade dos tempos de replicação, você pode [usar o Controle de Tempo de Replicação do S3 \(S3 RTC\)](#).

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. Além disso, para receber informações sobre falhas durante a replicação, você pode configurar a replicação de notificações de eventos do Amazon S3 para receber eventos de falha. Para obter mais informações, consulte [Receber eventos de falha de replicação com Notificações de Eventos do Amazon S3](#).

- Você pode chamar a operação da API HeadObject para verificar o status de replicação de um objeto. A operação da API HeadObject retorna o status de replicação PENDING, COMPLETED ou FAILED de um objeto. Em resposta a uma chamada da API HeadObject, o status da replicação é retornado no elemento x-amz-replication-status.

Note

Para executar HeadObject, você deve ter acesso de leitura ao objeto que está solicitando. Uma solicitação HEAD tem as mesmas opções de uma solicitação GET, sem realizar uma operação GET. Por exemplo, para executar uma solicitação HeadObject usando a AWS Command Line Interface (AWS CLI), você pode executar o comando a seguir. Substitua os *user input placeholders* por suas próprias informações.

```
aws s3api head-object --bucket my-bucket --key index.html
```

- Depois que HeadObject retornar os objetos com um status de replicação FAILED, você poderá usar a replicação em lote do S3 para replicar esses objetos com falha. Como alternativa, você pode recarregar os objetos com falha no bucket de origem, o que iniciará a replicação dos novos objetos.
- Na configuração de replicação do bucket de origem, verifique o seguinte:
 - O nome de recurso da Amazon (ARN) do bucket de destino está correto.

- O prefixo do nome de chave está correto. Por exemplo, se você definiu a configuração para replicar objetos com o prefixo Tax, apenas objetos com nomes de chaves como Tax/document1 ou Tax/document2 serão replicados. Um objeto com o nome de chave document3 não será replicado.
- O status da regra de replicação é Enabled.
- Verifique se o versionamento não foi suspenso em nenhum bucket na configuração da replicação. Tanto o bucket de origem quanto o de destino devem ter o versionamento ativado.
- Se uma regra de replicação for definida como Alterar a propriedade do objeto para o proprietário do bucket de destino, a o perfil do AWS Identity and Access Management (IAM) usado para replicação deverá ter a permissão `s3:ObjectOwnerOverrideToBucketOwner`. Essa permissão é concedida ao recurso (nesse caso, o bucket de destino). Por exemplo, a declaração Resource a seguir mostra como conceder essa permissão no bucket de destino:

```
{
  "Effect": "Allow",
  "Action": [
    "s3:ObjectOwnerOverrideToBucketOwner"
  ],
  "Resource": "arn:aws:s3:::DestinationBucket/*"
}
```

- Se o bucket de destino for de propriedade de outra conta, o proprietário do bucket de destino também deve conceder a permissão `s3:ObjectOwnerOverrideToBucketOwner` ao bucket de destino por meio da política de bucket de destino. Para usar o exemplo de política de bucket a seguir, substitua *user input placeholders* por suas próprias informações:

```
{
  "Version": "2012-10-17",
  "Id": "Policy1644945280205",
  "Statement": [
    {
      "Sid": "Stmt1644945277847",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789101:role/s3-replication-role"
      },
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateTags",

```

```
    "s3:ObjectOwnerOverrideToBucketOwner"  
  ],  
  "Resource": "arn:aws:s3:::DestinationBucket/*"  
}  
]  
}
```

Note

Se as configurações de propriedade do objeto de Imposto pelo proprietário do bucket, você não precisará atualizar a configuração para Alterar a propriedade do objeto para o proprietário do bucket de destino na regra de replicação. A alteração de propriedade do objeto ocorrerá por padrão. Para obter mais informações sobre alterações da propriedade da réplica, consulte [Alterar o proprietário da réplica](#).

- Se você estiver definindo a configuração de replicação em um cenário entre contas, no qual os buckets de origem e de destino pertencem a Contas da AWS diferentes, os buckets de destino não poderão ser configurados como buckets de Pagamento pelo solicitante. Para ter mais informações, consulte [Configuração de buckets de Pagamento pelo solicitante para transferências de armazenamento e uso](#).
- Se os objetos de origem de um bucket forem criptografados com uma chave do AWS Key Management Service (AWS KMS), a regra de replicação deverá ser configurada para incluir objetos criptografados pelo AWS KMS. Certifique-se de selecionar Replicar objetos criptografados com o AWS KMS nas configurações de Criptografia no console do Amazon S3. Depois, selecione uma chave do AWS KMS para criptografar objetos de destino.

Note

Se o bucket de destino estiver em uma conta diferente, especifique uma chave gerenciada pelo cliente do AWS KMS que seja de propriedade da conta de destino. Não use a chave gerenciada padrão do Amazon S3 (aws/s3). O uso da chave padrão criptografa os objetos com a chave gerenciada do Amazon S3 que pertence à conta de origem, impedindo que o objeto seja compartilhado com outra conta. Como resultado, a conta de destino não conseguirá acessar os objetos no bucket de destino.

Para usar uma chave do AWS KMS que pertence à conta de destino para criptografar os objetos de destino, a conta de destino deve conceder as permissões `kms:GenerateDataKey`

e `kms:Encrypt` ao perfil de replicação na política de chave do KMS. Para usar o exemplo de declaração a seguir na sua política de chave do KMS, substitua *user input placeholders* por suas próprias informações:

```
{
  "Sid": "AllowS3ReplicationSourceRoleToUseTheKey",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::123456789101:role/s3-replication-role"
  },
  "Action": ["kms:GenerateDataKey", "kms:Encrypt"],
  "Resource": "*"
}
```

Se você usar um asterisco (*) para a declaração `Resource` na política de chave do AWS KMS, a política concederá permissão para usar a chave do KMS somente para o perfil de replicação. A política não permite que o perfil de replicação eleve suas permissões.

Por padrão, a política de chave do KMS concede ao usuário raiz permissões totais para a chave. Essas permissões podem ser delegadas a outros usuários na mesma conta. A menos que haja declarações `Deny` na política de chave do KMS de origem, usar uma política do IAM para conceder permissões ao perfil de replicação para a chave do KMS de origem é suficiente.

Note

As políticas de chave do KMS que restringem o acesso a intervalos CIDR específicos, endpoints da VPC ou pontos de acesso do S3 podem causar falhas na replicação.

Se as chaves do KMS de origem ou de destino concederem permissões com base no contexto de criptografia, confirme se as chaves de bucket do Amazon S3 estão ativadas para os buckets. Se os buckets tiverem as chaves de bucket do S3 ativadas, o contexto de criptografia deverá ser o recurso no nível de bucket, assim:

```
"kms:EncryptionContext:arn:aws:arn": [
  "arn:aws:s3::SOURCE_BUCKET_NAME"
]
"kms:EncryptionContext:arn:aws:arn": [
  "arn:aws:s3::DESTINATION_BUCKET_NAME"
]
```

]

Além das permissões concedidas pela política de chave do KMS, a conta de origem deve adicionar as seguintes permissões mínimas à política do IAM do perfil de replicação:

```
{
  "Effect": "Allow",
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": [
    "SourceKmsKeyArn"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "kms:GenerateDataKey",
    "kms:Encrypt"
  ],
  "Resource": [
    "DestinationKmsKeyArn"
  ]
}
```

Para obter mais informações sobre como replicar objetos criptografados com o AWS KMS, consulte [Replicar objetos criptografados](#).

