

# Object Storage Service

## Guia de usuário

**Edição** 01  
**Data** 31-08-2020



**Copyright © Huawei Technologies Co., Ltd. 2023. Todos os direitos reservados.**

Nenhuma parte deste documento pode ser reproduzida ou transmitida em qualquer forma ou por qualquer meio sem consentimento prévio por escrito da Huawei Technologies Co., Ltd.

## **Marcas registadas e permissões**



HUAWEI e outras marcas registadas da Huawei são marcas registadas da Huawei Technologies Co., Ltd.

Todos as outras marcas registadas e os nomes registados mencionados neste documento são propriedade dos seus respectivos detentores.

## **Aviso**

Os produtos, serviços e funcionalidades adquiridos são estipulados pelo contrato feito entre a Huawei e o cliente. Todos ou parte dos produtos, serviços e funcionalidades descritos neste documento pode não estar dentro do âmbito de aquisição ou do âmbito de uso. Salvo especificação em contrário no contrato, todas as declarações, informações e recomendações neste documento são fornecidas "TAL COMO ESTÁ" sem garantias, ou representações de qualquer tipo, seja expressa ou implícita.

As informações contidas neste documento estão sujeitas a alterações sem aviso prévio. Foram feitos todos os esforços na preparação deste documento para assegurar a exatidão do conteúdo, mas todas as declarações, informações e recomendações contidas neste documento não constituem uma garantia de qualquer tipo, expressa ou implícita.

---

# Índice

---

<b>1 Antes de começar.....</b>	<b>1</b>
<b>2 Uso de OBS.....</b>	<b>3</b>
<b>3 Acesso a OBS.....</b>	<b>6</b>
<b>4 Visão geral das funções.....</b>	<b>12</b>
<b>5 Gerenciamento de permissões.....</b>	<b>13</b>
<b>6 Classes de armazenamento.....</b>	<b>14</b>
<b>7 Gerenciamento de bucket.....</b>	<b>19</b>
7.1 Visão geral.....	19
7.2 Criação de um bucket.....	20
7.3 Listagem de buckets.....	22
7.4 Visualização de informações de bucket.....	22
7.5 Gerenciamento de cotas de bucket.....	23
7.6 Exclusão de buckets.....	24
<b>8 Gerenciamento de objetos.....</b>	<b>25</b>
8.1 Visão geral do objeto.....	25
8.2 Criação de uma pasta.....	28
8.3 Carregamento de um objeto.....	29
8.4 Download de um objeto.....	32
8.5 Listagem de objetos.....	33
8.6 Cópia de objetos.....	34
8.7 Carregamento de multiparte.....	35
8.8 Visualização de informações de objetos.....	40
8.9 Compartilhamento de um arquivo.....	41
8.10 Compartilhamento de uma pasta.....	43
8.11 Gerenciamento de metadados de objetos.....	45
8.12 Restauração de objetos a partir do armazenamento de arquivos.....	56
8.13 Leitura direta.....	57
8.14 Exclusão de um objeto.....	57
8.15 Cancelamento da exclusão de um objeto.....	59
8.16 Gerenciamento de fragmentos.....	60

<b>9 Sistema de arquivos paralelos.....</b>	<b>61</b>
<b>10 Processamento de imagens.....</b>	<b>62</b>
<b>11 Gerenciamento do ciclo de vida.....</b>	<b>63</b>
<b>12 Replicação entre regiões.....</b>	<b>74</b>
<b>13 Encriptação do lado do servidor.....</b>	<b>78</b>
<b>14 Hospedagem de site estático.....</b>	<b>80</b>
<b>15 CORS.....</b>	<b>89</b>
<b>16 Vinculação de nome de domínio definido pelo usuário.....</b>	<b>91</b>
<b>17 Volta à fonte.....</b>	<b>93</b>
<b>18 Validação de URL.....</b>	<b>95</b>
<b>19 Tags.....</b>	<b>98</b>
<b>20 Inventário de bucket.....</b>	<b>99</b>
<b>21 Notificação de evento.....</b>	<b>105</b>
<b>22 Registro.....</b>	<b>113</b>
<b>23 Versionamento.....</b>	<b>117</b>
<b>24 Monitoramento.....</b>	<b>121</b>
<b>25 Auditoria.....</b>	<b>123</b>
<b>26 Descompressão online (OBT).....</b>	<b>124</b>
<b>27 Histórico de alterações.....</b>	<b>126</b>

# 1 Antes de começar

Object Storage Service (OBS) fornece uma série de documentos para ajudá-lo a entender e usar melhor o serviço.

Antes de usar OBS, leia este documento e outros documentos de ajuda relacionados (consulte [Tabela 1-1](#)) para entender os conceitos básicos, cenários de aplicação e operações de OBS, para que você possa começar rapidamente a usar o OBS.

**Tabela 1-1** Documentos de OBS

Documento	Descrição
<a href="#">Conhecimento progressivo</a>	Fornecer links para <i>Início rápido</i> , <i>Guia de usuário</i> , <i>Melhores práticas</i> , <i>Perguntas frequentes</i> , e <i>Soluções</i> , para que você possa localizar com facilidade e precisão o conteúdo que deseja visualizar.
<a href="#">Últimas notícias</a>	Descreve as últimas atualizações de OBS, incluindo novas funções e documentos.
<a href="#">Visão geral das funções</a>	Uma introdução geral às funções de OBS e regiões onde essas funções estão disponíveis.
<a href="#">Visão geral de serviço</a>	Conceitos básicos, vantagens, cenários de aplicativos, faturamento e gerenciamento de permissões para OBS.
<a href="#">Introdução</a>	Um guia rápido para ajudá-lo a começar com operações básicas, como a criação de buckets, o carregamento de objetos, e o download de objetos.
<a href="#">Guia de operação do Console</a>	Guia você pelas operações do console de OBS com exemplos de uso elaborados.
<a href="#">Guia de ferramentas</a>	Introduz as ferramentas de OBS, incluindo OBS Browser+, obsutil, e obsfs, com exemplos de uso fornecidos.
<a href="#">Guia de recursos de processamento de imagem</a>	Um guia de operação para o processamento de imagens de OBS. Ele aborda conceitos básicos, funções e perguntas frequentes e demonstra como usar esse recurso por meio do console de OBS ou chamada das API de OBS.

<b>Documento</b>	<b>Descrição</b>
<b>Guia de recursos do sistema de arquivos paralelo</b>	Apresenta os conceitos básicos, cenários de aplicativos, limitações, operações de usuário e faturamento para o sistema de arquivos paralelo de OBS (PFS).
<b>Melhores práticas</b>	Uma coleção de melhores práticas de OBS para cenários típicos, ajudando você a atingir suas metas de negócios com melhor desempenho, custos mais baixos e operações mais convenientes.
<b>Referências de API</b>	Lista as API REST fornecidas pelo OBS, incluindo amostras de solicitações, amostras de resposta e descrições de parâmetros. Ele ajuda você a usar as API para conduzir o desenvolvimento secundário.
<b>Referência de SDK</b>	Descreve os kits de desenvolvimento de software (os SDK) do OBS em linguagens de programação convencionais, como Java, Python, C, GO, Android e iOS. Os SDK fornecem detalhes de parâmetros, exemplos e perguntas frequentes para ajudá-lo a conduzir sem problemas a instalação e o desenvolvimento.
<b>Perguntas frequentes</b>	Oferece-lhe as respostas às perguntas mais frequentes sobre OBS.
<b>Vídeos</b>	Tutoriais em vídeo sobre as funções básicas de OBS são fornecidos para sua referência.

# 2

## Uso de OBS

Você pode gerenciar recursos de OBS das seguintes maneiras:

Ferramenta	Descrição	Como usar	Referência
Console de OBS	Console de OBS é uma GUI baseada na web. Você pode gerenciar todos os seus recursos de OBS através deste console.	Você precisa de uma conta (incluindo a senha) ou um usuário de IAM para fazer login no Console de OBS. Para obter detalhes, consulte <a href="#">Efetuando logon no Console de OBS</a> .	<a href="#">Guia de operação do Console</a>
SDK	Os SDK de OBS encapsulam as API fornecidas por OBS para simplificar o desenvolvimento. Os usuários podem usar as funções da API fornecidas pelos SDK de OBS para acessar os recursos do serviço de OBS.	Configure o ponto final durante a preparação do ambiente e defina as chaves de acesso (AK/SK) durante a inicialização. Para obter detalhes, consulte <a href="#">Referência de SDK</a> .	<a href="#">Referência de SDK</a>

Ferramenta	Descrição	Como usar	Referência
API	OBS fornece as API REST que suportam solicitações de HTTP/HTTPS. Você pode chamar essas API para criar, modificar e excluir buckets, bem como para carregar, baixar e excluir objetos.	<p>Configure o ponto final durante a preparação do ambiente e torne as chaves de acesso (AK e SK) parte de sua assinatura em todas as solicitações enviadas. Se você usar as API REST para desenvolver um programa, precisará calcular a assinatura com base no algoritmo de assinatura definido por OBS e adicionar a assinatura às solicitações.</p> <p>Para obter detalhes, consulte <a href="#">Autenticação de assinatura de usuário</a>.</p>	<a href="#">Referências de API</a>
OBS Browser+	OBS Browser+, uma ferramenta GUI, é um cliente de Windows para gerenciar recursos de OBS. Com OBS Browser+, você pode gerenciar facilmente os recursos de OBS a partir de uma extremidade local.	<p>Baixe OBS Browser+ e use as teclas de acesso (AK e SK) para autenticação de identidade. Você pode usar OBS Browser+ para acessar diretamente os recursos de OBS ou pode configurar um endereço de servidor para acessar os recursos de OBS.</p> <p>Para obter detalhes sobre como configurar o AK, SK e endpoint de OBS Browser+, consulte <a href="#">Onde posso obter chaves de acesso (AK e SK)</a>.</p>	<a href="#">Introdução a OBS Browser+</a>
obsutil	obsutil é uma ferramenta de linha de comando para acessar o OBS. Você pode usá-lo para executar operações comuns de configuração e gerenciamento em OBS. Se você estiver familiarizado com a interface de linha de comando (CLI), obsutil é a ferramenta recomendada para processamento em lote e tarefas automatizadas.	<p>Faça o download de obsutil, configure o endereço do servidor e use as chaves de acesso (AK e SK) para autenticação de identidade.</p> <p>Para obter detalhes sobre como configurar o AK, o SK e o endpoint do obsutil, consulte <a href="#">Execução da configuração inicial</a>.</p>	<a href="#">Introdução ao obsutil</a>



Ferramenta	Descrição	Como usar	Referência
obsfs	obsfs, construído com base no Filesystem in Userspace (FUSE), é uma ferramenta de sistema de arquivos fornecida por OBS para montar sistemas de arquivos paralelos nos SO de Linux. Ele permite que você acesse facilmente o espaço de armazenamento praticamente ilimitado em OBS da mesma maneira que você usaria um sistema de arquivos local.	Faça o download da ferramenta obsfs e use as chaves de acesso (AK e SK) para autenticação de identidade.  Para obter detalhes sobre como configurar a AK e a SK de obsfs, consulte <a href="#">Inicialização de obsfs</a> .	<a href="#">Introdução a obsfs</a>

# 3 Acesso a OBS

## Nome de domínio de OBS

Os dois conceitos a seguir estão relacionados aos nomes de domínio de OBS:

1. **Endpoint:** OBS fornece um endpoint para cada região. Um endpoint é um nome de domínio para acessar o OBS em uma determinada região e é usado para receber solicitações de acesso enviadas dessa região. Para obter uma lista completa dos endpoints do OBS e as regiões correspondentes, consulte [Regiões e endpoints](#).
2. **Nome de domínio de bucket:** Cada bucket em OBS tem um nome de domínio. Um nome de domínio é o endereço da Internet de um bucket e pode ser usado para acessar o bucket pela internet. É normalmente usado em cenários de desenvolvimento de aplicativos em nuvem e compartilhamento de dados.

Um nome de domínio de bucket de OBS está no formato *BucketName.Endpoint*.

*BucketName* indica o nome de um bucket e *Endpoint* indica o nome de domínio de OBS da região onde o bucket está localizado.

**Tabela 3-1** lista o nome de domínio do bucket e outros nomes de domínio de OBS, incluindo seus formatos e protocolos.

**Tabela 3-1** Nomes de domínio de OBS

Tipo	Estrutura	Descrição	Tipo de protocolo
Nome de domínio da região	[Estrutura] <b>Endpoint</b> [Exemplo] obs.cn-north-4.myhuaweicloud.com	Cada região tem um endpoint, que é o nome de domínio da região. Para obter um mapeamento completo entre regiões e endpoints de OBS, consulte <a href="#">Regiões e endpoints</a> .	HTTPS HTTP

Tipo	Estrutura	Descrição	Tipo de protocolo
Nome de domínio de bucket	[Estrutura] <b>BucketName.Endpoint</b> [Exemplo] bucketname.obs.cn-north-4.myhuaweicloud.com	Depois que um bucket é criado, você pode usar o nome de domínio para acessar o bucket. Você pode montar o nome de domínio do bucket colocando o nome do bucket e o ponto de extremidade de serviço de OBS juntos, ou você pode obtê-lo de <a href="#">Visualização de informações de bucket</a> no Console do OBS e no OBS Browser+.	HTTP HTTPS
Nome de domínio de objeto	[Estrutura] <b>BucketName.Endpoint/ObjectName</b> [Exemplo] bucketname.obs.cn-north-4.myhuaweicloud.com/object.txt	Depois que um objeto é carregado em um bucket, você pode usar o nome de domínio do objeto para acessar o objeto. Você pode montar o nome de domínio do objeto colocando o nome do bucket, o ponto de extremidade do serviço do OBS e o nome do objeto juntos, ou você pode obtê-lo através de <a href="#">Visualização de informações de objetos</a> no Console do OBS ou no OBS Browser+. Como alternativa, você pode chamar a API <code>GetObjectUrl</code> por meio de um SDK para obter o nome de domínio do objeto.	HTTP HTTPS
<a href="#">Hospedagem de site estático</a>	[Estrutura] <b>BucketName.obs-website.Endpoint</b> [Exemplo] bucketname.obs-website.cn-north-4.myhuaweicloud.com	Um nome de domínio de site estático é um nome de domínio de bucket quando o bucket é configurado para hospedar um site estático.	HTTP HTTPS
<a href="#">Vinculação de nome de domínio definido pelo usuário</a>	Nomes de domínio que foram licenciados pelo Ministério da Indústria e Tecnologia da Informação (MIIT) da China.	Você pode vincular um nome de domínio de usuário a um bucket para poder acessar o bucket por meio do nome de domínio de usuário.	HTTP

## Endpoints

OBS tem um endpoint em cada região.