- 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](#).
- Se os objetos ainda não estiverem se replicando depois de você validar as permissões, verifique se há declarações Deny explícitas nos seguintes locais:
 - Declarações Deny nas políticas do bucket de origem ou destino. 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"
```

- Declarações Deny ou limites de permissões associados ao perfil do IAM podem fazer com que a replicação falhe.
- Declarações Deny nas políticas de controle de serviço do AWS Organizations vinculadas às contas de origem ou de destino podem causar falha na replicação.
- Se a réplica do objeto não aparecer no bucket de destino, os seguintes problemas podem ter impedido a replicação:
 - 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 uma configuração de replicação do bucket A para o bucket B e para o bucket C, o Amazon S3 não replicará réplicas de objetos 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 visualizar um exemplo de política, consulte [Conceder permissões entre contas para fazer upload de objetos garantindo que o proprietário do bucket tenha controle total](#).
- Vamos supor que, na configuração da replicação, você adicione uma regra para replicar um subconjunto de objetos com uma tag específica. Nesse 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.

- Use Notificações de Eventos do Amazon S3 para notificar você sobre instâncias quando os objetos não são replicados para a Região da AWS 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 ter mais informações, consulte [Receber eventos de falha de replicação com Notificações de Eventos do Amazon S3](#).

Você também pode ver motivos da falha de replicação usando as notificações de eventos do Amazon S3. Para ver a lista de motivos da falha, consulte [Motivos de falha de replicação do Amazon S3](#).

Erros de replicação em lote

Para solucionar problemas de objetos que não estão se replicando no bucket de destino, verifique os diferentes tipos de permissões para o bucket, função de replicação e o perfil do IAM que são usados para criar o trabalho de replicação em lote. Além disso, verifique as configurações de acesso público e as configurações de propriedade do bucket.

Ao usar a replicação em lote, você pode encontrar um destes erros:

- O status da operação Batch falhou devido ao motivo: não foi possível gravar o relatório do trabalho em seu bucket de relatórios.

Esse erro ocorrerá se o perfil do IAM usado para o trabalho de operações em lote não conseguir colocar o relatório de conclusão no local especificado quando você criou o trabalho. Para resolver esse erro, verifique se o perfil do IAM tem permissões PutObject para o bucket em que você deseja salvar o relatório de conclusão das operações em lote. É uma prática recomendada entregar o relatório em um bucket diferente do bucket de origem.

- A operação em lote é concluída com falhas e o total de falhas não é 0.

Esse erro ocorre se houver problemas de permissões insuficientes de objetos com o trabalho de replicação em lote que está sendo executado. Se você estiver usando uma regra de replicação para o trabalho de replicação em lote, verifique se o perfil do IAM usado para replicação tem as permissões adequadas para acessar objetos do bucket de origem ou de destino. Você também pode verificar o [Relatório de conclusão da replicação em lote](#) para analisar o [Motivo de falha da replicação do Amazon S3](#) específico.

- O trabalho em lote foi executado com êxito, mas o número de objetos esperados no bucket de destino não é o mesmo.

Esse erro ocorre quando há uma incompatibilidade entre os objetos listados no manifesto fornecido no trabalho de replicação em lote e os filtros que você selecionou ao criar o trabalho. Você também pode receber essa mensagem quando os objetos no bucket de origem não corresponderem a nenhuma regra de replicação e não estiverem incluídos no manifesto gerado.

Solução de problemas do registro em log de acessos ao servidor

Os tópicos a seguir podem ajudar a solucionar problemas que você pode encontrar ao configurar o registro em log com o Amazon S3.

Tópicos

- [Mensagens de erro comuns ao configurar o registro em log](#)
- [Solução de problemas de falha de entrega](#)

Mensagens de erro comuns ao configurar o registro em log

As seguintes mensagens de erro comuns podem aparecer quando você ativa o registro em log por meio da AWS Command Line Interface (AWS CLI) e dos AWS SDKs:

Erro: registro em log de localização cruzada do S3 não é permitido

Se o bucket de destino (também conhecido como bucket-alvo) estiver em uma região diferente do bucket de origem, ocorrerá um erro Não é permitido o registro em log entre locais do S3. Para resolver esse erro, verifique se o bucket de destino configurado para receber os logs de acesso está na mesma Região da AWS e Conta da AWS que o bucket de origem.

Erro: o proprietário do bucket a ser registrado e o bucket de destino devem ser os mesmos

Quando você habilitar o registro em log de acesso ao servidor, esse erro ocorrerá se o bucket de destino especificado pertencer a uma conta diferente. Para resolver esse erro, verifique se o bucket de destino está na mesma Conta da AWS que o bucket de origem.

Note

Recomendamos que você escolha um bucket de destino diferente do bucket de origem. Quando o bucket de origem e o bucket de destino são os mesmos, logs adicionais são

criados para os logs que são gravados no bucket, o que pode aumentar a fatura de armazenamento. Esses logs extras sobre logs também podem dificultar a localização dos logs específicos que você está procurando. Para um gerenciamento de logs mais simples, recomendamos salvar logs de acesso em um bucket diferente. Para obter mais informações, consulte [the section called “Como faço para habilitar a entrega de logs?”](#).

Erro: o bucket de destino para registro em log não existe

O bucket de destino precisa existir antes de definir a configuração. Esse erro indica que o bucket de destino não existe ou não pôde ser encontrado. Verifique se o nome do bucket está escrito corretamente e tente novamente.

Erro: concessões de destino não são permitidas para buckets com proprietário de bucket aplicado

Esse erro indica que o bucket de destino usa a configuração Imposto pelo proprietário do bucket para a funcionalidade Propriedade de Objetos do S3. A configuração Imposto pelo proprietário do bucket não é compatível com concessões de destino (alvo). Para obter mais informações, consulte [Permissões para entrega de logs](#).

Solução de problemas de falha de entrega

Para evitar problemas de registro em log de acesso ao servidor, siga estas práticas recomendadas:

- O grupo de entrega de logs do S3 tem acesso de gravação ao bucket de destino: o grupo de entrega de logs do S3 entrega logs de acesso ao servidor para o bucket de destino. Uma política de bucket ou uma lista de controle de acesso (ACL) do bucket para conceder acesso de gravação ao bucket de destino. No entanto, recomendamos usar uma política de bucket em vez de uma ACL. Para obter mais informações sobre como conceder acesso de gravação ao bucket de destino, consulte [Permissões para entrega de logs](#).

Note

Se o bucket de destino usar a configuração Imposto pelo proprietário do bucket para a funcionalidade Propriedade de Objetos, esteja ciente do seguinte:

- As ACLs estão desativadas e não afetam mais as permissões. Isso significa que não é possível atualizar a ACL do bucket para conceder acesso ao grupo de entrega de log do S3. Em vez disso, para conceder acesso à entidade principal de serviço do registro em log, é necessário atualizar a política de bucket para o bucket de destino.

- Não é possível incluir concessões de destino na configuração de PutBucketLogging.

- A política de bucket para o bucket de destino permite o acesso aos logs: verifique a política do bucket de destino. Pesquise na política de bucket declarações que contenham "Effect": "Deny". Depois, verifique se a declaração Deny não está impedindo que os logs de acesso sejam gravados no bucket.
- A funcionalidade Bloqueio de Objetos do S3 não está habilitada no bucket de destino: verifique se o bucket de destino está com a funcionalidade Bloqueio de Objetos habilitada. O bloqueio de objetos bloqueia a entrega do log de acesso ao servidor. Você deve escolher um bucket de destino que não esteja com funcionalidade Bloqueio de Objetos habilitada.
- As chaves gerenciadas pelo Amazon S3 (SSE-S3) são selecionadas se a criptografia padrão está habilitada no bucket de destino: é possível usar a criptografia de bucket padrão no bucket de destino somente se você usa a criptografia do lado do servidor com as chaves gerenciadas pelo Amazon S3 (SSE-S3). A criptografia padrão do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-KMS) não é compatível com os buckets de destino do registro em log de acesso ao servidor. Para obter mais informações sobre como ativar a criptografia padrão, consulte [Configurar a criptografia padrão](#).
- O bucket de destino não tem a opção Pagamento pelo solicitante habilitada: o uso de um bucket de Pagamento pelo solicitante como bucket de destino para o registro em log de acesso ao servidor não é compatível. Para permitir a entrega de logs de acesso ao servidor, desabilite a opção Pagamento pelo solicitante no bucket de destino.
- Revise a política de controle de serviço do AWS Organizations: quando estiver usando o AWS Organizations, verifique as políticas de controle de serviço para garantir que o acesso ao Amazon S3 seja permitido. As políticas de controle de serviço especificam o número máximo de permissões para as contas afetadas. Pesquise na política de controle de serviço as declarações que contenham "Effect": "Deny" e verifique se as declarações Deny não estão impedindo que nenhum log de acesso seja gravado no bucket. Para obter mais informações, consulte [Políticas de controle de serviços \(SCPs\)](#) no Guia do usuário do AWS Organizations.
- Aguarde um tempo para que as alterações recentes na configuração do registro em log entrem em vigor: as ações de habilitar o registro em log de acesso ao servidor pela primeira vez ou alterar o bucket de destino para os logs levam tempo para entrar em vigor totalmente. Pode levar mais de uma hora para que todas as solicitações sejam registradas e entregues adequadamente.

Para verificar se há falha de entrega de log, ative as métricas de solicitação do Amazon CloudWatch. Se os logs não forem entregues em algumas horas, procure a métrica `4xxErrors`, que pode indicar falhas na entrega dos logs. Para obter mais informações sobre como ativar as

métricas de solicitação, consulte [the section called “Criar uma configuração de métricas para todos os objetos”](#).

Solucionar problemas de versionamento

Os tópicos a seguir podem ajudar a solucionar alguns problemas comuns de versionamento do Amazon S3.

Tópicos

- [Quero recuperar objetos que foram excluídos acidentalmente em um bucket com versionamento ativado](#)
- [Quero excluir permanentemente objetos com versionamento](#)
- [Ocorreu uma degradação do desempenho depois de ativar o versionamento do bucket](#)

Quero recuperar objetos que foram excluídos acidentalmente em um bucket com versionamento ativado

Em geral, quando as versões do objeto são excluídas dos buckets do S3, não há como o Amazon S3 recuperá-las. No entanto, se você habilitou o versionamento do S3 no bucket do S3, uma solicitação DELETE que não especifica um ID de versão não pode excluir permanentemente um objeto. Em vez disso, um marcador de exclusão é adicionado como espaço reservado. Esse marcador de exclusão torna-se a versão atual de objeto.

Para verificar se os objetos excluídos foram excluídos permanentemente ou temporariamente (com um marcador de exclusão em seu lugar), 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 nome do bucket que contém o objeto.
4. Na lista Objetos, ative a opção Mostrar versões à direita da barra de pesquisa e, depois, pesquise o objeto excluído na barra de pesquisa. Essa alternância estará disponível somente se o versionamento tiver sido habilitado anteriormente no bucket.

Você também pode usar o [Inventário do S3 para pesquisar objetos excluídos](#).

5. Se você não conseguir encontrar o objeto depois de ativar Mostrar versões ou criar um relatório de inventário, e também não conseguir encontrar um [marcador de exclusão](#) do objeto, a exclusão será permanente e o objeto não poderá ser recuperado.

Você também pode verificar o status de um objeto excluído usando a operação de API `HeadObject` do AWS Command Line Interface (AWS CLI). Para fazer isso, use o comando `head-object` a seguir e substitua *user input placeholders* por suas próprias informações:

```
aws s3api head-object --bucket DOC-EXAMPLE-BUCKET --key index.html
```

Se você executar o comando `head-object` em um objeto com versionamento, cuja versão atual é um marcador de exclusão, você receberá um erro 404 Não encontrado. Por exemplo:

Ocorreu um erro (404) ao chamar a operação `HeadObject`: não encontrado

Se você executar o comando `head-object` em um objeto com versionamento e fornecer o ID da versão do objeto, o Amazon S3 recuperará os metadados do objeto, confirmando que ele ainda existe e não foi excluído permanentemente.

```
aws s3api head-object --bucket DOC-EXAMPLE-BUCKET --key index.html --  
version-id versionID
```

```
{  
  "AcceptRanges": "bytes",  
  "ContentType": "text/html",  
  "LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",  
  "ContentLength": 77,  
  "VersionId": "Zg5HyL7m.eZU9iM7AV1JkrqAiE.0UG4q",  
  "ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",  
  "Metadata": {}  
}
```

Se o objeto for encontrado e a versão mais recente for um marcador de exclusão, a versão anterior do objeto ainda existirá. Como o marcador de exclusão é a versão atual do objeto, você pode recuperá-lo excluindo o marcador de exclusão.

Depois de remover permanentemente o marcador de exclusão, a segunda versão mais recente do objeto se torna a versão atual dele, tornando-o disponível novamente. Para obter uma descrição visual de como os objetos são recuperados, consulte [Remover marcadores de exclusão](#).

Para remover uma versão específica de um objeto, você deve ser o proprietário do bucket. Para excluir permanentemente um marcador de exclusão, inclua seu respectivo ID de versão em uma solicitação `DeleteObject`. Para excluir o marcador de exclusão, use o comando a seguir e substitua *user input placeholders* por suas próprias informações:

```
aws s3api delete-object --bucket DOC-EXAMPLE-BUCKET --key index.html --  
version-id versionID
```

Para obter mais informações sobre o comando `delete-object`, consulte [delete-object](#) na Referência de comandos da AWS CLI. Para obter mais informações sobre como excluir permanentemente os marcadores de exclusão, consulte [Gerenciamento de marcadores de exclusão](#).

Quero excluir permanentemente objetos com versionamento

Em um bucket com versionamento, uma solicitação `DELETE` sem um ID de versão não pode excluir permanentemente um objeto. Em vez disso, essa solicitação insere um marcador de exclusão.

Para excluir permanentemente objetos com versionamento, você pode escolher entre os seguintes métodos:

- Crie uma regra de ciclo de vida do S3 para excluir permanentemente as versões não atuais. Para excluir permanentemente versões não atuais de objetos, selecione Excluir permanentemente versões não atuais de objetos, e insira um número em Dias após os objetos ficarem desatualizados. 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). Para obter mais informações sobre como criar essa regra, consulte [Definir uma configuração de ciclo de vida do S3](#).
- Exclua uma versão especificada incluindo o ID da versão na solicitação `DELETE`. Para obter mais informações, consulte [Como excluir permanentemente objetos com versionamento](#).
- Crie uma regra de ciclo de vida para expirar as versões atuais. Para expirar as versões atuais dos objetos, selecione Expirar versões atuais de objetos e adicione um número em Dias após a criação do objeto. Para obter mais informações sobre como criar essa regra de ciclo de vida, consulte [Definir uma configuração de ciclo de vida do S3](#).
- Para excluir permanentemente todos os objetos com versionamento e excluir marcadores, crie duas regras de ciclo de vida: uma para expirar as versões atuais e excluir permanentemente as versões não atuais dos objetos, e outra para excluir marcadores de exclusão de objetos expirados.

Em um bucket com versionamento ativado, uma solicitação DELETE que não especifica um ID de versão pode remover somente objetos com um ID de versão NULL. Se o objeto foi carregado quando o versionamento estava ativado, uma solicitação DELETE que não especifica um ID de versão cria um marcador de exclusão desse objeto.

Note

Para buckets com o bloqueio de objetos do S3 ativado, uma solicitação de DELETE objeto com um ID de versão de objeto protegido causa um erro 403 Acesso negado. Uma solicitação de DELETE objeto sem um ID de versão adiciona um marcador de exclusão como a versão mais recente do objeto com uma resposta 200 OK. Objetos protegidos pelo bloqueio de objetos não podem ser excluídos permanentemente até que seus períodos de retenção e retenções legais sejam removidos. Para ter mais informações, consulte [the section called “Como o bloqueio de objetos do S3 funciona”](#).

Ocorreu uma degradação do desempenho depois de ativar o versionamento do bucket

A degradação do desempenho poderá ocorrer em buckets com versionamento ativado se houver muitos marcadores de exclusão ou objetos com versionamento e se as práticas recomendadas não forem seguidas.

Marcadores de exclusão em excesso

Depois que você habilita o versionamento em um bucket, uma solicitação DELETE sem um ID de versão feita a um objeto cria um marcador de exclusão com um ID de versão exclusivo. As configurações do ciclo de vida com a regra Expirar versões atuais de objetos adicionam um marcador de exclusão com um ID de versão exclusivo a cada objeto. Marcadores de exclusão excessivos podem reduzir o desempenho no bucket.

Quando o versionamento é suspenso em um bucket, o Amazon S3 marca o ID da versão como NULL em objetos recém-criados. Uma ação de expiração em um bucket com versionamento suspenso faz com que o Amazon S3 crie um marcador de exclusão com NULL como o ID da versão. Em um bucket com versionamento suspenso, um marcador de exclusão NULL é criado para qualquer solicitação de exclusão. Esses marcadores de exclusão NULL também são chamados de marcadores de exclusão de objeto expirado quando todas as versões de objeto são excluídas e resta

apenas um único marcador de exclusão. Se muitos marcadores de exclusão NULL se acumularem, ocorrerá uma degradação do desempenho no bucket.

Objetos com versionamento em excesso

Se um bucket com versionamento ativado contiver objetos com milhões de versões, poderá ocorrer um aumento nos erros 503 Serviço indisponível. Se você perceber um aumento significativo do número de respostas HTTP 503 Serviço indisponível recebidas para solicitações PUT ou DELETE de objetos a um bucket com o versionamento ativado, talvez tenha um ou mais objetos no bucket com milhões de versões. Quando você tem objetos com milhões de versões, o Amazon S3 limita automaticamente as solicitações ao bucket. As solicitações de controle de utilização protegem o bucket de uma quantidade excessiva de tráfego de solicitação, o que pode potencialmente impedir outras solicitações feitas ao mesmo bucket.

Para determinar quais objetos têm milhões de versões, use o inventário do S3. O inventário do S3 gera um relatório que fornece uma lista de arquivos simples dos objetos em um bucket. Para ter mais informações, consulte [Inventário do Amazon S3](#).

Para verificar se há um grande número de objetos com versionamento no bucket, use as métricas da Lente de Armazenamento do S3 para visualizar a Contagem de objetos da versão atual, o Número de objetos de versões não atuais e a Contagem de objetos do marcador de exclusão. Para obter mais informações sobre métricas da Lente de Armazenamento, consulte [Glossário de métricas de lente de armazenamento do Amazon S3](#).

A equipe do Amazon S3 incentiva os clientes a investigarem aplicações que sobrescrevem repetidamente o mesmo objeto, criando potencialmente milhões de versões desse objeto, para determinar se a aplicação está funcionando conforme o esperado. Por exemplo, uma aplicação que sobrescreve o mesmo objeto a cada minuto durante uma semana pode criar mais de dez mil versões. Recomendamos armazenar menos de cem mil versões para cada objeto. Se você tiver um caso de uso que exija milhões de versões para um ou mais objetos, entre em contato com a equipe do AWS Support a fim de obter ajuda para determinar uma solução melhor.

Práticas recomendadas

Para evitar problemas de degradação do desempenho relacionada ao versionamento, recomendamos aplicar as seguintes práticas recomendadas:

- Habilite uma regra de ciclo de vida para expirar as versões anteriores dos objetos. Por exemplo, você pode criar uma regra de ciclo de vida para expirar versões não atuais após 30 dias de

inatividade do objeto. Você também poderá manter várias versões não atuais se não quiser excluir todas elas. Para obter mais informações, consulte [Definir uma configuração do ciclo de vida do S3](#).

- Habilite uma regra de ciclo de vida para excluir marcadores de exclusão de objetos expirados que não tenham objetos de dados associados no bucket. Para obter mais informações, consulte [Remover marcadores de exclusão de objeto expirado](#).

Para ver mais práticas recomendadas de otimização de desempenho do Amazon S3, consulte [Padrões de design de melhores práticas](#).

Obter IDs de solicitação do Amazon S3 para o AWS Support

Sempre que você entrar em contato com o AWS Support devido a erros ou comportamento inesperado no Amazon S3, é necessário fornecer os IDs das solicitações associadas à ação com falha. O AWS Support usa esses IDs das solicitações para ajudar 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 ou eventos de dados do AWS 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 [Entrar em contato com a AWS](#) ou a [Documentação do AWS Support](#).

Usar 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 ferramenta para escutar na porta em que o tráfego do Amazon S3 passa ao enviar outra solicitação HTTP do Amazon S3.

Para solicitações HTTP, o par de IDs de solicitação será parecido com isto:

```
x-amz-request-id: 79104EXAMPLEB723
x-amz-id-2: I0WQ4fDEXAMPLEQM+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 tem ferramentas de desenvolvedor que podem ser usadas para visualizar 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, use as ferramentas de desenvolvedor do navegador 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 no [AWS re:Post](#).

Usar os 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 o log detalhado em cada solicitação e resposta, não recomendamos habilitar o log em sistemas de produção, já que solicitações ou respostas grandes podem provocar uma lentidão significativa em uma aplicação.

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 Go para obter IDs de solicitação

É possível configurar o registro em log usando o SDK para Go. Consulte mais informações em [Response metadata](#) no Guia do desenvolvedor do SDK para Go V2.

Usar o SDK para PHP para obter IDs de solicitação

É possível configurar o log usando PHP. Para obter mais informações, consulte [Como posso ver quais dados são enviados pela rede?](#) no Guia do desenvolvedor do 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 obter e retornar 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. Para obter a solicitação ou a resposta registrada em log, chame `getCachedResponseMetadata(AmazonWebServiceRequest request).getRequestID()`.

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 Guia do desenvolvedor do AWS SDK for Java.

Usar o AWS SDK for .NET para obter IDs de solicitação

É possível configurar o log com o AWS SDK for .NET usando a ferramenta integrada de log `System.Diagnostics`. Para obter mais informações, consulte a publicação [Registro em log com o AWS SDK para .NET](#) no Blog do desenvolvedor da AWS.

Note

Por padrão, o log retornado contém somente informações de erros. Para obter os IDs de solicitação, o arquivo de configuração deve ter `AWSLogMetrics` (e, opcionalmente, `AWSResponseLogging`) adicionado.

Usar o SDK para Python (Boto3) para obter IDs de solicitação

Com o AWS SDK for Python (Boto3), você pode registrar respostas específicas. É possível usar esse recurso para capturar somente os cabeçalhos relevantes. O código a seguir mostra como registrar em log partes da resposta a 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 mais informações, consulte [Discernir informações úteis de respostas de erro](#) na Referência de API do AWS SDK para Python (Boto).

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 de API do AWS SDK para Python (Boto).