Geralmente, o endpoint realizado em uma solicitação de acesso ao OBS deve ser o endpoint da região onde reside o recurso solicitado. No entanto, em alguns casos especiais, você pode usar qualquer endpoint.

### 1. Cenários em que o endpoint na solicitação deve ser consistente com o endpoint da região onde residem os recursos solicitados

Ao acessar um bucket ou um objeto, o endpoint na solicitação deve ser o endpoint da região onde o bucket ou objeto reside.

Por exemplo, se o bucket **mybucket** estiver na região **cn-north-4**, você poderá listar objetos no bucket enviando uma solicitação mostrada no exemplo a seguir:

#### A correct example of request and response for listing objects:

##### [Solicitação]

```
GET / HTTP/1.1
Host: mybucket.obs.cn-north-4.myhuaweicloud.com
Accept: */*
Date: Thu, 10 Mar 2016 08:51:25 GMT
Authorization: authorization
```

##### [Resposta]

```
HTTP/1.1 200 OK
x-obs-request-id: 0001EF710C000001536176DA465E4E6G
x-obs-id-2: Rdj0zZvRkihRcjCQUqjKDGt8JuAgi2CGuLiP7Pv/cYYplsS0xTFJQHP5vSg5yOYC
Content-Type: application/xml
Date: Thu, 10 Mar 2016 16:58:12 GMT
x-obs-bucket-location: cn-north-4
Content-Length: 259

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ListBucketResult xmlns="http://obs.myhuaweicloud.com/doc/2015-06-30/">
  <Name>mybucket</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>>false</IsTruncated>
  <Contents>
    <Key>object001</Key>
    <LastModified>2015-07-01T00:32:16.482Z</LastModified>
    <ETag>"2fa3bcaaec668adc5da177e67a122d7c"</ETag>
    <Size>12041</Size>
    <Owner>
      <ID>b4bf1b36d9ca43d984fbc9491b6fce9</ID>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
</ListBucketResult>
```

Se o endpoint na solicitação não for consistente com o endpoint da região solicitada, uma mensagem de erro será retornada indicando que o bucket não existe.

No exemplo anterior, o bucket **mybucket** está na região **cn-north-4**. Se você usar o endpoint **cn-south-1 (mybucket.obs.cn-south-1.myhuaweicloud.com)** para acessar o bucket, o HTTP 404 será retornado, indicando que o bucket não existe. Nesse caso, você pode chamar a API para [Obtenção da localização de bucket](#) para obter a ID de região do bucket e reenviar a solicitação.

#### Um exemplo incorreto de solicitação e resposta para listar objetos:

##### [Solicitação]

```
GET / HTTP/1.1
Host: mybucket.obs.cn-south-1.myhuaweicloud.com
Accept: */*
Date: Thu, 10 Mar 2016 08:51:25 GMT
Authorization: authorization
```

[Resposta]

```
HTTP/1.1 404 NoSuchBucket
x-obs-request-id: 0001EF710C000001536176DA465E4E6H
x-obs-id-2: Rdj0zZvRkihRcjCQUqjKdGt8JuAgi2CGuLiP7Pv/cYYplsS0xTFJQHP5vSg5yOYL
Date: Thu, 10 Mar 2016 08:51:30 GMT
Content-Length: 0
```

2. **Cenários em que qualquer endpoint pode ser usado em uma solicitação**

Para obter a lista de buckets ou as informações de região de um bucket, você pode usar qualquer ponto de extremidade de região nas solicitações de API enviadas, pois essas API pesquisam os buckets solicitados em todas as regiões.

**Um exemplo de solicitação e resposta para obter informações de região de um bucket:**

Por exemplo, se o bucket **mybucket** estiver na região **cn-north-4** e o endpoint da região **cn-south-1** for usado na solicitação, as informações de localização do bucket ainda poderão ser obtidas.

[Solicitação]

```
GET /?location HTTP/1.1
Host: mybucket.obs.cn-south-1.myhuaweicloud.com
Accept: */*
Date: Thu, 10 Mar 2016 08:51:25 GMT
Authorization: authorization
```

[Resposta]

```
HTTP/1.1 200 OK
x-obs-request-id: 0001EF710C000001536176DA465E4E6G
x-obs-id-2: Rdj0zZvRkihRcjCQUqjKdGt8JuAgi2CGuLiP7Pv/cYYplsS0xTFJQHP5vSg5yOYC
Content-Type: application/xml
Date: Thu, 10 Mar 2016 16:58:12 GMT
Content-Length: length
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Location xmlns="http://obs.myhuaweicloud.com/doc/2015-06-30/">cn-north-4</
Location>
```

## Acesso ao OBS pela internet

Acessar OBS em uma rede pública pode gerar tráfego de entrada para operações de gravação (por exemplo, carregar dados para o OBS), bem como tráfego de saída para operações de leitura (por exemplo, baixar dados de OBS). O tráfego de entrada não incorre em taxas, mas o tráfego de saída sim.

Para obter detalhes sobre o preço do tráfego de saída pela Internet, consulte [Detalhes de preço do produto](#).

Se aceder ao OBS através de uma rede pública, pode utilizar uma URL para especificar recursos no OBS. Um URL de OBS é estruturado da seguinte forma:

**Protocol://[BucketName.]Endpoint[:Port]/[Object][?Param]**

**Tabela 3-2** Parâmetros em um URL de OBS

Parâmetro	Descrição	Obrigatório ou opcional
Protocolo	Indica o protocolo usado para enviar solicitações, que pode ser de HTTP ou HTTPS. HTTPS é um protocolo que garante acesso seguro aos recursos. O OBS suporta HTTP e HTTPS.	Obrigatório
BucketName	Nome do bucket solicitado, que identifica exclusivamente um bucket no OBS.	Opcional
Endpoint	Nome de domínio (endpoint) da região onde o bucket de OBS está localizado.  Para obter detalhes sobre o nome de domínio de OBS de cada região, consulte <a href="#">Regiões e endpoints</a> .	Obrigatório
Porta	Indica a porta habilitada para protocolos usados para enviar solicitações. O valor varia com a implantação do servidor de software. Se nenhum número de porta for especificado, o protocolo usará o valor padrão. Cada protocolo de transmissão tem seu número de porta padrão.  No OBS, o número da porta HTTP padrão é <b>80</b> e o do HTTPS é <b>443</b> .	Opcional
Objeto	Caminho do recurso do objeto solicitado.	Opcional
Param	Recurso específico contido por um bucket ou objeto. O valor padrão desse parâmetro indica que o próprio bucket ou objeto é obtido.	Opcional

Exemplo: Você tem um bucket chamado **mybucket** na região CN North-Beijing4 (**cn-north-4**). O bucket contém um objeto chamado **myfolder/myfile.txt**. O URL para acessar o objeto pela rede pública é a seguinte:

<https://mybucket.obs.cn-north-4.myhuaweicloud.com/myfolder/myfile.txt>

#### **NOTA**

Todas as solicitações de API, exceto a de listagem de objetos, devem conter o **BucketName**. Considerando o desempenho e a confiabilidade da resolução de DNS, o OBS exige que o nome do bucket preceda o **Endpoint** quando uma solicitação que transporta um nome de bucket for construída para formar um nome de domínio de três níveis, também mencionado como nome de domínio de acesso de hospedagem virtual.

## Acesso a OBS pela intranet

O acesso ao OBS através de uma rede privada refere-se ao acesso ao OBS através da rede de comunicação interna entre diferentes serviços da HUAWEI CLOUD. Acessar o OBS através de uma rede privada pode gerar tráfego de entrada para operações de gravação (por exemplo, carregar dados para o OBS), bem como tráfego de saída para operações de leitura (por

exemplo, baixar dados do OBS). Nesse caso, nem o tráfego de entrada nem de saída incorre em taxas.

Por exemplo, você pode acessar o OBS de um Elastic Cloud Server (ECS) em uma rede privada. Esse acesso não é suscetível a problemas de qualidade da rede pública e também reduz custos.

O OBS fornece as melhores práticas para configuração. Para obter detalhes, consulte [Acesso OBS pela intranet](#).

## Verificação da versão do OBS (OBS 2.0 ou OBS 3.0)

A arquitetura de OBS passou por duas gerações: OBS 2.0 e OBS 3.0. Um bucket recém-criado é armazenado no OBS 3.0 por padrão, e a versão do bucket é o OBS 3.0. No entanto, os buckets criados anteriormente ainda estão no OBS 2.0.

Os recursos e funções básicos de OBS são suportados pelo OBS 3.0 e pelo OBS 2.0. Alguns novos recursos são suportados apenas pelo OBS 3.0, como processamento de imagem e replicação entre regiões.

Você pode verificar a versão do bucket no Console do OBS ou usar a API **Head Bucket** para verificar se seu bucket está no OBS 2.0 ou no OBS 3.0. Os detalhes estão a seguir:

**Método 1: Efetue login no console de OBS e verifique as informações básicas do bucket.**

Se **Bucket Version** for **3.0**, o bucket será armazenado no OBS 3.0. Caso contrário, o bucket é armazenado no OBS 2.0.

**Método 2: Use a API Head Bucket para verificar a versão do bucket.**

Solicitação de exemplo:

```
HEAD / HTTP/1.1
Host: bucketname.obs.cn-north-4.myhuaweicloud.com
Accept: */*
Date: WED, 01 Jul 2015 02:23:25 GMT
Authorization: auth string
```

Resposta de exemplo:

```
HTTP/1.1 200 OK
Server: OBS
x-obs-request-id: BF2600000163D80E4C5F20FDD5BD0085
Content-Type: application/xml
x-obs-version: 3.0
x-obs-id-2: 32AAAQAAEAABAAAQAAEAABAAAQAAEAABCS8wS9100114oMwMdniv7XmdAvfewrQq
Date: WED, 01 Jul 2015 02:23:25 GMT
Content-Length: 0
```

Na mensagem de resposta, **x-obs-version: 3.0** indica que o bucket está armazenado no OBS 3.0. Se esse cabeçalho não existir ou o valor desse cabeçalho for exibido de outra forma, o bucket será armazenado no OBS 2.0.

Para obter detalhes sobre a API **Head Bucket**, consulte [Obtenção de metadados de bucket](#).

# 4 Visão geral das função

---

Para obter detalhes, consulte [Visão geral das função de OBS](#).



# 5 Gerenciamento de permissões

## Cenário de aplicação

Por padrão, os recursos de OBS (buckets e objetos) são privados. Somente os proprietários de recursos podem acessar seus recursos de OBS. Sem autorização, outros usuários não podem acessar seus recursos de OBS. O controle de permissão OBS significa conceder permissões a outras contas ou usuários do IAM editando políticas de acesso. Por exemplo, se você tiver um bucket, poderá autorizar outro usuário do IAM a fazer upload de objetos para o bucket. Você também pode abrir buckets para o público, para que qualquer pessoa possa acessar seus buckets pela Internet. OBS oferece vários métodos para ajudá-lo a atribuir permissões de recursos a outras pessoas. Os proprietários de recursos podem formular políticas de controle de permissões diferentes com base nos requisitos de serviço para garantir a segurança dos dados.

## Como usar essa função

Você pode usar console de OBS, as API, os SDK, OBS Browser+, ou obsutil para configurar permissões.

Ferramenta	Referência
Console de OBS	<a href="#">Controle de permissão</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Permissões e ações suportadas</a> <a href="#">Configuração de uma política de bucket</a> <a href="#">Configuração de uma ACL de bucket</a> <a href="#">Configuração de uma ACL de objeto</a>
OBS Browser+	-
obsutil	<a href="#">Configuração de propriedades de bucket</a> <a href="#">Configuração de propriedades de objeto</a>

# 6 Classes de armazenamento

## Cenário de aplicação

OBS fornece três classes de armazenamento: Padrão, Acesso Infrequente e Arquivo.

Diferentes classes de armazenamento atendem a diferentes requisitos de desempenho e custos de armazenamento.

- A classe de armazenamento Padrão apresenta baixa latência de acesso e alta taxa de transferência. É, portanto, ideal para armazenar grandes quantidades de arquivos quentes (frequentemente acessados todos os meses) ou arquivos pequenos (menos de 1 MB). Seus cenários de aplicativos incluem análise de big data, aplicativos móveis, vídeos quentes e aplicativos sociais.
- A classe de armazenamento Acesso Infrequente é ideal para armazenar dados que são acessados com pouca frequência (menos de 12 vezes por ano), mas têm requisitos de resposta rápida. Seus cenários de aplicativos incluem sincronização de arquivos, compartilhamento de arquivos e backup corporativo.
- A classe de armazenamento Arquivo é ideal para arquivar dados raramente acessados (uma vez por ano, em média). Seus cenários de aplicação incluem arquivamento de dados e backups de dados de longo prazo. A classe de armazenamento Arquivo é segura, durável e barata, e pode ser usada para substituir bibliotecas de fitas. No entanto, pode levar horas para restaurar dados da classe de armazenamento Arquivo.

## Comparação de classes de armazenamento

Item comparado	Padrão	Acesso Infrequente	Arquivo
Características	Desempenho de alto nível, altamente confiável e disponível	Acesso de armazenamento confiável, barato e em tempo real	Armazenamento de longo prazo para dados arquivados a um custo muito baixo

Item comparado	Padrão	Acesso Infrequente	Arquivo
Cenários de aplicação	Aplicativos hospedados na nuvem, compartilhamento de dados, compartilhamento de conteúdo e armazenamento de dados a quente	Aplicativos de disco da Web, backup corporativo, arquivamento ativo e monitoramento de dados	Arquivo, armazenamento de imagens médicas, armazenamento de vídeo e substituição de bibliotecas de fitas
Durabilidade	99,999999999%	99,999999999%	99,999999999%
Durabilidade (Multi-AZ)	99,999999999%	99,999999999%	Multi-AZ não suportado
Disponibilidade	99,99%	99%	99%
Disponibilidade (Multi-AZ)	99,995%	99,5%	Multi-AZ não suportado
Unidade de medição mínima <sup>a</sup>	64 KB	64 KB	64 KB
Duração mínima de armazenamento <sup>b</sup>	N/D	30 dias	90 dias
Restauração de dados	N/A	Cobrado de acordo com a quantidade de dados restaurados (medido em GB)	Duas opções de restauração: urgente e padrão Cobrado de acordo com a quantidade de dados restaurados (medido em GB)
Processamento de imagem	Compatível	Compatível	Incompatível

 **NOTA**

a: Unidade de medida mínima refere-se ao limite de tamanho inferior para faturamento de objetos. Por exemplo, se você carregar um objeto padrão de 32 KB (menor que a unidade de medida mínima de 64 KB), será cobrado por um objeto de 64 KB.

b: A duração mínima de armazenamento refere-se ao limite inferior de armazenamento para cobrança de objetos. Por exemplo, se um objeto de Acesso Infrequente tiver sido armazenado no OBS por 20 dias (menor que a duração mínima de armazenamento de 30 dias) e, em seguida, excluído, você será cobrado por 30 dias.

## Classes de armazenamento de bucket e classes de armazenamento de objeto

Você pode especificar a classe de armazenamento para um bucket ao criar o bucket. Você também pode alterar a classe de armazenamento de um bucket após a criação do bucket.

Um objeto herda a classe de armazenamento do bucket onde é carregado. Você pode especificar uma classe de armazenamento para um objeto ao carregá-lo ou pode alterar a classe de armazenamento de objeto depois que o objeto é carregado.

Alterar a classe de armazenamento de um bucket não altera as classes de armazenamento de objetos existentes no bucket, mas os objetos recém-carregados herdam a nova classe de armazenamento por padrão.

**NOTA**

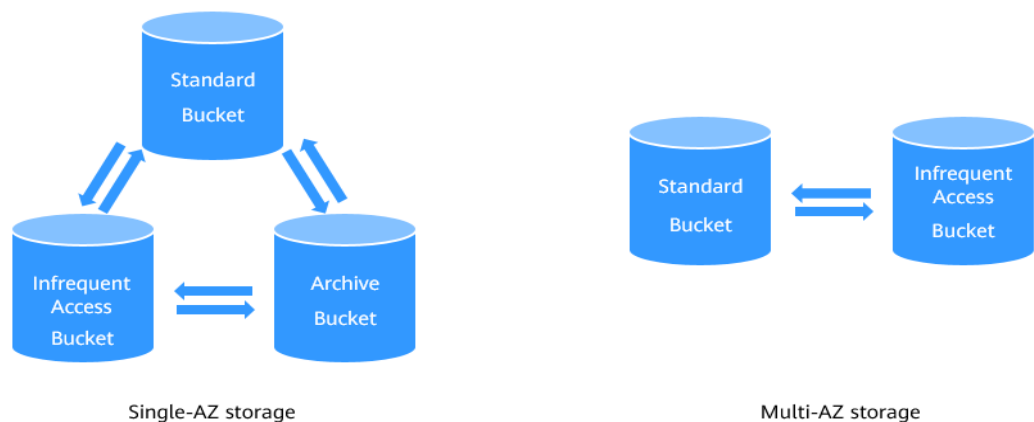
Por padrão, os objetos na classe de armazenamento Arquivo devem ser restaurados para a classe de armazenamento Padrão antes de poderem ser baixados. Você pode habilitar **Leitura Direta** para baixar objetos diretamente na classe de armazenamento Arquivo sem restaurá-los primeiro. Leitura Direta será cobrada.

## Alteração das classes de armazenamento de bucket

A classe de armazenamento de um bucket só pode ser alterada manualmente.

**Figura 6-1** lista as regras de transição da classe de armazenamento do bucket. As classes de armazenamento OBS Padrão e Acesso Infrequente suportam as diretivas de redundância de dados single-AZ e multi-AZ, enquanto a classe de armazenamento Arquivo suporta apenas a diretiva de redundância single-AZ. A política de redundância de dados de um bucket não pode ser modificada, mesmo quando sua classe de armazenamento é alterada.

**Figura 6-1** Regras de alteração da classe de armazenamento de bucket



**Nota:**

- Alteração da classe de armazenamento de um bucket não altera as classes de armazenamento de objetos existentes no bucket. A classe de armazenamento de um objeto carregado posteriormente herdará a nova classe de armazenamento do bucket por padrão. Você também pode **configurar regras do ciclo de vida** para alterar classes de armazenamento de objetos em um lote.

Por exemplo, se **bucket1** estiver na classe de armazenamento Padrão e contiver **object1**, quando a classe de armazenamento do **bucket1** for transferida para Acesso Infrequente, **object1** ainda estará na classe de armazenamento Padrão. Se você fizer upload de **object2** para **bucket1** após a transição, **object2** estará na classe de armazenamento Acesso infrequente.

- Se a classe de armazenamento de um bucket for alterada de Arquivo para Padrão ou Acesso Infrequente, os objetos no bucket não serão restaurados automaticamente da classe de armazenamento Arquivo.

## Alteração de classes de armazenamento de objetos

A classe de armazenamento de um objeto pode ser alterada manualmente ou automaticamente. **Figura 6-2** lista as regras de transição da classe de armazenamento de objetos.

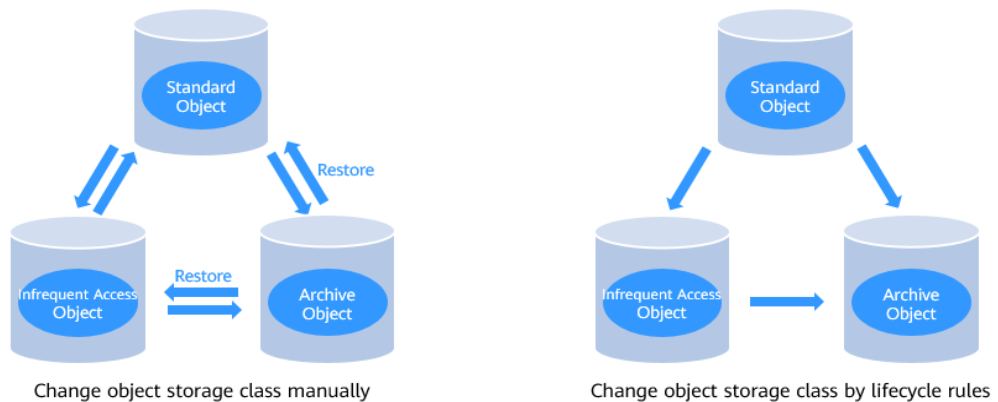
- Transição manual: Depois que um objeto é carregado, você pode alterar manualmente suas classes de armazenamento.

Os objetos podem ser alterados manualmente da classe de armazenamento Padrão para Acesso Infrequente ou Arquivo, ou da classe de armazenamento Acesso Infrequente para Padrão ou Arquivo, mas os objetos na classe de armazenamento Arquivo devem ser restaurados antes de serem alterados manualmente para a classe de armazenamento Padrão ou Acesso Infrequente. A transição de objetos na classe de armazenamento Acesso Infrequente ou Arquivo incorre em taxas de recuperação. Selecione uma opção de transição apropriada com base em suas necessidades.

- Transição automática: O OBS fornece regras de ciclo de vida para que você faça a transição automática de objetos de uma classe de armazenamento para outra. Para mais detalhes, consulte **Gerenciamento do ciclo de vida**.

Os objetos podem ser alterados automaticamente da classe de armazenamento Padrão para Acesso Infrequente ou Arquivo, ou da classe de armazenamento Acesso Infrequente to Arquivo.

**Figura 6-2** Regras de alteração da classe de armazenamento de objetos



### Nota:

- Depois que os objetos na classe de armazenamento Padrão são transferidos para a classe de armazenamento Acesso infrequente ou Arquivo, seu status de restauração é Não Restaurado.
- Objetos na classe de armazenamento Acesso Infrequente têm um período de armazenamento mínimo de 30 dias e aqueles na classe de armazenamento Arquivo têm um período de armazenamento mínimo de 90 dias. Se um objeto for transferido para outra classe de armazenamento antes de ser armazenado pelo período mínimo de armazenamento necessário, você precisará pagar pela duração total do armazenamento (30 dias ou 90 dias).

## Como usar essa função

Você pode usar Console do OBS, as API, os SDK, OBS Browser+, ou obsutil para configurar classes de armazenamento para buckets e objetos.

Ferramenta	Referência
Console de OBS	<a href="#">Criação de um bucket</a> <a href="#">Carregamento de um arquivo</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Configuração da classe de armazenamento para um bucket</a> <a href="#">Criação de um bucket</a> (adicione a configuração de cabeçalho <b>x-obs-storage-class</b> ). <a href="#">Carregamento de objetos - PUT</a> (adicione a configuração de cabeçalho <b>x-obs-storage-class</b> ). <a href="#">Carregamento de objetos - POST</a> (adicione a configuração de cabeçalho <b>x-obs-storage-class</b> ).
OBS Browser+	-
obsutil	<a href="#">Criação de um bucket</a> <a href="#">Configuração de propriedades do bucket</a> <a href="#">Carregamento de um objeto</a> <a href="#">Configuração de propriedades do objeto</a>

# 7 Gerenciamento de bucket

---

## 7.1 Visão geral

Um bucket é um contêiner virtual usado para armazenar **objetos** em OBS. O OBS fornece armazenamento simples na forma de buckets e objetos. Diferente da estrutura de diretórios de várias camadas convencional dos sistemas de arquivos, todos os objetos em um bucket são armazenados na mesma camada lógica.

Cada bucket tem seus próprios atributos, como classe de armazenamento (para mais informações, consulte **Classes de armazenamento**), permissões de acesso, e **regiões**. Você pode especificar permissões de acesso, classe de armazenamento, e regiões ao criar buckets. Você também pode configurar atributos avançados para atender aos requisitos de armazenamento em diferentes cenários.

OBS fornece as seguintes classes de armazenamento: Padrão, acesso pouco frequente, e arquivamento. Com suporte para diferentes classes de armazenamento, OBS atende a diversos requisitos de desempenho e custo de armazenamento. Ao criar um bucket, você pode especificar uma classe de armazenamento para o bucket, que pode ser alterada posteriormente.

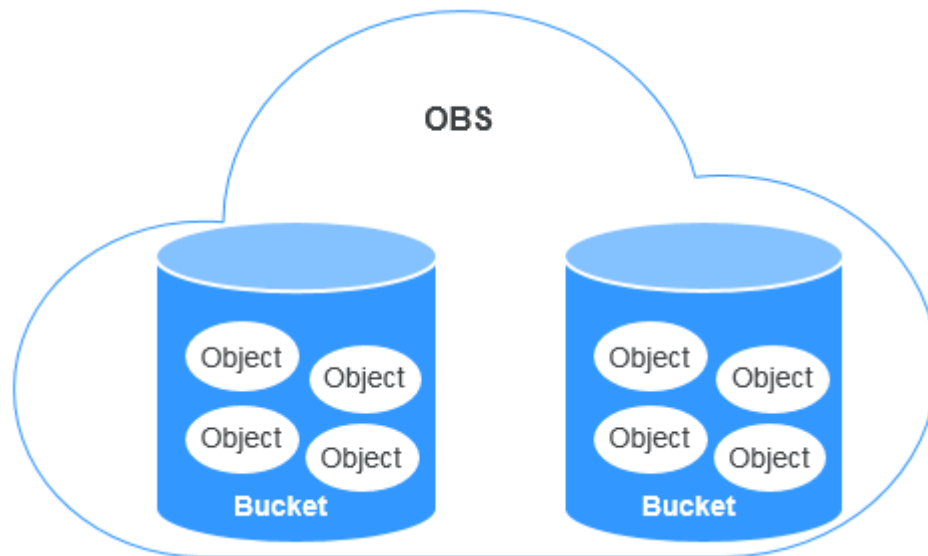
Em OBS, cada nome de bucket deve ser exclusivo e não pode ser alterado. A região onde um bucket reside não pode ser alterada depois que o bucket é criado. Quando um bucket é criado, a lista de controle de acesso (ACL) do bucket é gerada por padrão. A ACL é usada para conceder permissões de recursos aos usuários. Somente usuários autorizados podem executar operações como criar, excluir, exibir e configurar buckets.