Usar o SDK para Ruby para obter IDs de solicitação

É possível obter os IDs de solicitação usando o SDK para Ruby versões 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)
```

Para obter dicas de como obter informações de rede de um cliente da AWS, consulte [Dica de depuração: obter informações de rastreamento de rede de um cliente](#).

Usar a AWS CLI para obter IDs de solicitação

Para obter os IDs de solicitação ao usar a AWS Command Line Interface (AWS CLI), adicione `--debug` ao comando .

Usar o Windows PowerShell para obter IDs de solicitação

Para obter informações sobre como recuperar logs com o Windows PowerShell, consulte a publicação [Log de respostas no AWS Tools for Windows PowerShell](#) no blog de desenvolvimento do .NET.

Usar eventos de dados do AWS CloudTrail para obter IDs de solicitação

Um bucket do Amazon S3 que é configurado com eventos de dados do CloudTrail para registrar operações de API no nível de objeto do S3 fornece informações detalhadas sobre as ações realizadas por um usuário, um perfil ou um serviço da AWS no Amazon S3. É possível [identificar os IDs de solicitação do S3 consultando os eventos do CloudTrail com o Athena](#).

Usar o registro de acesso ao servidor do S3 para obter IDs de solicitação

Um bucket do Amazon S3 configurado para o registro de acesso ao servidor do S3 fornece registros detalhados para cada solicitação feita ao bucket. É possível identificar os IDs de solicitação do S3 [consultando os logs de acesso ao servidor usando o Athena](#).

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.

Alteração	Descrição	Data
O Inventário Amazon S3 é compatível com a chave de condição s3:InventoryAccessibleOptionalFields	O Inventário Amazon S3 é compatível com a chave de condição s3:InventoryAccessibleOptionalFields para controlar se os usuários podem incluir campos de metadados opcionais em seus relatórios. Consulte mais informações em Control S3 Inventory report configuration creation .	20 de fevereiro de 2024
Suporte a IPv6 para o S3 no Outposts	Agora é possível acessar buckets do S3 no Outposts usando IPv6 via endpoints de pilha dupla do S3 no Outposts. O suporte a IPv6 para S3 no Outposts permite que você gerencie buckets do S3 no Outposts e controle os recursos do ambiente de gerenciamento em redes IPv6.	16 de janeiro de 2024
Nova classe de armazenamento de zona única e alta	A classe Amazon S3 Express One Zone é uma classe de armazenamento de zona	28 de novembro de 2023

[performance do Amazon S3: S3 Express One Zone](#)

única e alto desempenho do Amazon S3 desenvolvida com o propósito específico de fornecer acesso consistente aos dados e com latência inferior a dez milissegundos para as aplicações mais sensíveis à latência. Para obter mais informações, consulte [S3 Express One Zone](#).

[O Mountpoint para Amazon S3 adiciona suporte à classe S3 Express One Zone](#)

Agora você pode montar buckets de diretório da classe S3 Express One Zone com o [Mountpoint](#).

28 de novembro de 2023

[Versão do esquema de invocação do Lambda](#)

A funcionalidade Operações em Lote do Amazon S3 apresenta uma nova versão do esquema de invocação do Lambda para uso com trabalhos da funcionalidade Operações em Lote que atuam em buckets de diretório. Para obter mais informações, consulte [Usar o Lambda e a funcionalidade Operações em Lote do Amazon S3 com buckets de diretório](#).

28 de novembro de 2023

[Ação de importação para buckets de diretório](#)

Lançamento da ação de importação no Amazon S3. A importação é um método simplificado para criar trabalhos da funcionalidade Operações em Lote do Amazon S3 a fim de copiar objetos de buckets de uso geral para buckets de diretório. Para obter mais informações, consulte [Importing objects into a directory bucket](#).

28 de novembro de 2023

[Gerenciar o acesso ao S3 com a funcionalidade Concessões de Acesso do S3](#)

A funcionalidade Concessões de Acesso do Amazon S3 permite que você gerencie permissões de dados em grande escala para entidades principais do AWS Identity and Access Management (IAM), além de identidades de diretórios corporativos, como Azure AD. Agora você pode aplicar permissões do S3 de privilégio mínimo e escalar facilmente essas permissões com base nas necessidades da sua empresa. Para obter mais informações, consulte [Gerenciar o acesso com a funcionalidade Concessões de Acesso do S3](#).

26 de novembro de 2023

[O Mountpoint para Amazon S3 adiciona o recurso de armazenamento em cache](#)

Com o [Mountpoint](#), agora é possível configurar o armazenamento em cache para dados acessados repetidamente.

22 de novembro de 2023

[Melhoria da geração de manifestos da funcionalidade Operações em Lote do Amazon S3](#)

Agora você pode instruir a funcionalidade Operações em Lote do Amazon S3 a gerar um manifesto automaticamente com base nos critérios de filtro de objetos especificados ao criar um trabalho. Essa opção está disponível para trabalhos de replicação em lote criados no console do Amazon S3 ou para qualquer tipo de trabalho criado usando a AWS CLI, SDKs da AWS ou a API REST do Amazon S3. Para obter mais informações, consulte [Criar um trabalho de operações em lote do S3](#).

22 de novembro de 2023

[Buckets do Amazon S3 existentes agora podem adicionar configurações da funcionalidade Bloqueio de Objetos](#)

Agora é possível habilitar a funcionalidade Bloqueio de Objetos em um bucket do Amazon S3 existente. Você pode definir retenções jurídicas e períodos de retenção para buckets novos ou existentes. Para obter mais informações, consulte [Usar o bloqueio de objetos do S3](#).

20 de novembro de 2023

[Métricas de solicitação da Lente de Armazenamento do S3 para prefixos](#)

A Lente de Armazenamento do S3 apresenta métricas de solicitação para prefixos em um bucket do Amazon S3. Para obter mais informações, consulte [Categorias de métricas](#).

17 de novembro de 2023

[Grupos de Lente de Armazenamento do Amazon S3](#)

A Lente de Armazenamento do S3 apresenta grupos da Lente de Armazenamento, um filtro personalizado definido para objetos com base em metadados de objetos. Para obter mais informações, consulte [Trabalhar com grupos de Lente de Armazenamento do Amazon S3](#).

15 de novembro de 2023

[Nova política do IAM](#)

O S3 no Outposts agora conta com o `AWSServiceRoleForS3OnOutposts`, um perfil vinculado a serviços para ajudar a gerenciar recursos de rede para você. Para obter mais informações, consulte [Usar perfis vinculados a serviços para o S3 no Outposts](#).

3 de outubro de 2023

[O Amazon S3 fornece a hora da Last-Modified para marcadores de exclusão.](#)

O Amazon S3 fornece a hora da Last-Modified dos marcadores de exclusão nos cabeçalhos de resposta das operações de API Head e Get do S3. Para obter mais informações, consulte [Trabalhar com marcadores de exclusão](#).

27 de setembro de 2023

[Atualização do Amazon S3 para políticas gerenciadas pela AWS](#)

O Amazon S3 adicionou permissões s3:Describe* a AmazonS3ReadOnlyAccess. Para receber mais informações, consulte [Políticas gerenciadas da AWS para o Amazon S3](#).

11 de agosto de 2023

[Horários de início aprimorados para solicitações de restauração padrão feitas por meio do Operações em Lote do S3](#)

As recuperações padrão para solicitações de restauração feitas por meio do Operações em Lote do S3 agora podem começar em minutos. Para receber mais informações, consulte [Opções de recuperação de arquivo](#).

9 de agosto de 2023

[Foi adicionado o Mountpoint, um cliente com alta taxa de throughput para montar um bucket Amazon S3 como um sistema de arquivos local.](#)

Com o [Mountpoint](#), as aplicações podem acessar objetos armazenados no Amazon S3 por meio de operações de arquivo, oferecendo às aplicações acesso ao armazenamento elástico e ao throughput do Amazon S3 por meio de uma interface de arquivo.

9 de agosto de 2023

[Criptografia de camada dupla do lado do servidor com chaves do AWS Key Management Service \(DSSE-KMS\)](#)

A criptografia de camada dupla do lado do servidor com chaves do AWS Key Management Service (AWS KMS) (DSSE-KMS) aplica duas camadas de criptografia aos objetos quando eles são carregados no Amazon S3. Para ter mais informações, consulte [Utilização da criptografia de camada dupla do lado do servidor com chaves do AWS KMS](#).

13 de junho de 2023

[O Amazon S3 habilita o Bloqueio de Acesso Público do S3 e desabilita as listas de controle de acesso \(ACLs\) do S3 para todos os novos buckets.](#)

Agora o Amazon S3 automaticamente habilita o Bloqueio de Acesso Público do S3 e desabilita as listas de controle de acesso (ACLs) do S3 para todos os novos buckets em todas as Regiões AWS . Para ter mais informações, consulte [Bloquear o acesso público ao armazenamento do Amazon S3](#) e [Controlar a propriedade de objetos e desabilitar ACLs para seu bucket](#).

27 de abril de 2023

[Métrica de falha de operações de replicação do S3](#)

O Amazon S3 adiciona uma nova métrica do Amazon CloudWatch para monitorar falhas de replicação do S3. Para obter mais informações, consulte [Monitoramento do progresso com métricas de replicação](#).

5 de abril de 2023

[DNS privado](#)

O AWS PrivateLink para Amazon S3 agora é compatível com o DNS privado. Para ter mais informações, consulte [DNS privado](#).

14 de março de 2023

[Suporte aos pontos de acesso entre contas no console do Amazon S3](#)

O Amazon S3 agora é compatível com a criação de pontos de acesso entre contas com o console do Amazon S3. Para obter mais informações, consulte [Criar pontos de acesso](#).

14 de março de 2023

[O Amazon S3 no Outposts é compatível com a replicação do S3 no Outposts](#)

Com a replicação do S3 local, você pode replicar automaticamente objetos para um único bucket de destino do Outposts ou para vários buckets de destino. Os buckets de destino podem estar em um AWS Outposts diferente ou no mesmo Outposts que o bucket de origem. Para ter mais informações, consulte [Replicar objetos para o S3 no Outposts](#).

14 de março de 2023

[Alias de pontos de acesso do Amazon S3 Object Lambda](#)

Quando você cria um ponto de acesso do Object Lambda, o Amazon S3 gera automaticamente um alias exclusivo para seu ponto de acesso do Object Lambda. É possível usar esse alias em vez de um nome de bucket do Amazon S3 ou o nome do recurso da Amazon (ARN) do ponto de acesso do Object Lambda em uma solicitação para operações de plano de dados do ponto de acesso. Para ter mais informações, consulte [Como usar um alias em estilo de bucket para seu ponto de acesso do Object Lambda](#).

14 de março de 2023

[Suporte entre contas a pontos de acesso multirregião do Amazon S3](#)

O Amazon S3 agora é compatível com a criação de pontos de acesso multirregionais entre contas com o console do Amazon S3. Para ter mais informações, consulte [Criar pontos de acesso multirregionais](#).

14 de março de 2023

[Pontos de acesso entre contas](#)

O Amazon S3 oferece suporte à criação de pontos de acesso entre contas. Você pode criar um ponto de acesso entre contas usando a AWS Command Line Interface (AWS CLI) ou a operação `CreateAccessPoint` da API REST. Para obter mais informações, consulte [Criar pontos de acesso](#).

30 de novembro de 2022

[Amazon S3 oferece suporte a controles de failover para pontos de acesso multirregionais do Amazon S3](#)