Uma conta (incluindo todos os usuários do IAM sob a conta) pode criar um máximo de 100 buckets. Você pode aproveitar o recurso de controle de permissão refinado de OBS para planejar e usar adequadamente os buckets. Por exemplo, você pode criar pastas em um bucket com base em prefixos de objeto e usar **versionamento refinado** para isolar dados entre diferentes departamentos.. Não há restrição quanto à quantidade e ao tamanho total dos objetos em um bucket.

Como o OBS é baseado em uma arquitetura RESTful sobre HTTP e HTTPS, você pode usar localizadores de recursos uniformes (os URL) para localizar recursos.

**Figura 7-1** ilustra a relação entre buckets e objetos no OBS.

Figura 7-1 Buckets e objetos



Você pode **usar métodos diferentes** para **criar buckets** com base em seus hábitos de uso e necessidades de armazenamento. Depois que um bucket é criado, você pode usar diferentes maneiras para **carregar arquivos (dados) para o bucket**, onde esses arquivos são armazenados como objetos. No OBS, baldes e objetos estão localizados em **regiões diferentes**. Você pode usar métodos diferentes para acessar o mesmo bucket e recursos na mesma região.

## 7.2 Criação de um bucket

### Cenário de aplicação

Os buckets são containers que armazenam objetos em OBS. Você precisa criar um bucket antes de começar a armazenar dados em OBS.

### Pré-requisitos

Para criar um bucket, verifique se você tem uma conta registrada, saldo suficiente na conta, chaves de acesso (AK e SK), e as informações do endpoint. Para obter detalhes, consulte [Introdução](#).

### Restrições

- Depois que um bucket é criado, nome e região dele não podem ser alterados. Especifique uma região adequada e um nome de bucket ao criar o bucket.
- Uma conta (incluindo todos os usuários do IAM sob a conta) pode criar um máximo de 100 buckets. Você pode aproveitar o recurso de controle de permissão refinado de OBS para planejar e usar adequadamente os buckets. Por exemplo, você pode criar pastas em um bucket com base em prefixos de objeto e usar **controle de permissão refinado** para implementar o isolamento de permissão entre departamentos. No entanto, não há restrição quanto ao número e tamanho total de objetos em um bucket.
- Um nome de bucket faz parte do nome de domínio de acesso e precisa ser resolvido. Portanto, o nome do bucket deve estar em conformidade com as regras de nomeação de



domínio de DNS. Ao receber uma solicitação de criação de bucket, OBS verifica estritamente o nome do bucket. Nome de um bucket:

- Deve ser exclusivo em todas as contas e regiões.
- Deve ter de 3 a 63 caracteres. Use somente letras minúsculas, dígitos, hífens (-), e pontos finais (.). Se um nome de bucket contiver um período (.), a verificação do certificado de segurança pode ser acionada quando você acessar o bucket ou objetos.
- Um nome de bucket não pode começar ou terminar com um período (.) ou hífen (-), e não pode conter dois períodos consecutivos (..) e um hífen adjacentes.
- Não pode ser formatado como um endereço IP.

#### AVISO

Se você usar um bucket com períodos (.) no nome para acessar a OBS, o cliente exibirá uma mensagem indicando que o bucket é arriscado, por exemplo, um alarme vermelho pode ser exibido no prompt de segurança do navegador. Isso ocorre porque o certificado curinga SSL corresponde apenas a buckets sem períodos (.) em seus nomes quando HTTPS é usado para acesso a OBS. Recomendamos que você evite usar períodos (.) em nomes de buckets.

## Como usar essa função

Você pode usar console de OBS, as API, os SDK, OBS Browser+, ou obsutil para criar um bucket.

Ferramenta	Referência
Console de OBS	<a href="#">Criação de um bucket</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em linguagens de programação convencionais. Para obter detalhes, consulte o guia do desenvolvedor na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Criação de um bucket</a>
OBS Browser+	-
obsutil	<a href="#">Criação de um bucket</a>

## Como acessar um bucket

Depois que um bucket é criado, você pode usar o nome de domínio para acessar o bucket. Você pode montar o nome de domínio colocando o nome do bucket e o ponto final juntos, ou você pode obtê-lo [visualizando as informações básicas do bucket](#) no console de OBS ou em OBS Browser+.

Um nome de domínio de acesso é estruturado da seguinte forma:

[Estrutura] *BucketName.Endpoint*

[Estrutura] `bucketname.obs.cn-north-4.myhuaweicloud.com`

## Causas de falhas de criação de bucket e soluções

Para obter detalhes, consulte [Falha ao criar um bucket](#)

## 7.3 Listagem de buckets

### Cenário de aplicação

Você pode listar todos os buckets criados para exibir suas informações.

### Como usar essa função

Você pode usar console de OBS, as API, os SDK, OBS Browser+, ou obsutil para listar buckets.

Ferramenta	Referência
Console de OBS	Efetue login no console de OBS e selecione <b>Object Storage</b> . Em seguida, todos os buckets sob sua conta serão exibidos.
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em linguagens de programação convencionais. Para obter detalhes, consulte o guia do desenvolvedor na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Listagem de buckets</a>
OBS Browser+	Faça login em OBS Browser+. Todos os buckets sob sua conta são exibidos na lista de buckets.
obsutil	<a href="#">Listagem de buckets</a>

## 7.4 Visualização de informações de bucket

### Cenário de aplicação

Depois de criar um bucket no console de OBS, você pode visualizar as informações do bucket, incluindo suas estatísticas e detalhes básicos.

### Como usar essa função

Você pode usar console de OBS, as API, os SDK, OBS Browser+, ou obsutil para exibir informações básicas do bucket.

Ferramenta	Referência
Console de OBS	<a href="#">Visualização de informações de um bucket</a>

Ferramenta	Referência
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em linguagens de programação convencionais. Para obter detalhes, consulte o guia do desenvolvedor na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Obtenção de metadados de bucket</a>
OBS Browser+	-
obsutil	<a href="#">Consultação de propriedades de bucket</a>

## 7.5 Gerenciamento de cotas de bucket

### Cenário de aplicação

Por padrão, não todo o sistema de OBS nem qualquer bucket único tem limitações no tamanho total ou na quantidade de objetos que podem ser armazenados. Você pode definir uma cota para um bucket para limitar o tamanho total dos objetos que podem ser carregados no bucket. Se o tamanho total do objeto atingir o limite superior, os carregamentos de objetos falharão.

Uma cota de bucket pode controlar o carregamento de objetos somente depois que a cota é definida. Se a cota do bucket for menor que a capacidade dos objetos carregados, os objetos existentes não serão excluídos, mas os novos objetos não poderão ser carregados. Nesse caso, você pode carregar novos objetos somente depois de excluir alguns objetos existentes até que a capacidade de armazenamento usada seja menor que o limite da cota.

#### NOTA

- Uma cota de bucket deve ser um inteiro não negativo, em bytes. O valor máximo é  $2^{63} - 1$ .
- OBS não fornece uma API para excluir cotas de bucket. Você pode definir a cota do bucket como 0 para cancelar o limite da cota.

### Como usar essa função

Você pode usar as API e os SDK para gerenciar cotas de bucket.

Ferramenta	Referência
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em linguagens de programação convencionais. Para obter detalhes, consulte o guia do desenvolvedor na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Configuração da cota de armazenamento de bucket</a>

## 7.6 Exclusão de buckets

### Cenário de aplicação

Você pode excluir buckets indesejados para liberar a cota de buckets. Certifique-se de que um balde seja esvaziado antes de excluí-lo.

Um balde vazio deve atender aos seguintes requisitos:

- Não há nenhum objeto ou qualquer versão histórica de um objeto no bucket.
- Não há tarefa de carregamento de várias partes não mesclada no bucket, ou seja, não há fragmento no bucket.

#### NOTA

- Se o controle de versão estiver ativado para o bucket, certifique-se de que todas as versões históricas e versões com o **Delete Marker** (que também são consideradas versões históricas) tenham sido excluídas.
- O nome de um bucket excluído pode ser reutilizado pelo menos 30 minutos após a exclusão.

### Como usar essa função

Você pode usar console de OBS, as API, os SDK, OBS Browser+, ou obsutil para excluir buckets.

Ferramenta	Referência
Console de OBS	<a href="#">Exclusão de um bucket</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em linguagens de programação convencionais. Para obter detalhes, consulte o guia do desenvolvedor na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Exclusão de buckets</a>
OBS Browser+	-
obsutil	<a href="#">Exclusão de um bucket</a>

### Causas de falhas e soluções de exclusão de bucket

Para obter detalhes, consulte [Falha ao excluir um bucket](#).

# 8 Gerenciamento de objetos

---

## 8.1 Visão geral do objeto

Um objeto é a unidade básica de armazenamento de dados no OBS. Ele consiste em dados e metadados de objeto que descrevem os atributos desse objeto. Os dados carregados no OBS são armazenados como objetos em **buckets**.

Um objeto consiste em dados, metadados e uma chave.

- Uma chave especifica o nome de um objeto. Uma chave de objeto é uma cadeia UTF-8 que está entre 0 e 1024 caracteres. Cada objeto é identificado exclusivamente por uma chave em um bucket.
- Metadados: especificam a descrição do objeto. Metadados incluem metadados do sistema e metadados personalizados. Metadados são carregados no OBS como pares de chave/valor.
  - Metadados do sistema são gerados automaticamente pelo OBS e usados para processar dados de objetos, como Data, Comprimento do conteúdo, Última modificação e Conteúdo-MD5.
  - Você pode especificar metadados personalizados para descrever o objeto ao carregar o objeto em OBS.
- Dados: refere-se ao conteúdo que o objeto contém.

Geralmente, objetos são gerenciados como arquivos. No entanto, OBS é um serviço de armazenamento baseado em objetos e não há conceito de arquivos e pastas. Para facilitar o gerenciamento de dados, OBS fornece um método para simular pastas. Ao adicionar uma barra (/) em um nome de objeto, por exemplo, **test/123.jpg** você pode simular **test** como uma pasta e **123.jpg** como o nome de um arquivo na pasta **test**. No entanto, a chave do objeto permanece **test/123.jpg**.

Ao carregar um objeto, você pode definir uma **classe de armazenamento** para o objeto. Se nenhuma classe de armazenamento for especificada, a classe de armazenamento do objeto permanecerá igual à do bucket. Você também pode alterar a classe de armazenamento de um objeto existente em um bucket.

Você pode **Carregamento de um objeto** em **Como usar essa função** com base em seus hábitos e cenários de serviço. Em seguida, o OBS armazena os arquivos como objetos no bucket. No OBS, os buckets estão localizados em **diferentes regiões**. Não importa o método usado, você pode acessar o mesmo bucket e seus recursos na mesma região.

## Diretrizes sobre nomear chaves de objeto

Embora quaisquer caracteres UTF-8 possam ser usados em um nome de chave de objeto, nomear chaves de objeto de acordo com as diretrizes a seguir pode ajudar a maximizar a compatibilidade das chaves de objeto com outros aplicativos. As maneiras de analisar caracteres especiais variam de acordo com os aplicativos. As diretrizes a seguir ajudam os nomes de chave de objeto a atender substancialmente aos requisitos de DNS, caracteres de segurança da Web, analisadores XML e outras API.

Os seguintes conjuntos de caracteres podem ser usados livremente em nomes de chaves.

Caracteres alfanuméricos (também conhecidos como caracteres não reservados)	[0-9a-zA-Z]
Caracteres especiais (também conhecidos como caracteres reservados)	Ponto de exclamação (!) Hífen (-) Sublinhado (_) Período (.) Asterisco (*) Aspas simples (') Colchete esquerdo "(" Suporte direito ")"

Seguem-se exemplos de nomes de chave de objetos válidos:

```
4my-organization  
my.great_photos-2014/jan/myvacation.jpg  
videos/2014/birthday/video1.wmv
```

## Porcentagem-codificação de caracteres reservados

Se um caractere reservado tem um significado especial (conhecido como propósito reservado) em um URI e o caractere deve ser usado para outros fins no URI, esse caractere deve ser codificado por porcentagem. Use UTF-8 para codificar caracteres não ASCII. Caso contrário, os nomes dos objetos carregados no OBS podem ser diferentes do esperado. Por exemplo, se o caractere reservado "/" for usado como delimitador de componentes de caminho em um URI, o caractere terá um significado especial (separar um nome de bucket de um nome de objeto). Se "/" for usado em um componente do caminho em um URI, use três caracteres "%2F" ou "%2f" para substituir "/".

## Caracteres que podem exigir processamento especial

Caracteres que exigem codificação em um nome de chave

- Ampersand (&)
- Sinal do dólar (\$)
- Ponto-e-vírgula (;)
- Cólón (:)

- Mais sinal (+): O OBS decodifica os sinais de adição (+) em um URI de solicitação em espaços. Se um nome de chave de objeto original contém sinais de adição (+), ele deve ser codificado em %2B antes de ser colocado no URI da solicitação.
- Espaço: Um grande número de espaços consecutivos pode ser perdido em alguns casos.
- Sinal da igualdade (=)
- No sinal (@)
- Vírgula (,)
- Ponto de interrogação (?)
- Caracteres ASCII: 00-1F na forma hexadecimal (0-31 na forma decimal) e 7F (127 na forma decimal)

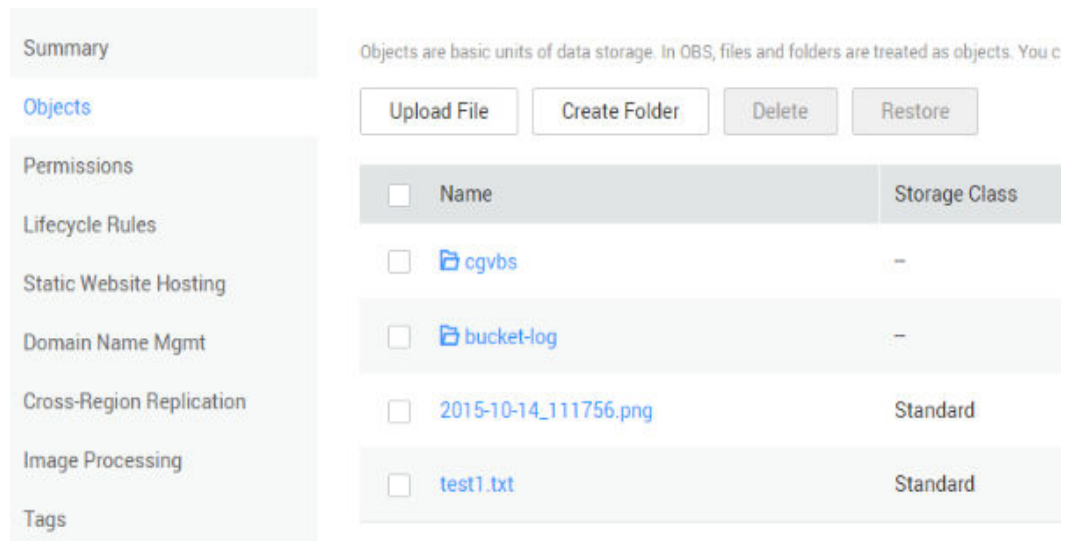
Evite usar os seguintes caracteres em nomes de chave, pois esses caracteres exigem muito processamento especial para manter a consistência em todos os aplicativos.

- Barra invertida (\)
- Braçadeira esquerda ({)
- Caracteres ASCII não imprimíveis (128-255 caracteres decimais)
- Inserir símbolo (^)
- Braçadeira direita (})
- Caractere percentual (%)
- Acentuação / Desmarcar (')
- Colchete direito (])
- Aspa
- Maior do que o sinal (>)
- Colchete esquerdo ([)
- Tilde (~)
- Menos do que o sinal (<)
- Hashtag (#)
- Barra vertical (|)

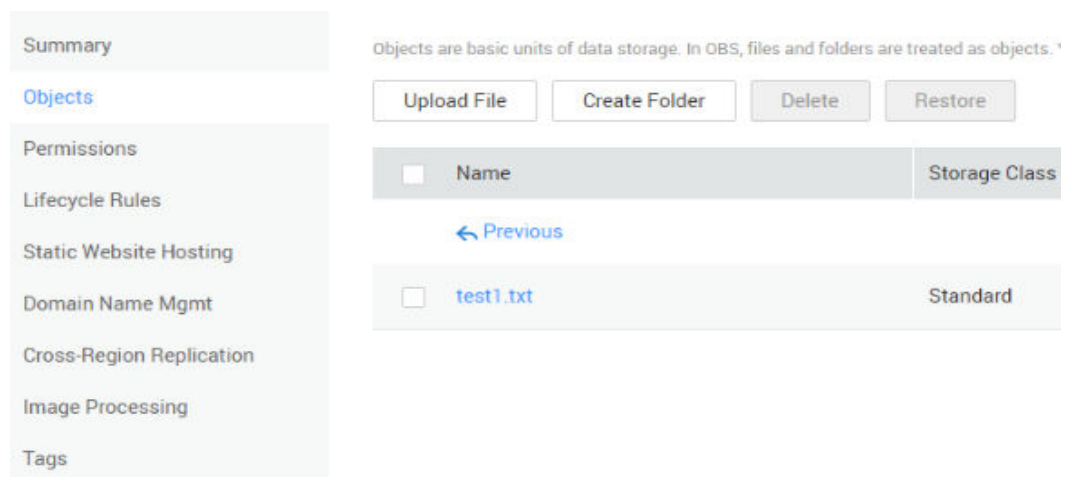
Observe que OBS adota uma estrutura plana, na qual você cria buckets e armazena objetos em buckets. Não há sub-buckets ou subpastas na estrutura. No entanto, você pode usar prefixos de nome de chave e delimitadores para deduzir a estrutura lógica como console de OBS faz. O conceito de pasta está disponível no console de OBS. Suponha que o bucket (**companybucket**) contenha quatro objetos com as seguintes chaves de objeto:

- bucket-log/log01.txt
- cgvbs/test1.txt
- 2015-10-14\_111756.png
- test1.txt

O console do OBS usa o prefixo do nome da chave (**bucket-log/** e **cgvbs/**) e o separador (/) para exibir a estrutura de pastas, como mostra a figura a seguir.



As chaves **2015-10-14\_111756.png** e **key test1.txt** não têm prefixos. Portanto, os objetos aparecem no nível raiz do bucket. Se você abrir a pasta **cgvbs/**, você verá que ela contém o objeto **test1.txt**.



Suponha que seu bucket **companybucket**) contenha dois objetos com as seguintes chaves de objeto:

- obj
- 1/../obj

Se você chamar uma API (por exemplo, uso do SDK) para obter os objetos, poderá obtê-los com êxito. Se você acessar os objetos do console, **obj** do objeto será retornado com base no relacionamento relativo. Porque **../** tem uma semântica especial em um URI, evite usá-lo em chaves de objeto.

## 8.2 Criação de uma pasta

### Cenário de aplicação

Você pode criar uma pasta em um bucket para facilitar a classificação e o gerenciamento dos dados armazenados no OBS.



Ao contrário de um sistema de arquivos, OBS não envolve os conceitos de arquivo e pasta. Para facilitar o gerenciamento de dados, OBS fornece um método para simular pastas. No OBS, um objeto é simulado como uma pasta adicionando uma barra (/) ao final do nome do objeto no console de OBS. Quando objetos são listados por chamar a API, o nome do objeto obtido é o caminho do objeto e o conteúdo após a última barra (/) é o nome real do objeto. Se o caminho terminar com uma barra (/), isso indica que o objeto é uma pasta. A profundidade hierárquica do objeto não afeta o desempenho de acesso ao objeto.

## Como usar essa função

Você pode usar console de OBS, as API, os SDK, OBS Browser+, ou obsutil para criar uma pasta.

Ferramenta	Referência
Console de OBS	<a href="#">Criação de uma pasta</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> . <b>NOTA</b> Criar uma pasta no OBS é criar um objeto cujo tamanho é 0 e cujo nome termina com uma barra (/), no essencial.
As API	- <b>NOTA</b> Criar uma pasta no OBS é criar um objeto cujo tamanho é 0 e cujo nome termina com uma barra (/), no essencial.
OBS Browser+	-
obsutil	<a href="#">Criação de uma pasta</a>

## 8.3 Carregamento de um objeto

### Cenário de aplicação

Você pode carregar arquivos ou pastas para um bucket de OBS existente. Se você quiser classificar os arquivos a serem carregados, crie uma pasta e carregue dos arquivos relacionados para a pasta.

Para obter detalhes sobre como criar uma pasta, consulte [Criação de uma pasta](#).

Esses arquivos podem ser textos, imagens, vídeos ou qualquer outro tipo de arquivo.

### Restrições

- Console de OBS tem restrições sobre o tamanho e o número de arquivos carregados.
  - Em regiões onde o carregamento em lote é suportado, um máximo de 100 arquivos podem ser carregados por vez, com um tamanho total não superior a 5 GB. Se você carregar apenas um arquivo no modo de carregamento em lote, esse arquivo não poderá exceder 5 GB de tamanho.

- Em regiões onde o carregamento em lote não é suportado, apenas um ficheiro pode ser carregado de cada vez, com um tamanho não superior a 50 MB.
- Você pode usar OBS Browser+ e obsutil para carregar arquivos com um tamanho total menor que 48,8 TB. OBS Browser+ permite que você faça o carregamento de um máximo de 500 arquivos por vez. Não há limite para o número de arquivos que você pode carregar usando obsutil por vez.
- Usando SDK/API PUT, POST ou operações anexáveis, você pode carregar arquivos com um tamanho total menor que 5 GB.
- Usando o carregamento de várias partes de SDK/API ou o carregamento retomável de SDK, você pode fazer carregamento de arquivos com um tamanho total menor que 48,8 TB.
- A função de carregamento em lote está disponível somente quando as seguintes condições são atendidas:
  - a. A região em que o bucket reside suporta carregamento em lote. Atualmente, as seguintes regiões suportam carregamento em lote: CN-Hong Kong, AP-Bangkok e AP-Singapura.
  - b. A versão do bucket deve ser **3.0**. Para obter detalhes sobre como exibir a versão do bucket, consulte [Verificação da versão do OBS \(OBS 2.0 ou OBS 3.0\)](#).
- Você pode carregar pastas diretamente por OBS Console, OBS Browser+, ou obsutil.
- Você pode definir uma cota de bucket para limitar a capacidade dos objetos a serem carregados em um bucket. Para mais detalhes, consulte [Gerenciamento de cotas de bucket](#).
- Se o controle de versão estiver desativado e o nome de um arquivo recém-carregado for o mesmo de um arquivo no bucket, o arquivo recém-carregado substituirá automaticamente o arquivo existente e não reterá as informações ACL do arquivo existente. Se o nome da pasta recém-carregada for o mesmo de uma pasta no bucket, as duas pastas serão mescladas e os arquivos na nova pasta substituirão os arquivos homônimos na pasta antiga.
- Se o controle de versão estiver ativado e o nome de um arquivo recém-carregado for o mesmo de um arquivo no bucket, uma nova versão será adicionada ao arquivo existente. Para obter detalhes sobre o controle de versão, consulte [Versionamento](#).

## Métodos de carregamento

Método de carregamento	Descrição
PUT (Transmissão de carregamento)	Use o método PUT ou POST quando o tamanho do arquivo a ser carregado for menor que 5 GB. Para obter detalhes sobre as diferenças entre os dois métodos de carregamento, consulte <a href="#">Quais são as diferenças entre os métodos de carregamento PUT e POST?</a>
POST (carregamento baseado em navegador)	

Método de carregamento	Descrição
Carregamento multiparte	Use esse método quando o tamanho do arquivo a ser carregado for maior que 5 GB e menor que 48,8 TB. Para mais detalhes, consulte <a href="#">Carregamento de multiparte</a> . <b>NOTA</b> Se você precisar carregar uma grande quantidade de dados (maior que 48,8 TB) para OBS, consulte <a href="#">Migração de dados locais para OBS</a> .
Transferência retomável	O carregamento de arquivos grandes geralmente falha devido a más condições de rede ou falhas no programa. É um desperdício de recursos reiniciar o processo de carregamento após uma falha de carregamento, e o processo de carregamento reiniciado ainda pode sofrer com a rede instável. Para resolver esses problemas, você pode usar a API para carregamento retomável, cujo princípio de funcionamento é dividir o arquivo a ser carregado em várias partes e carregá-las separadamente. Esse método economiza recursos e melhora a eficiência no re-carregamento, além de acelerar o processo de carregamento ao carregar simultaneamente partes. Para obter detalhes, consulte <a href="#">OBS suporta transferência de dados retomáveis?</a>
carregamento síncrono de objetos incrementais	Esse método sincroniza todo o conteúdo no caminho de origem local com o bucket de destino especificado em OBS, garantindo que o conteúdo seja consistente entre o caminho local e o bucket de destino. Sincronização incremental tem os seguintes significados: 1) Incremento: Compare o arquivo de origem com o objeto de destino e faça carregamento somente do arquivo de origem que tiver alterações. 2) Sincronização: Depois que o comando for executado, verifique se o caminho de origem local é um subconjunto do bucket de destino especificado por OBS. Ou seja, qualquer arquivo no caminho de origem local tem seu objeto correspondente no bucket de destino em OBS. Para obter detalhes, consulte <a href="#">Carregamento síncrono de objetos incrementais</a> .
Carregamento apensável	A operação AppendObject adiciona dados ao final de um objeto em um bucket especificado. Se não houver um objeto homônimo no bucket, um novo objeto será criado. Para obter detalhes, consulte <a href="#">Anexação de um objeto</a> .

## Como usar essa função

Você pode usar console de OBS, as API, os SDK, OBS Browser+, ou obsutil para carregar um objeto.

Ferramenta	Referência
Console de OBS	<a href="#">Carregamento de um arquivo</a>

Ferramenta	Referência
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Operações em objetos</a>
OBS Browser+	<a href="#">Carregar por arrastamento</a>
obsutil	<a href="#">Carregamento de um objeto</a>

## Operações relacionadas

Você pode especificar sua [classe de armazenamento](#) ao fazer carregamento de um objeto ou alterar sua classe de armazenamento após o carregamento.