O Amazon S3 adicionou controles de failover para pontos de acesso multirregionais do Amazon S3. Esses controles permitem que você transfira o tráfego de solicitações de acesso a dados do S3 que passam por um ponto de acesso multirregional do Amazon S3 para outra Região da AWS em questão de minutos para testar e criar aplicações altamente disponíveis. Para obter mais informações, consulte [Controles de failover para pontos de acesso multirregionais do Amazon S3](#).

28 de novembro de 2022

[Lente de Armazenamento do Amazon S3 aumenta a visibilidade em toda a organização com 34 novas métricas](#)

A Lente de Armazenamento do S3 adicionou 34 novas métricas para revelar oportunidades de otimização de custos mais profundas, identificar práticas recomendadas de proteção de dados e melhorar a performance das workloads de aplicações. Para obter mais informações, consulte [Métricas da Lente de Armazenamento do S3](#).

17 de novembro de 2022

[Amazon S3 oferece suporte a taxas de solicitações de restauração mais altas para S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive](#)

O Amazon S3 oferece suporte a solicitações de restauração a uma taxa de até 1.000 transações por segundo, por Conta da AWS, para as classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive.

15 de novembro de 2022

[Amazon S3 no Outposts oferece suporte a ações e filtros adicionais para o Ciclo de Vida do S3](#)

O S3 no Outposts oferece suporte a regras adicionais do Ciclo de Vida do S3 para otimizar o gerenciamento da capacidade. Você pode expirar objetos à medida que envelhecem ou quando são substituídos por versões mais recentes. Você pode criar uma regra de ciclo de vida para um bucket inteiro ou um subconjunto de objetos dentro de um bucket ao filtrar por prefixos, etiquetas de objeto ou tamanho de objeto. Para obter mais informações, consulte [Criar e gerenciar uma configuração de ciclo de vida](#).

2 de novembro de 2022

[Suporte à Replicação do S3 para objetos SSE-C](#)

Você pode replicar objetos criados com criptografia do lado do servidor usando chaves de criptografia fornecidas pelo cliente. Para obter mais informações sobre como replicar objetos criptografados, consulte [Replicar objetos criados com criptografia do lado do servidor \(SSE-C, SSE-S3, SSE-KMS\)](#).

24 de outubro de 2022

[Amazon S3 no Outposts oferece suporte a alias de pontos de acesso](#)

Com o S3 on Outposts, você deve usar pontos de acesso para acessar qualquer objeto em um bucket do Outposts. Toda vez que você cria um ponto de acesso para um bucket, o S3 no Outposts gera automaticamente um alias de ponto de acesso. É possível usar esse alias de ponto de acesso em vez de um ARN de ponto de acesso para qualquer operação de plano de dados. Para obter mais informações, consulte [Usar um alias em estilo de bucket para um ponto de acesso de bucket do S3 no Outposts](#).

21 de outubro de 2022

[S3 Object Lambda oferece suporte às operações HeadObject , ListObjects e ListObjectsV2](#)

Você pode usar código personalizado para modificar os dados retornados por solicitações GET, LIST e HEAD padrão do S3 para filtrar linhas, redimensionar imagens dinamicamente, editar dados confidenciais e muito mais. Para obter mais informações, consulte [Transformar objetos com o S3 Object Lambda](#).

04 de outubro de 2022

[Amazon S3 no Outposts oferece suporte ao versionamento do S3](#)

Quando habilitado, o versionamento do S3 salva várias cópias distintas de um objeto no mesmo bucket. O versionamento do S3 pode ser usado para preservar, recuperar e restaurar todas as versões de cada objeto armazenado em buckets do Outposts. O versionamento do S3 ajuda você a se recuperar de ações não intencionais de usuários e de falhas da aplicação. Para obter mais informações, consulte [Gerenciar o versionamento do S3 para um bucket do S3 no Outposts](#).

21 de setembro de 2022

[AWS Backup for Amazon S3](#)

O AWS Backup é um serviço totalmente gerenciado e baseado em políticas que você pode usar para definir uma política de backup central para 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 a replicação em lote do S3 para replicar objetos existentes](#)

Com a replicação em lote do S3, você pode replicar objetos que existiam antes que uma configuração de replicação estivesse em vigor. A replicação de objetos existentes é feita por meio do uso de um trabalho de operações em lote. A replicação em lote do S3 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](#)

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](#)

Você pode aplicar a configuração imposta 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. A configuração imposta pelo proprietário do bucket simplifica o gerenciamento de acesso aos 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](#)

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](#)

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](#)

O AWS Backup é um serviço totalmente gerenciado e baseado em políticas que você pode usar para definir uma política de backup central para 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 for Amazon S3](#)

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](#)

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](#)

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](#)

É possível criar regras de ciclo de vida com base no tamanho do objeto ou especificar quantas versões de objeto desatualizadas deverão ser mantidas. Para obter mais informações, consulte [Exemplos de configuração do Ciclo de Vida do S3](#).

23 de novembro de 2021

[Publicar métricas do Amazon S3 Storage Lens 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. Para obter mais informações, consulte [Monitorar métricas do S3 Storage Lens no CloudWatch](#).

22 de novembro de 2021

[Pontos de acesso multirregionais](#)

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 no Outposts adiciona acesso local direto às aplicações](#)

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 \(CoIP\)](#) 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 S3](#)

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 recurso de operações em lote do S3 são compatíveis com o status de chave de bucket do S3](#)

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](#)

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](#)

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](#)

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](#)

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 GET padrão do S3 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](#)

Com o AWS PrivateLink para Amazon S3, você pode se conectar diretamente ao S3 usando um endpoint de interface em sua nuvem privada virtual (VPC) em vez de se conectar pela Internet. Os endpoints de interface são diretamente acessíveis a partir de aplicações on-premises ou em uma Região da AWS diferente. 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](#)

Os eventos de gerenciamento do S3 no Outposts estão disponíveis por meio de 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](#)

O Amazon S3 oferece uma forte consistência de leitura após gravação para solicitações PUT e DELETE de objetos no bucket do 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, etiquetas de objeto do Amazon S3 e metadados de objeto (por exemplo, objeto HEAD) são fortemente 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](#)

A sincronização de modificação de réplica do Amazon S3 mantém metadados de objeto, como etiquetas, 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](#)

As chaves de bucket do Amazon S3 reduzem o custo da criptografia do lado do servidor do Amazon S3 com 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 [Reduzir o custo do SSE-KMS usando chaves de bucket do S3](#).

1º de dezembro de 2020

[Amazon S3 Storage Lens](#)

18 de novembro de 2020

A Lente de Armazenamento do S3 agrega suas métricas e exibe as informações na seção Account snapshot (Snapshot da conta) na página Buckets do console do Amazon S3. A Lente de Armazenamento do S3 também oferece um painel interativo 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. O painel tem opções de detalhamento para gerar e visualizar insights no nível da organização, conta, Região da AWS, classe de armazenamento, bucket, prefixo ou grupo de Lente de Armazenamento. Você também pode enviar uma exportação diária de métricas em CSV ou no formato Parquet para um bucket do S3. Para obter mais informações, consulte [Avaliar a atividade e o uso do armazenamento com o S3 Storage Lens](#).

[Rastrear solicitações do S3 usando o AWS X-Ray](#)

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](#). Para obter mais informações, consulte [Rastrear solicitações usando o X-Ray](#).

16 de novembro de 2020

[Métricas de replicação do S3](#)

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](#)

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](#)

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 objeto do S3](#)

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](#)

Com o Amazon S3 no Outposts, é possível criar buckets do S3 em seus recursos do AWS Outposts, além de armazenar e recuperar facilmente objetos no ambiente 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, de 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](#)

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](#)

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 objetos do S3 com operações em lote do S3](#).

4 de maio de 2020

[Suporte a operações em lote do S3 para retenção legal de bloqueio de objetos](#)

Agora você pode usar operações em lote com bloqueio de objetos do S3 para adicionar uma 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

[Etiquetas de trabalho para operações em lote do S3](#)

Você pode adicionar tags aos trabalhos de operações em lote do S3 para controlar e rotular esses trabalhos. Para obter mais informações, consulte [Etiquetas para trabalhos de operações em lote do S3](#).

16 de março de 2020

[Pontos de acesso do Amazon S3](#)

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 [Gerenciar o acesso a dados com pontos de acesso do Amazon S3](#).

2 de dezembro de 2019

[Analisador de acesso para Amazon S3](#)

O Access Analyzer 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 \(S3 RTC\)](#)

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 [Replicating objects using S3 Replication Time Control \(S3 RTC\)](#) [“Replicar objetos usando o Controle de Tempo de Replicação do S3 (S3 RTC)”].

20 de novembro de 2019

[Replicação para a mesma região](#)

Você pode usar a replicação para a mesma região (SRR) a fim de copiar objetos entre buckets do Amazon S3 na mesma Região da AWS. Para obter informações sobre replicação entre regiões (CRR) e replicação para a mesma região, consulte [Replicação](#).

18 de setembro de 2019

[Suporte da replicação entre regiões ao bloqueio de objetos do S3](#)

A replicação entre regiões agora oferece suporte ao bloqueio de objetos. Para obter mais informações, consulte [What does Amazon replicate?](#) (“O que o Amazon S3 replica?”).

28 de maio de 2019

[Operações em lote do S3](#)

Usando o recurso de 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 \(Hong Kong\)](#)