- Os objetos podem ser alterados da classe de armazenamento Padrão para Acesso infrequente ou Arquivo, ou da classe de armazenamento Acesso infrequente to Padrão ou Arquivo, mas os objetos na classe de armazenamento Arquivo devem ser restaurados antes de serem alterados para a classe de armazenamento Padrão ou Acesso Infrequente. A transição de objetos na classe de armazenamento Acesso infrequente ou Arquivo incorre em taxas de recuperação. Selecione uma opção de transição apropriada com base em suas necessidades.
- Objetos na classe de armazenamento Acesso infrequente têm um período de armazenamento mínimo de 30 dias e aqueles na classe de armazenamento Arquivo têm um período de armazenamento mínimo de 90 dias. Se um objeto for transferido para outra classe de armazenamento antes de ser armazenado pelo período mínimo de armazenamento necessário, você precisará pagar pela duração total do armazenamento (30 dias ou 90 dias).
- Você também pode configurar uma regra de ciclo de vida para alterar a classe de armazenamento de um objeto. Para mais detalhes, consulte [Gerenciamento do ciclo de vida](#).

## Causas de falhas de carregamento de objetos e soluções

Para obter detalhes, consulte [Falha ao carregar um objeto](#).

# 8.4 Download de um objeto

## Cenário de aplicação

Você pode baixar objetos de OBS para um computador local.

OBS suporta download em lote de objetos. Para obter detalhes, consulte [OBS suporta download em lote?](#)

### NOTA

Objetos na classe de armazenamento Arquivo podem ser baixados somente quando estiverem no estado **Restored**.

## Como usar essa função

Você pode usar console de OBS, OBS Browser+, as API, os SDK, ou obsutil para baixar objetos e usar OBS Browser+ e obsutil para baixar pastas.

Ferramenta	Referência
Console de OBS	<a href="#">Download de um objeto</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Download de objetos</a>
OBS Browser+	-
obsutil	<a href="#">Download de um objeto</a>

## 8.5 Listagem de objetos

### Cenário de aplicação

Você pode exibir a lista de objetos criados listando objetos.

### Como usar essa função

Você pode usar console de OBS, OBS Browser+, as API, os SDK, ou obsutil para listar objetos.

Ferramenta	Referência
Console de OBS	Faça login no console de OBS. Selecione <b>Object Storage</b> , clique no nome de um intervalo e escolha <b>Objects</b> no painel de navegação esquerdo para exibir todos os objetos no intervalo.
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Listagem de objetos em um bucket</a>
OBS Browser+	-
obsutil	<a href="#">Listagem de objetos</a>

## 8.6 Cópia de objetos

### Cenário de aplicação

Você pode copiar objetos no OBS. Um objeto de até 5 GB pode ser copiado por cada operação. Para copiar um objeto com mais de 5 GB, use a API de carregamento de várias partes. Especificamente, esta operação de cópia permite:

- Crie uma cópia para um objeto.
- Renomeie um objeto criando uma cópia para ele e excluindo o objeto de origem.
- Editar os metadados do objeto. Cada objeto tem metadados, que é um conjunto de pares nome-valor. Você pode definir os metadados do objeto ao fazer o upload do objeto. Depois de carregar o objeto, você pode modificar seus metadados usando uma API. Para obter detalhes, consulte [Modificação de metadados do objeto](#). Você também pode criar uma cópia de objeto e definir os metadados. Na operação de cópia, defina o objeto de destino como sendo o mesmo que o objeto de origem.

Cada objeto contém metadados, incluindo metadados do sistema e metadados definidos pelo usuário. Você pode controlar alguns metadados do sistema. Quando você copia um objeto, os metadados do sistema controlados pelo usuário e os metadados definidos pelo usuário também são copiados. OBS redefine os metadados controlados pelo sistema. Por exemplo, quando você copia um objeto, o OBS redefine a data de criação do objeto copiado. Na solicitação de cópia, você não precisa definir esse valor.

Ao copiar um objeto, você pode querer atualizar alguns metadados. Por exemplo, se o objeto de origem estiver configurado para estar no armazenamento Padrão, talvez você queira alterar a classe de armazenamento da cópia do objeto para Acesso Infrequente. Você também pode modificar alguns metadados definidos pelo usuário do objeto de origem. Se você quiser modificar metadados, mesmo que apenas uma parte dos metadados, que podem ser configurados pelos usuários (definidos pelos usuários ou pelo sistema), especifique todos os metadados que podem ser configurados pelos usuários no objeto de origem na solicitação.

Se você quiser que os objetos sejam copiados automaticamente para outra região, crie uma regra de replicação entre regiões para que a recuperação de desastres fora do local possa ser garantida. Para mais detalhes, consulte [Replicação entre regiões](#).

#### NOTA

Quando o controle de versão estiver desativado para um bucket, se você fizer uma cópia do **objecta** e salvá-lo como **objectb**, e um objeto chamado **objectb** já existir, o novo **objectb** substituirá o existente. Depois que a cópia for executada com sucesso, somente o novo **objectb** poderá ser baixado porque o antigo **objectb** foi excluído. Portanto, antes de copiar um objeto, certifique-se de que não existe nenhum objeto com o mesmo nome que a cópia do objeto para impedir que os dados sejam excluídos por engano. Objeto de origem **objecta** não muda durante a cópia.

Não é possível determinar se uma solicitação foi executada com sucesso usando apenas **status\_code** no cabeçalho retornado pelo HTTP. Se 200 em **status\_code** for retornado, o servidor recebeu a solicitação e começa a processar a solicitação. O corpo na resposta mostra se a solicitação foi executada com sucesso. A solicitação é executada com sucesso somente quando o corpo contém ETag; caso contrário, a solicitação não será executada.

### Como usar essa função

Você pode usar as API, os SDK, OBS Browser+, ou obsutil para copiar objetos.

Ferramenta	Referência
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Cópia de objetos</a>
OBS Browser+	-
obsutil	<a href="#">Cópia de objetos</a>

## 8.7 Carregamento de multiparte

Carregamento de multiparte permite carregar um único objeto como um grupo de partes separadamente. Cada parte é uma parte de dados de objetos consecutivos. Você pode fazer carregamento de partes de objetos em qualquer sequência ou carregá-las de forma independente. Uma parte pode ser recarregada após uma falha de carregamento, sem afetar outras partes. Depois que todas as partes são carregadas, o OBS mescla essas partes para criar o objeto. Geralmente, se o tamanho de um objeto atingir 100 MB, o carregamento de multiparte é recomendado. Por exemplo, você deseja carregar um objeto (500 MB) em um bucket do OBS. Nesse caso, você pode usar a ferramenta OBS Browser+ para carregar o objeto em várias partes. A ferramenta divide automaticamente o objeto em várias partes para carregamento. Como alternativa, você pode fazer uma chamada de API para carregamento de multiparte, melhorando a eficiência do carregamento e reduzindo falhas.

Carregamento de multiparte oferece os seguintes benefícios:

- Melhoria da taxa de transferência: Você pode fazer carregamento de partes em paralelo para melhorar a taxa de transferência.
- Recuperação rápida de quaisquer falhas de rede: Peças de tamanho pequeno podem minimizar o impacto da falha no carregamento causada por erros de rede.
- Suspensão conveniente e retomada do carregamento de objetos: Você pode fazer o carregamento de partes a qualquer momento. Um carregamento de multiparte não tem um período de validade. Você deve concluir ou cancelar explicitamente o carregamento de multiparte.
- Iniciação do carregamento antes de saber o tamanho de um objeto: Você pode carregar um objeto ao criá-lo.

A API de carregamento de multiparte permite o carregamento de um objeto grande em várias partes. Você pode fazer carregamento de um novo objeto grande ou criar uma cópia de um objeto existente usando essa API.

O procedimento para carregar várias seções é o seguinte: Iniciação do carregamento (inicialização da tarefa de carregamento), carregamento de partes e conclusão do carregamento (mescla das partes carregadas). Ao receber uma solicitação de mescla de partes, o OBS mescla as partes carregadas para criar um novo objeto. O objeto pode ser acessado como outros objetos.

Você pode listar todas as tarefas de carregamento de multiparte em andamento ou obter a lista de partes carregadas de uma tarefa de carregamento de multiparte especificada. A seguir, descrevem-se as operações detalhadas.

## Inicialização de uma tarefa de carregamento de multiparte

Quando você envia uma solicitação para iniciar o carregamento de multiparte, o OBS retorna uma resposta com o ID de carregamento, que é o identificador exclusivo do carregamento de multiparte. Esse ID deve ser incluído na solicitação para fazer carregamento de partes, listar partes carregadas, concluir um carregamento de multiparte ou cancelar um carregamento de multiparte.

## Carregamento de uma parte

Ao carregar partes, você deve especificar o ID de carregamento e os números de parte. Você pode selecionar qualquer número de parte entre 1 e 10.000. Um número de parte identifica exclusivamente uma parte e sua localização no objeto que você está carregando. Se o número de uma parte carregada for usado para carregar uma nova parte, a parte carregada será sobrescrita. Sempre que você carrega uma parte, o OBS retorna o cabeçalho ETag na resposta. Para cada tarefa de carregamento de parte, é necessário registrar os números de parte e os valores de ETag. Esses números de parte e valores de ETag são necessários nas operações subsequentes de conclusão da tarefa de carregamento de multiparte.

---

### AVISO

Depois que a tarefa de carregamento de multiparte é inicializada e uma ou mais partes são carregadas, você deve mesclar as partes ou cancelar a tarefa de carregamento de multiparte. Caso contrário, você tem que pagar pela taxa de armazenamento das partes carregadas. O OBS libera o armazenamento e pára de cobrar a taxa de armazenamento somente depois que as partes carregadas são mescladas ou a tarefa de carregamento de multiparte é cancelada.

---

Quando várias operações de carregamento simultâneas são executadas para a mesma parte de um objeto, o servidor está em conformidade com a política Last Write Win, mas a hora referida em Last Write é a hora em que os metadados da parte são criados. Para garantir a precisão dos dados, o cliente deve ser bloqueado durante o carregamento simultâneo para a mesma parte de um objeto. O carregamento simultâneo de diferentes partes de um objeto não requer que o cliente seja bloqueado.

## Cópia de uma parte

Depois de criar um trabalho de carregamento de multiparte, você pode especificar os ID de carregamento e partes de carregamento para a tarefa de carregamento especificada. Você também pode chamar a API para cópia de partes para adicionar partes. Uma parte de um objeto ou todo o objeto pode ser copiado como uma parte.

---

### AVISO

Não é possível determinar se uma solicitação foi executada com sucesso usando apenas **status\_code** no cabeçalho retornado pelo HTTP. Se 200 em **status\_code** for retornado, o servidor recebeu a solicitação e começa a processar a solicitação. O corpo na resposta mostra se a solicitação foi executada com sucesso. A solicitação é executada com sucesso somente quando o corpo contém Etag. Caso contrário, o pedido não será executado.

---

Se você copiar o objeto de origem como uma parte chamada part1 e outra part1 já existir antes da operação de cópia, a part1 original será substituída pela nova após a operação de



cópia. Depois que a cópia for bem-sucedida, somente a nova parte 1 será exibida. Os dados da antiga parte 1 serão apagados. Portanto, assegure-se de que a parte de destino não exista ou não tenha valor ao copiar uma parte. Caso contrário, os dados podem ser apagados por engano. O objeto de origem não muda durante a cópia.

## Mescla de componentes e cancelamento de uma tarefa de carregamento de multiparte

Ao mesclar partes, o OBS cria um objeto ao padronizar múltiplas partes em ordem crescente. Se algum metadado de objeto for fornecido na inicialização de uma tarefa de carregamento de parte, o OBS associará os metadados ao objeto. Depois que o carregamento de multiparte for concluído, as partes não existirão mais. Uma solicitação de mescla de partes deve conter o ID de carregamento, números de parte e uma lista de valores ETag correspondentes. As respostas do OBS incluem o ETag que identifica exclusivamente os dados do objeto composto. O ETag não é o valor de hash MD5 dos dados do objeto. Você pode cancelar uma tarefa de carregamento de multiparte. Depois que uma tarefa de carregamento de multiparte é cancelada, o ID de carregamento não pode ser usado para carregar qualquer parte. Em seguida, o OBS libera o armazenamento de todas as partes carregadas. Se você parar um carregamento de multiparte em andamento, o carregamento ainda será concluído (o resultado pode ser bem-sucedido ou fracassado). Para liberar a capacidade de armazenamento ocupada por todos os componentes carregados, cancele o carregamento de multiparte após a conclusão de toda a tarefa.

---

### AVISO

Se 10 partes forem carregadas, mas apenas nove forem selecionadas para mescla, as partes que não forem mescladas serão automaticamente excluídas pelo sistema. As partes que não são mescladas não podem ser restauradas após serem excluídas. Antes de mesclar as partes, adote a API usada para listar as partes que foram carregadas para verificar todas as partes para garantir que nenhuma parte seja perdida.

---

## Listagem de partes carregadas

Você pode listar as partes de uma tarefa de carregamento de multiparte especificada ou as partes de todas as tarefas de carregamento de multiparte em andamento. As informações sobre as partes carregadas em um carregamento de multiparte específico serão retornadas para uma solicitação de lista de partes carregadas. Para cada solicitação para listar partes carregadas, o OBS retorna informações sobre partes carregadas no carregamento de multiparte específico. Informações sobre um máximo de 1000 partes podem ser devolvidas. Se mais de 1000 partes forem carregadas em um carregamento de multiparte, você precisará enviar várias solicitações para listar todas as partes carregadas. A lista de partes carregadas não inclui partes mescladas.

---

### AVISO

Uma lista devolvida só pode ser usada para verificação. Após a conclusão de um carregamento de multiparte, o resultado na lista não é mais válido. No entanto, quando os números de parte e os valores de ETag retornados pelo OBS são carregados, a lista de números de parte especificados pelo usuário será reservada.

---

## Listagem de tarefas de carregamento de multiparte

Você pode obter a lista de tarefas de carregamento de multiparte inicializadas listando as tarefas de carregamento de multiparte no bucket. Tarefas de carregamento de multiparte inicializadas referem-se às tarefas de carregamento de multiparte que não são mescladas ou canceladas após a inicialização. Um máximo de 1000 tarefas de carregamento de multiparte podem ser retornadas para cada solicitação. Se o número de tarefas de carregamento de multiparte em andamento exceder 1000, você precisará enviar mais solicitações para consultar as tarefas restantes.

**Tabela 8-1** lista as restrições na listagem de carregamentos de multiparte

**Tabela 8-1** Restrições na listagem de carregamentos de multiparte

Item	Restrição
Tamanho do objeto	Até 48,8 TB
Número máximo de partes para cada tarefa de carregamento	10.000
Número da parte	1 - 10.000 (incluído)
Tamanho da parte	O tamanho da parte está entre 5 MB e 5 GB. O tamanho da última parte é entre 0 bytes a 5 GB.
Número máximo de partes carregadas retornadas em resposta à solicitação de lista de partes carregadas.	1000
Número máximo de tarefas de carregamento de multiparte inicializadas retornadas em resposta à solicitação de lista de tarefas de várias partes inicializadas.	1000

## Operações e permissões de carregamento de multiparte

Você pode realizar o carregamento de multiparte somente após receber a permissão. Você pode usar ACL, políticas de bucket ou políticas de usuário para conceder aos usuários a permissão. A tabela a seguir lista as operações de carregamento de multiparte e as permissões necessárias que podem ser concedidas por ACL, políticas de bucket ou políticas de usuário.

Operação	Permissão necessária
Iniciação de uma tarefa de carregamento de multiparte	Para executar esta operação, você precisa ter a permissão PutObject. O proprietário de um bucket pode permitir que os fiduciários executem a operação PutObject.

Operação	Permissão necessária
Carregamento de partes	<p>Para executar esta operação, você precisa ter a permissão PutObject.</p> <p>Somente o iniciador de um carregamento de multiparte pode carregar partes. O proprietário do bucket deve conceder ao iniciador de carregamento de multiparte a permissão PutObject para que o iniciador possa carregar partes do objeto.</p>
Cópia de partes	<p>Para executar esta operação, você precisa ter a permissão PutObject, bem como a permissão GetObject no objeto a ser copiado.</p> <p>Somente o iniciador de um carregamento de multiparte pode copiar partes. O proprietário do bucket deve conceder ao iniciador de carregamento de multiparte a permissão PutObject para que o iniciador possa carregar partes do objeto.</p>
Mescla de partes	<p>Para executar esta operação, você precisa ter a permissão PutObject.</p> <p>Somente o iniciador de um carregamento de multiparte pode mesclar partes. O proprietário do bucket deve conceder ao iniciador de carregamento de multiparte a permissão PutObject para que o iniciador possa concluir o carregamento de multiparte.</p>
Cancelamento de uma tarefa de carregamento de multiparte	<p>Para executar esta operação, você precisa ter a permissão AbortMultipartUpload.</p> <p>Por padrão, somente o proprietário do bucket e o iniciador de carregamento de multiparte têm essa permissão. Além da configuração padrão, o proprietário do bucket pode permitir que os fiduciários executem essa operação. O proprietário do bucket também pode negar quaisquer fiduciários que estejam executando essa operação.</p>

Operação	Permissão necessária
Listagem de partes carregadas	<p>Para executar esta operação, você precisa ter a permissão ListMultipartUploadParts.</p> <p>Por padrão, o proprietário do bucket pode listar as partes carregadas de qualquer carregamento de multiparte no bucket. O iniciador de carregamento de multiparte pode listar as partes carregadas de um carregamento de multiparte específico.</p> <p>Além da configuração padrão, o proprietário do bucket pode permitir que os fiduciários executem essa operação. O proprietário do bucket também pode negar quaisquer fiduciários que estejam executando essa operação.</p>
Listagem de tarefas de carregamento de multiparte	<p>Para listar tarefas de carregamento de multiparte no bucket, você precisa ter a permissão ListBucketMultipartUploads.</p> <p>Além da configuração padrão, o proprietário do bucket pode permitir que os fiduciários executem essa operação.</p>

## As API REST aplicáveis ao carregamento de multiparte

As seções a seguir na *Referência de API de Object Storage Service* descrevem as API REST relevantes para o carregamento de multiparte.

- ListBucketMultipartUpload
- InitiateMultipartUpload
- UploadPart
- UploadPart-Copy
- ListParts
- CompleteMultipartUpload
- AbortMultipartUpload

## 8.8 Visualização de informações de objetos

### Cenário de aplicação

Depois de carregar um objeto, você pode exibir as informações ou propriedades do objeto.

### Como usar essa função

Você pode usar console de OBS, as API, os SDK, OBS Browser+, ou obsutil para exibir as informações de um objeto.

Ferramenta	Referência
Console de OBS	-
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Consulta de metadados de objetos</a>
OBS Browser+	-
obsutil	<a href="#">Consulta de propriedades de objetos</a>

## 8.9 Compartilhamento de um arquivo

### Cenário de aplicação

Você pode compartilhar arquivos com outros usuários temporária ou permanentemente.

- Compartilhamento temporário:

Compartilhar um arquivo com outros usuários por meio do URL temporário do arquivo. Todos os URL compartilhados são temporários com um período de validade.

Um URL temporário consiste no nome de domínio de acesso e nas informações de autenticação temporária de um arquivo. Exemplo:

```
https://bucketname.obs.cn-north-4.myhuaweicloud.com:443/image.png?  
AccessKeyId=xxx&Expires=xxx&response-content-disposition=xxx&x-obs-security-  
token=xxx&Signature=xxx
```

As informações de autenticação temporária contêm os parâmetros **AccessKeyId**, **Expires**, **x-obs-security-token**, e **Signature**. Os parâmetros **AccessKeyId**, **x-obs-security-token**, e **Signature** são usados para autenticação. O parâmetro **Expires** especifica o período de validade da autenticação.

Para obter detalhes sobre o método e os parâmetros de autenticação temporária, consulte [Autenticação de assinatura em um URL](#) em *Referência de API de Object Storage Service*.

Um URL temporário também contém o parâmetro **response-content-disposition**, que define se um objeto é diretamente baixado ou visualizado no navegador da web quando é acessado. O valor é determinado pelo navegador da web com base no **Content-Type** do objeto compartilhado.

- Partilha permanente:

Se você definir a permissão de um objeto para permitir que usuários anônimos o leiam, os usuários anônimos poderão acessar o objeto por meio da URL que você compartilhou. Para obter detalhes sobre como configurar permissões, consulte [Concessão de permissões de leitura pública em objetos para usuários anônimos](#).

O método de usar um navegador para acessar objetos varia dependendo do tipo de objeto. Você pode abrir arquivos **.txt** e **.html** diretamente usando um navegador. No entanto, quando você abre arquivos **.exe** e **.dat** usando um navegador, os arquivos são baixados automaticamente para o computador local.

Para obter detalhes sobre como obter o URL de acesso ao objeto, consulte [Como obtenho o caminho de acesso a um objeto?](#)

Se você não conseguiu acessar um objeto, corrija o problema de acordo com [Falha ao acessar um objeto por meio de seu URL](#).

## Restrições

- Objetos encriptados não podem ser compartilhados.
- You need to restore an Archive object before sharing it to other users.
- O código de autorização não é necessário para o compartilhamento temporário de arquivos, mas é necessário para o compartilhamento temporário de pastas. Para mais detalhes, consulte [Compartilhamento de uma pasta](#).
- Apenas os buckets da versão 3.0 ou posterior suportam o compartilhamento temporário de arquivos. Para obter detalhes sobre como consultar a versão do bucket, consulte [Verificação da versão do OBS \(OBS 2.0 ou OBS 3.0\)](#).
- A tabela a seguir descreve o período de validade da URL dos arquivos que são compartilhados temporariamente usando diferentes ferramentas.

Ferramenta	Período de validade do URL
Console de OBS	O URL é válido por 1 minuto a 18 horas. Depois que um objeto é compartilhado, o sistema irá gerar uma URL que contém as informações de autenticação temporária, válida por cinco minutos desde a sua geração por padrão. Cada vez que você altera o período de validade de um URL, o OBS obtém as informações de autenticação novamente para gerar um novo URL para compartilhamento. O novo URL torna-se válido no momento em que o período de validade é alterado.
Os SDK	Configure o parâmetro <b>Expires</b> para especificar quando uma autorização temporária expira. A autorização temporária expira em 24 horas.
As API	Configure o parâmetro <b>Expires</b> para especificar quando uma autorização temporária expira. A autorização temporária expira em 24 horas.
OBS Browser+	Quando você faz login no OBS Browser+ usando uma conta e senha, um URL compartilhado será válido por no máximo 24 horas. O período de validade padrão é de 10 horas. Se for necessário um período de validade mais longo, use o AK/SK permanente para fazer login.
obsutil	Use o parâmetro <b>e</b> para especificar quando um URL de download de objeto expira. O valor mínimo é 60s e o padrão é 300s.

## Como usar essa função

Você pode usar console de OBS, os SDK, as API, OBS Browser+, ou obsutil para compartilhar arquivos.

Ferramenta	Referência
Console de OBS	<a href="#">Compartilhamento de um arquivo</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Autenticação de assinatura em um URL</a>
OBS Browser+	-
obsutil	<a href="#">Geração do link de download de um objeto</a>

## Operações relacionadas

Para baixar um URL temporário gerada com wget, use "" e **-O** para especificar o nome do arquivo, ou um erro ocorrerá.

Exemplo:

```
[root@ecs-test ~]# wget "Temporary URL" -O abc
```

No comando anterior, *abc* indica um novo nome de arquivo.

## 8.10 Compartilhamento de uma pasta

### Cenário de aplicação

Você pode compartilhar pastas com outros usuários temporariamente ou permanentemente.

- Partilha temporária:

Compartilhar uma pasta com outros usuários por meio do URL temporário do arquivo. Todos os URL compartilhados são temporários com um período de validade.

Você precisa preparar um código de extração de seis dígitos antes de compartilhar uma pasta. Depois que o link de compartilhamento da pasta é criado, o OBS agrega automaticamente os links de download de todos os objetos na pasta a um site estático hospedado por um bucket público do OBS. Em seguida, os usuários que têm o URL temporário e o código de extração podem acessar o site estático e fazer o download de arquivos.

Um URL temporário consiste no nome de domínio de acesso e nas informações de autenticação temporária de uma pasta. Exemplo:

```
https://bucketname.obs.cn-north-4.myhuaweicloud.com:443/image.png?  
AccessKeyId=xxx&Expires=xxx&response-content-disposition=xxx&x-obs-security-  
token=xxx&Signature=xxx
```

As informações de autenticação temporária contêm os parâmetros **AccessKeyId**, **Expires**, **x-obs-security-token**, e **Signature**. Os parâmetros **AccessKeyId**, **x-obs-security-token**, e **Signature** são usados para autenticação. O parâmetro **Expires** especifica o período de validade da autenticação.

Para obter detalhes sobre o método e os parâmetros de autenticação temporária, consulte [Autenticação de assinatura em um URL](#) em *Referência de API de Object Storage Service*.

Um URL temporário também contém o parâmetro **response-content-disposition**, que define se um objeto é diretamente baixado ou visualizado no navegador da Web quando é acessado. O valor é determinado pelo navegador da Web com base no **Content-Type** do objeto compartilhado.

- **Compartilhamento permanente:**  
Se você definir a permissão de um objeto para permitir que usuários anônimos o leiam, os usuários anônimos poderão acessar o objeto por meio da URL que você compartilhou. Para obter detalhes sobre como configurar permissões, consulte [Concessão de permissões de leitura pública em objetos para usuários anônimos](#).  
Para obter detalhes sobre como obter o URL de acesso ao objeto, consulte [Como obtenho o caminho de acesso a um objeto?](#)  
Se você não conseguiu acessar um objeto, corrija o problema de acordo com [Falha ao acessar um objeto por meio de sua URL](#).

## Restrições

- You need to restore an Archive object before sharing it to other users.
- A função de compartilhamento de pastas é restrita a apenas algumas regiões.
- Somente os buckets da versão 3.0 ou posterior suportam o compartilhamento temporário de pastas. Para obter detalhes sobre como consultar a versão do bucket, consulte [Verificação da versão do OBS \(OBS 2.0 ou OBS 3.0\)](#).
- A tabela a seguir descreve o período de validade do URL das pastas que são compartilhadas temporariamente usando ferramentas diferentes.

Ferramenta	Período de validade do URL
Console de OBS	O URL é válido por 1 minuto a 18 horas. Depois que um objeto é compartilhado, o sistema irá gerar uma URL que contém as informações de autenticação temporária, válida por cinco minutos desde a sua geração por padrão. Cada vez que você altera o período de validade de um URL, OBS obtém as informações de autenticação novamente para gerar um novo URL para compartilhamento. O novo URL torna-se válido no momento em que o período de validade é alterado.
As API	Configure o parâmetro <b>Expires</b> para especificar quando uma autorização temporária expira. A autorização temporária expira em 24 horas.
OBS Browser+	Quando você faz login no OBS Browser+ usando uma conta e senha, um URL compartilhado será válido por no máximo 24 horas. O período de validade padrão é de 10 horas. Se for necessário um período de validade mais longo, use o AK/SK permanente para fazer login.
obsutil	Use o parâmetro adicional <b>vp</b> para especificar por quanto tempo um código de autorização é válido. O período de validade padrão é de um dia. As unidades permitidas incluem m (meses), w (semanas), d (dias), h (horas), min (minutos) e s (segundos). Se nenhuma unidade de tempo for especificada, o valor é calculado em segundos.