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](#)

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](#)

Agora o Amazon S3 oferece uma nova classe de armazenamento de arquivos, a Deep Archive (DEEP_ARCHIVE) do S3 Glacier, 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](#)

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](#)

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 S3](#)

O Amazon S3 agora oferece a funcionalidade de bloqueio de objetos que fornece proteções de gravação única e leitura múltipla (WORM) para objetos do Amazon S3. Para obter mais informações, consulte [Bloquear objetos](#).

26 de novembro de 2018

[Restaurar atualização rápida](#)

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 Eventos](#)

As Notificações de Eventos 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 na classe de armazenamento Flexible Retrieval do S3 Glacier](#)

A operação PUT do Amazon S3 agora oferece suporte à especificação de Flexible Retrieval do S3 Glacier como 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 API REST de PUT diretamente para Flexible Retrieval do S3 Glacier, consulte [Histórico do documento: PUT diretamente](#)

26 de novembro de 2018

<u>Nova classe de armazenamento</u>	<u>em Flexible Retrieval do S3 Glacier.</u> O Amazon S3 agora oferece uma nova classe de armazenamento chamada S3 Intelligent-Tiering (INTELLIGENT_TIERING) projetada para dados duradouros com padrões de acesso alternados ou desconhecidos. Para obter mais informações, consulte <u>Classes de armazenamento</u>	26 de novembro de 2018
<u>Bloqueio de acesso público do Amazon S3</u>	O Amazon S3 agora inclui a possibilidade de bloquear acesso público a buckets e objetos por bucket ou conta. Para obter mais informações, consulte <u>Usar o Amazon S3 Block Public Access.</u>	15 de novembro de 2018

[Filtrar melhorias nas regras de replicação entre regiões \(CRR\)](#)

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](#)

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 (SelectScannedBytes e SelectReturnedBytes).

5 de setembro de 2018

[Atualizações agora disponíveis em RSS](#)

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	<p>Exemplos de código atualizados:</p> <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 AWS SDK for PHP 3.0. Para obter mais informações sobre o AWS SDK for PHP 3.0, consulte AWS SDK for PHP. • Ruby: atualização do código de exemplo para que os exemplos funcionem com o AWS SDK for Ruby versão 3. 	30 de abril de 2018
O Amazon S3 agora relata classes de armazenamento S3 Glacier Flexible Retrieval e ONEZONE_IA para métricas de	<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"> • 	30 de abril de 2018

Alteração	Descrição	Data
armazenamento do Amazon CloudWatch Logs	<p>Para objetos das classes de armazenamento ONEZONE_IA e STANDARD_IA, o Amazon S3 relata objetos com menos de 128 KB como tendo 128 KB. Para ter mais informações, consulte Uso de classes de armazenamento do Amazon S3.</p> <ul style="list-style-type: none"> • 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 os preços da classe de armazenamento STANDARD <p>Para ter mais informações, consulte Transição de objetos usando o Amazon S3 Lifecycle.</p> <p>Para obter mais informações sobre métricas de armazenamento, consulte Monitoramento de métricas com o Amazon CloudWatch.</p>	
Nova classe de armazenamento	<p>O Amazon S3 agora oferece uma nova classe de armazenamento, STANDARD_IA (IA significa “infrequent access”, inglês para “acesso com pouca frequência”), para armazenar objetos. Essa classe de armazenamento é otimizada para dados armazenados por longo tempo e acessados com menos frequência. Para ter mais informações, consulte Uso de classes de armazenamento do Amazon S3.</p>	4 de abril de 2018

Alteração	Descrição	Data
Amazon S3 Select	O Amazon S3 agora oferece suporte à recuperação de conteúdo de objetos com base em uma expressão SQL. Para ter mais informações, consulte Filtragem e recuperação de dados usando o Amazon S3 Select .	4 de abril de 2018
Região Ásia-Pacífico (Osaka-Local)	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. <div data-bbox="479 667 1318 1033" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <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> </div>	12 de fevereiro de 2018
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.	29 de novembro de 2017

Alteração	Descrição	Data
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 ter mais informações, consulte Inventário do Amazon S3 .	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 pela AWS (SSE-KMS). Para ter mais informações, consulte Definir o comportamento padrão da criptografia para os buckets do Amazon S3 .	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 ter mais informações, consulte Inventário do Amazon S3 .	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 de CRR a fim de alterar a propriedade da réplica para a Conta da AWS que tem o bucket de destino. Para ter mais informações, consulte Alterar o proprietário da réplica.• 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 ter mais informações, consulte Replicar objetos criados com criptografia do lado do servidor (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).	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

Alteração	Descrição	Data
Marcação de objetos	<p>O Amazon S3 agora oferece suporte para marcação de objetos. Ela permite classificar o armazenamento. Os prefixos de nome de chave de objeto também permitem classificar o armazenamento, mas a marcação de objetos adiciona outra dimensão a isso.</p> <p>Há benefícios adicionais 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 ter mais informações, consulte Categorizando seu armazenamento usando tags.</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 ter mais informações, consulte Gerenciando seu ciclo de vida de armazenamento .	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 ter mais informações, consulte Monitoramento de métricas com o Amazon CloudWatch .	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 ter mais informações, consulte Inventário do Amazon S3 .	29 de novembro de 2016

Alteração	Descrição	Data
Análise do Amazon S3: análise de classe de armazenamento	O novo recurso de análise de classe de armazenamento do Amazon S3 observa padrões de acesso de dados para ajudar você a determinar quando fazer a transição do armazenamento STANDARD acessado menos frequentemente para a classe de armazenamento STANDARD_IA IA significa “infrequent access”, inglês para “acesso com pouca frequência”). 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 ajudar a melhorar suas configurações 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.	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 ter mais informações, consulte Restaurar um objeto arquivado .	21 de novembro de 2016
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 ter mais informações, consulte Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail .	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

Alteração	Descrição	Data
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 ter mais informações, consulte Conceitos básicos do Amazon S3 Transfer Acceleration .	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 ter mais informações, consulte Fazer solicitações para o Amazon S3 por meio do IPv6 .	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 .	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 do ciclo de vida agora permite que você instrua o Amazon S3 a remover os marcadores de exclusão de objeto expirado em um bucket com versionamento. Para ter mais informações, consulte Elementos para descrever ações de ciclo de vida .	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• Elementos para descrever ações de ciclo de vida <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

Alteração	Descrição	Data
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 para carregamentos fracionados	<p>As políticas do IAM agora oferecem suporte para a chave de condição <code>s3:x-amz-storage-class</code> do Amazon S3. Para ter mais informações, consulte Exemplos de políticas de bucket usando chaves de condição.</p> <p>Você não precisa mais ser o iniciador de um multipart upload para fazer upload de partes e terminar o upload. Para ter mais informações, consulte API de multipart upload e permissões.</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 ter mais informações, consulte Uso de classes de armazenamento do Amazon S3.</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 ter mais informações, consulte Gerenciando seu ciclo de vida de armazenamento.</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 ter mais informações, consulte Replicação de objetos.</p>	16 de setembro de 2015
Integração do AWS CloudTrail	<p>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 buckets do S3, modificações de controle de acesso ou alterações de configuração de ciclo de vida. Para ter mais informações, consulte Registrar chamadas de API do Amazon S3 em log usando AWS CloudTrail.</p>	1 de setembro de 2015

Alteração	Descrição	Data
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 respectiva Conta da AWS. Os clientes que precisarem de buckets adicionais poderão aumentar esse limite enviando um aumento de limite de serviço. Para ter informações sobre como aumentar o limite do bucket, consulte Cotas do AWS service (Serviço da AWS) na Referência geral da AWS. Para obter mais informações, consulte Uso da SDKs AWS e Restrições e limitações do bucket .	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 ter mais informações, consulte Regiões .	4 de agosto de 2015
Notificações de eventos	As Notificações de Eventos 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 ter mais informações, consulte Notificações de eventos do Amazon S3 .	28 de julho de 2015

Alteração	Descrição	Data
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 Standard, total de bytes para Reduced Redundancy Storage e número total de objetos para determinado bucket do S3. Para ter mais informações, consulte Monitoramento de métricas com o Amazon CloudWatch .	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 ter mais informações, consulte Esvaziar um bucket .	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 ter mais informações, consulte Controlar o acesso a partir de VPC endpoints com políticas de bucket .	29 de abril de 2015
Notificações de eventos	As Notificações de Eventos do Amazon S3 foram atualizadas para oferecer suporte à alternância para permissões com base em recursos para funções do AWS Lambda. Para ter mais informações, consulte Notificações de eventos do Amazon S3 .	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 ter mais informações, consulte Replicação de objetos .	24 de março de 2015

Alteração	Descrição	Data
Notificações de eventos	<p>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 <code>s3:ReducedRedundancyLostObject</code> 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>	13 de novembro de 2014
Criptografia do lado do servidor com chaves fornecidas pelo cliente	<p>Criptografia no lado do servidor com chaves do AWS Key Management Service (AWS KMS) (SSE-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	<p>O Amazon S3 já está disponível na região Europa (Frankfurt).</p>	23 de outubro de 2014

Alteração	Descrição	Data
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	<p>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 ter mais informações, consulte Gerenciando seu ciclo de vida de armazenamento.</p>	20 de maio de 2014
Tópicos de controle de acesso revisados	<p>Documentação revisada de controle de acesso do Amazon S3. Para ter mais informações, consulte Gerenciamento de identidade e acesso para o Amazon S3.</p>	15 de abril de 2014
Tópico de registro de acesso de servidor revisado	<p>Documentação revisada de registro de acesso de servidor. Para ter mais informações, consulte Registrar em log as solicitações com registro em log de acesso ao servidor.</p>	26 de novembro de 2013
Exemplos de SDK do .NET atualizados para a versão 2.0	<p>Os exemplos de SDK do .NET neste guia agora são compatíveis com a versão 2.0.</p>	26 de novembro de 2013

Alteração	Descrição	Data
Suporte a SOAP via HTTP descontinuado	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
Suporte para variáveis de política do IAM	<p>Agora, a linguagem de política do IAM oferece suporte a 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 políticas baseadas em identidade e para o Amazon S3.</p>	3 de abril de 2013
Suporte do console para Pagamento pelo solicitante	Agora você pode configurar seu bucket para pagamentos pelo solicitante usando o console do Amazon S3. Para ter mais informações, consulte Configuração de buckets de Pagamento pelo solicitante para transferências de armazenamento e uso .	31 de dezembro de 2012

Alteração	Descrição	Data
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, eles podem usar example.com em vez de www.example.com). Muitos clientes já hospedam sites estáticos no Amazon S3 que podem ser acessados de 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 ter mais informações, consulte Hospedagem de um site estático usando o Amazon S3.</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

Alteração	Descrição	Data
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, defina regras de arquivamento identificando 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 ter mais informações, consulte Gerenciando seu ciclo de vida de armazenamento.</p>	13 de novembro de 2012
Suporte para redirecionamentos de página de site	<p>Para um bucket que é configurado como um site, o Amazon S3 agora oferece suporte ao redirecionamento de uma solicitação de um objeto para outro objeto no mesmo bucket ou um URL externo. Para ter mais informações, consulte (Opcional) Configurar um redirecionamento de uma página da Web.</p> <p>Para obter informações sobre a hospedagem de sites, consulte Hospedagem de um site estático usando o Amazon S3.</p>	4 de outubro de 2012
Suporte para compartilhamento de recursos de origem cruzada (CORS)	<p>O Amazon S3 oferece suporte para compartilhamento de recursos de origem cruzada (CORS). O CORS define uma maneira de os aplicativos web clientes carregados em um domínio poderem interagir com ou acessar recursos em outro domínio. Com suporte a CORS no Amazon S3, você pode criar aplicações Web cliente avançados no Amazon S3 e permitir seletivamente o acesso de domínio cruzado aos recursos do Amazon S3. Para ter mais informações, consulte Usar o compartilhamento de recursos de origem cruzada (CORS).</p>	31 de agosto de 2012

Alteração	Descrição	Data
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 .	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 Exigir MFA.</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 ter mais informações, consulte Acessar e listar um bucket do Amazon S3 .	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 ter mais informações, consulte Excluir objetos do Amazon S3 .	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 .	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 Especificar a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3)).• Nova seção sobre como criar e testar amostras de Ruby (consulte Usar o AWS SDK for Ruby - versão 3).	17 de outubro de 2011

Alteração	Descrição	Data
Suporte para criptografia do lado do servidor	<p>O Amazon S3 agora oferece suporte para criptografia do lado do servidor. Ela permite solicitar ao Amazon S3 para criptografar seus dados em repouso, ou seja, criptografar seus dados de objeto quando o Amazon S3 grava seus dados em discos nos datacenters. Além das atualizações de API REST, o AWS SDK for Java e o .NET oferecem a funcionalidade necessária para solicitar criptografia do lado do servidor. Você também pode solicitar a criptografia de servidor ao fazer upload de objetos usando o Console de Gerenciamento da AWS Management Console. Para saber mais sobre a criptografia de dados, acesse Usar criptografia de dados.</p>	4 de outubro de 2011
Atualização da documentação	<p>Além de correções de erros de documentação, esta versão inclui os seguintes aprimoramentos:</p> <ul style="list-style-type: none">• Adição de exemplos de Ruby e PHP à seção Fazer solicitações.• Adição de seções que descrevem como gerar e usar pre-signed URLs. Para obter mais informações, consulte Compartilhar objetos com URLs pré-assinados e Compartilhar objetos com URLs pré-assinados.• Atualização de uma seção existente para apresentar o AWS Explorer para Eclipse e Visual Studio. Para ter mais informações, consulte Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers.	22 de setembro de 2011

Alteração	Descrição	Data
<p>Suporte para enviar solicitações usando credenciais de segurança temporárias</p>	<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 de 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 ter mais informações, consulte Fazer solicitações.</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>	<p>3 de agosto de 2011</p>
<p>API de multipart upload estendida para permitir a cópia de objetos de até 5 TB</p>	<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, <code>UploadPart (Copy)</code>. Você pode usar essa operação do multipart upload para copiar objetos com até 5 TB. Para ter mais informações, consulte Copiar e mover objetos.</p> <p>Para obter informações conceituais sobre a API do multipart upload, consulte Carregar e copiar objetos usando multipart upload.</p>	<p>21 de junho de 2011</p>
<p>Chamadas de API SOAP via HTTP desativadas</p>	<p>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.</p>	<p>6 de junho de 2011</p>

Alteração	Descrição	Data
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 ter mais informações, consulte Criar um perfil 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 Entidades principais para políticas de bucket.</p>	6 de junho de 2011
Novo link	As informações de endpoint deste serviço agora estão localizadas na Referência geral da AWS. Para ter mais informações, consulte Regiões e endpoints na Referência geral da AWS .	1 de março de 2011
Suporte para a hospedagem de sites estáticos no Amazon S3	O Amazon S3 introduz o suporte aprimorado para hospedagem de sites estáticos. Isso inclui suporte para documentos de índice e documentos de erros personalizados. Ao usar esses recursos, as solicitações para a raiz de seu bucket ou uma subpasta (por exemplo, <code>http://mywebsite.com/subfolder</code>) retornam o documento de índice em vez da lista de objetos em seu bucket. Se um erro for encontrado, o Amazon S3 retornará sua mensagem de erro personalizada em vez de uma mensagem de erro do Amazon S3. Para ter mais informações, consulte Hospedagem de um site estático usando o Amazon S3 .	6 de junho de 2011

Alteração	Descrição	Data
As informações de endpoint deste serviço agora estão localizadas na Referência geral da AWS. Para ter mais informações, consulte Regiões e endpoints na Referência geral da AWS .	1 de março de 2011	
Suporte para a hospedagem de sites estáticos no Amazon S3	O Amazon S3 introduz o suporte aprimorado para hospedagem de sites estáticos. Isso inclui suporte para documentos de índice e documentos de erros personalizados. Ao usar esses recursos, as solicitações para a raiz de seu bucket ou uma subpasta (por exemplo, <code>http://mywebsite.com/subfolder</code>) retornam o documento de índice em vez da lista de objetos em seu bucket. Se um erro for encontrado, o Amazon S3 retornará sua mensagem de erro personalizada em vez de uma mensagem de erro do Amazon S3. Para ter mais informações, consulte Hospedagem de um site estático usando o Amazon S3 .	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 ter mais informações, consulte Baixar objetos .	14 de janeiro de 2011

Alteração	Descrição	Data
Suporte para objetos grandes	O Amazon S3 aumentou o tamanho máximo de um objeto que você pode armazenar em um bucket do S3 de 5 GB para 5 TB. Se estiver usando a API REST, você poderá fazer upload de objetos de até 5 GB em uma única operação PUT. Para objetos maiores, você deve usar a API REST do multipart upload para fazer upload de objetos em partes. Para ter mais informações, consulte Carregar e copiar objetos usando multipart upload .	9 de dezembro de 2010
Multipart upload	O multipart upload permite fazer uploads mais flexíveis mais rapidamente no Amazon S3. Ele permite carregar um único objeto como um conjunto de partes. Para ter mais informações, consulte Carregar e copiar objetos usando multipart upload .	10 de novembro de 2010
Suporte de ID canônico em políticas de bucket	Agora você pode especificar IDs canônicos em políticas de bucket. Para ter mais informações, consulte Entidades principais para políticas de bucket .	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 ter 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 .	14 de julho de 2010

Alteração	Descrição	Data
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 ter mais informações, consulte Políticas de bucket para o Amazon S3 .	6 de julho de 2010
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 .	9 de junho de 2010
Novo endpoint para a Europa (Irlanda)	O Amazon S3 agora fornece um endpoint para a Europa (Irlanda): <code>http://s3-eu-west-1.amazonaws.com</code> .	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) .	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 .	28 de abril de 2010

Alteração	Descrição	Data
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 e Usar versionamento .	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 <code>s3-us-west-1.amazonaws.com</code> . Para obter mais informações, consulte Buckets e regiões .	2 de dezembro de 2009
AWS SDK for .NET	A AWS agora fornece bibliotecas, códigos de exemplo, tutoriais e outros recursos para os desenvolvedores de software que preferem criar aplicativos usando operações de API específicas da linguagem .NET em vez de REST ou SOAP. Essas bibliotecas oferecem as funções básicas (não incluídas nas APIs REST ou SOAP), como autenticação de solicitação, novas tentativas de solicitação e processamento de erros, para que você possa começar a usar com mais facilidade. Para obter mais informações sobre bibliotecas e recursos específicos da linguagem, consulte Desenvolvimento com o Amazon S3 usando os AWS SDKs e Explorers .	11 de novembro de 2009

Glossário do AWS

Para obter a terminologia mais recente da AWS, consulte o [glossário da AWS](#) na Referência do Glossário da AWS.