## Como usar essa função

Você pode usar Console do OBS, os SDK, as API, OBS Browser+, ou obsutil para compartilhar pastas.

Ferramenta	Referência
Console de OBS	<a href="#">Compartilhamento de uma pasta</a>
As API	<a href="#">Autenticação de assinatura em um URL</a>
OBS Browser+	-
obsutil	<a href="#">Criação de um código de autorização para compartilhamento de diretório</a>

## 8.11 Gerenciamento de metadados de objetos

### Cenário de aplicação

Metadados de objeto são um conjunto de pares nome-valor que descrevem o objeto e são usados para gerenciamento de objetos.

Atualmente, apenas os metadados definidos pelo sistema são suportados.

Os metadados definidos pelo sistema são classificados nos seguintes tipos: controlados pelo sistema e controlados pelo usuário. Por exemplo, metadados como **Last-Modified** são controlados pelo sistema e não podem ser modificados. Você pode modificar os metadados, como **ContentLanguage**, por meio da API. Os metadados que podem ser modificados são descritos a seguir:

Tabela 8-2 Metadados de OBS

Nome	Descrição
ContentDisposition	<p>Fornecer um nome de arquivo padrão para o objeto que está sendo solicitado. Quando um objeto está sendo baixado ou acessado, o arquivo com o nome de arquivo padrão é exibido diretamente no navegador ou uma caixa de diálogo de download é exibida se o arquivo estiver sendo acessado.</p> <p>Por exemplo, selecione <b>ContentDisposition</b> como o nome de metadados e digite <b>attachment;filename="testfile.xls"</b> como o valor de metadados de um objeto. Se você acessar o objeto por meio de um link, uma caixa de diálogo será exibida diretamente para o download de objetos e o nome do objeto será alterado para <b>testfile.xls</b>.</p> <p>Para obter detalhes, consulte a definição sobre ContentDisposition em HTTP.</p>

Nome	Descrição
ContentLanguage	Indica o idioma ou idiomas destinados ao público. Portanto, um usuário pode diferenciar de acordo com o idioma preferido do usuário. Para obter detalhes, consulte a definição sobre ContentLanguage em HTTP.
WebsiteRedirectLocation	<p>Redireciona um objeto para outro objeto ou para um URL externo. A função de redirecionamento é implementada usando hospedagem de sites estáticos.</p> <p>Por exemplo, você pode executar as seguintes operações para implementar o redirecionamento de objeto:</p> <ol style="list-style-type: none"> <li>1. Defina os metadados do objeto <b>testobject.html</b> no diretório raiz do bucket <b>testbucket</b>. Selecione <b>WebsiteRedirectLocation</b> para <b>Name</b> e insira <b>http://www.example.com</b> para <b>Value</b>.</li> </ol> <p><b>NOTA</b>                      OBS só suporta redirecionamento para objetos no diretório raiz de um bucket.</p> <ol style="list-style-type: none"> <li>2. Configure a hospedagem de site estático para o bucket <b>testbucket</b> e defina o objeto <b>testobject.html</b> no bucket como a home page padrão do site estático hospedado.</li> <li>3. Se você acessar o objeto <b>testobject.html</b> por meio do link de URL fornecido na página <b>Configure hospedagem de site estático</b>, a solicitação de acesso será redirecionada para <b>http://www.example.com</b>.</li> </ol>
ContentEncoding	<p>Formato de codificação de conteúdo quando um objeto é baixado. As opções são as seguintes:</p> <ul style="list-style-type: none"> <li>● Padrão: <b>compress</b>, <b>deflate</b>, <b>exi</b>, <b>identity</b>, <b>gzip</b>, e <b>pack200-gzip</b></li> <li>● Outros: <b>br</b>, <b>bzip2</b>, <b>lzma</b>, <b>peerdist</b>, <b>sdch</b>, <b>xpress</b>, <b>xz</b></li> </ul>
CacheControl	<p>Comportamento de cache da página da Web quando o objeto especificado é baixado.</p> <ul style="list-style-type: none"> <li>● Capacidade de cache: <b>public</b>, <b>private</b>, <b>no-cache</b>, e <b>only-if-cached</b></li> <li>● Tempo de expiração: <b>max-age=&lt;seconds&gt;</b>, <b>s-maxage=&lt;seconds&gt;</b>, <b>max-stale[=&lt;seconds&gt;]</b>, <b>min-fresh=&lt;seconds&gt;</b>, <b>stale-while-revalidate=&lt;seconds&gt;</b>, <b>stale-if-error=&lt;seconds&gt;</b></li> <li>● Re-verificação e recarregamento: <b>must-revalidate</b>, <b>proxy-revalidate</b>, <b>immutable</b></li> <li>● Outros: <b>no-store</b>, <b>no-transform</b></li> </ul>
Vencimento	Tempo de expiração do cache (GMT)
ContentType	<p>Tipo de arquivo de um objeto.</p> <p>Para mais detalhes, consulte <a href="#">Tipo de conteúdo de metadados do objeto</a>.</p>

 **NOTA**

- Quando o versionamento é ativado para um bucket, você pode definir metadados para objetos que são **Versão mais recente**, mas não pode definir metadados para objetos que são **Versão histórica**.
- Você não pode definir metadados de objeto para um objeto Arquivo.

## Tipo de conteúdo de metadados do objeto

Quando um objeto é carregado em OBS por meio do console de OBS ou de uma ferramenta, o sistema corresponde automaticamente o valor de **Content-Type** com base na extensão de nome de arquivo do objeto. Quando você acessa um objeto por meio de um navegador da Web, o sistema especifica um aplicativo para abrir o objeto de acordo com o valor de **Content-Type**. Você pode modificar o **Content-Type** de um objeto com base em sua extensão de nome de arquivo.

 **NOTA**

Se você carregar um objeto chamando uma API, especifique o valor de **Content-Type** porque o sistema não corresponde automaticamente ao valor de **Content-Type**.

**Tabela 8-3** Valores comuns do tipo de conteúdo

Extensão de nome de arquivo	Tipo de conteúdo	Extensão de nome de arquivo	Tipo de conteúdo
* (binary stream, which does not know the type of the file to be downloaded)	application/octet-stream	.tif	image/tiff
.001	application/x-001	.301	application/x-301
.323	text/h323	.906	application/x-906
.907	drawing/907	.a11	application/x-a11
.acp	audio/x-mei-aac	.ai	application/postscript
.aif	audio/aiff	.aifc	audio/aiff
.aiff	audio/aiff	.anv	application/x-anv
.asa	text/asa	.asf	video/x-ms-asf
.asp	text/asp	.asx	video/x-ms-asf
.au	audio/basic	.avi	video/avi
.awf	application/vnd.adobe.workflow	.biz	text/xml
.bmp	application/x-bmp	.bot	application/x-bot

Extensão de nome de arquivo	Tipo de conteúdo	Extensão de nome de arquivo	Tipo de conteúdo
.c4t	application/x-c4t	.c90	application/x-c90
.cal	application/x-cals	.cat	application/vnd.ms-pki.seccat
.cdf	application/x-netcdf	.cdr	application/x-cdr
.cel	application/x-cel	.cer	application/x-x509-ca-cert
.cg4	application/x-g4	.cgm	application/x-cgm
.cit	application/x-cit	.class	java/*
.cml	text/xml	.cmp	application/x-cmp
.cmx	application/x-cmx	.cot	application/x-cot
.crl	application/pkix-crl	.crt	application/x-x509-ca-cert
.csi	application/x-csi	.css	text/css
.cut	application/x-cut	.dbf	application/x-dbf
.dbm	application/x-dbm	.dbx	application/x-dbx
.dcd	text/xml	.dcx	application/x-dcx
.der	application/x-x509-ca-cert	.dgn	application/x-dgn
.dib	application/x-dib	.dll	application/x-msdownload
.doc	application/msword	.dot	application/msword
.drw	application/x-drw	.dtd	text/xml
.dwf	Model/vnd.dwf	.dwf	application/x-dwf
.dwg	application/x-dwg	.dxb	application/x-dxb
.dxf	application/x-dxf	.edn	application/vnd.adobe.edn
.emf	application/x-emf	.eml	message/rfc822
.ent	text/xml	.epi	application/x-epi
.eps	application/x-ps	.eps	application/postscript
.etd	application/x-ebx	.exe	application/x-msdownload
.fax	image/fax	.fdf	application/vnd.fdf

Extensão de nome de arquivo	Tipo de conteúdo	Extensão de nome de arquivo	Tipo de conteúdo
.fif	application/fractals	.fo	text/xml
.frm	application/x-frm	.g4	application/x-g4
.gbr	application/x-gbr	.	application/x-
.gif	image/gif	.gl2	application/x-gl2
.gp4	application/x-gp4	.hgl	application/x-hgl
.hmr	application/x-hmr	.hpg	application/x-hpgl
.hpl	application/x-hpl	.hqx	application/mac-binhex40
.hrf	application/x-hrf	.hta	application/hta
.htc	text/x-component	.htm	text/html
.html	text/html	.htt	text/webviewhtml
.htx	text/html	.icb	application/x-icb
.ico	image/x-icon	.ico	application/x-ico
.iff	application/x-iff	.ig4	application/x-g4
.igs	application/x-igs	.iii	application/x-iphone
.img	application/x-img	.ins	application/x-internet-signup
.isp	application/x-internet-signup	.IVF	video/x-ivf
.java	java/*	.jfif	image/jpeg
.jpe	image/jpeg	.jpe	application/x-jpe
.jpeg	image/jpeg	.jpg	image/jpeg
.jpg	application/x-jpg	.js	application/x-javascript
.jsp	text/html	.la1	audio/x-liquid-file
.lar	application/x-laplayer-reg	.latex	application/x-latex
.lavs	audio/x-liquid-secure	.lbm	application/x-lbm
.lmsff	audio/x-la-lms	.ls	application/x-javascript
.ltr	application/x-ltr	.mlv	video/x-mpeg

Extensão de nome de arquivo	Tipo de conteúdo	Extensão de nome de arquivo	Tipo de conteúdo
.m2v	video/x-mpeg	.m3u	audio/mpegurl
.m4e	video/mpeg4	.mac	application/x-mac
.man	application/x-troff-man	.math	text/xml
.mdb	application/msaccess	.mdb	application/x-mdb
.mfp	application/x-shockwave-flash	.mht	message/rfc822
.mhtml	message/rfc822	.mi	application/x-mi
.mid	audio/mid	.midi	audio/mid
.mil	application/x-mil	.mml	text/xml
.mnd	audio/x-musicnet-download	.mns	audio/x-musicnet-stream
.mocha	application/x-javascript	.movie	video/x-sgi-movie
.mp1	audio/mp1	.mp2	audio/mp2
.mp2v	video/mpeg	.mp3	audio/mp3
.mp4	video/mp4	.mpa	video/x-mpg
.mpd	application/vnd.ms-project	.mpe	video/x-mpeg
.mpeg	video/mpg	.mpg	video/mpg
.mpga	audio/rn-mpeg	.mpp	application/vnd.ms-project
.mps	video/x-mpeg	.mpt	application/vnd.ms-project
.mpv	video/mpg	.mpv2	video/mpeg
.mpw	application/vnd.ms-project	.mpx	application/vnd.ms-project
.mtx	text/xml	.mxx	application/x-mmxp
.net	image/pnetvue	.nrf	application/x-nrf
.nws	message/rfc822	.odc	text/x-ms-odc
.out	application/x-out	.p10	application/pkcs10

Extensão de nome de arquivo	Tipo de conteúdo	Extensão de nome de arquivo	Tipo de conteúdo
.p12	application/x-pkcs12	.p7b	application/x-pkcs7-certificates
.p7c	application/pkcs7-mime	.p7m	application/pkcs7-mime
.p7r	application/x-pkcs7-certreqresp	.p7s	application/pkcs7-signature
.pc5	application/x-pc5	.pci	application/x-pci
.pcl	application/x-pcl	.pcx	application/x-pcx
.pdf	application/pdf	.pdf	application/pdf
.pdx	application/vnd.adobe.pdx	.pfx	application/x-pkcs12
.pgl	application/x-pgl	.pic	application/x-pic
.pko	application/vnd.ms-pki.pko	.pl	application/x-perl
.plg	text/html	.pls	audio/scpls
.plt	application/x-plt	.png	image/png
.png	application/x-png	.pot	application/vnd.ms-powerpoint
.ppa	application/vnd.ms-powerpoint	.ppm	application/x-ppm
.pps	application/vnd.ms-powerpoint	.ppt	application/vnd.ms-powerpoint
.ppt	application/x-ppt	.pr	application/x-pr
.prf	application/pics-rules	.prn	application/x-prn
.prt	application/x-prt	.ps	application/x-ps
.ps	application/postscript	.ptn	application/x-ptn
.pwz	application/vnd.ms-powerpoint	.r3t	text/vnd.rn-realtex3d
.ra	audio/vnd.rn-realaudio	.ram	audio/x-pn-realaudio
.ras	application/x-ras	.rat	application/rat-file
.rdf	text/xml	.rec	application/vnd.rn-recording

Extensão de nome de arquivo	Tipo de conteúdo	Extensão de nome de arquivo	Tipo de conteúdo
.red	application/x-red	.rgb	application/x-rgb
.rjs	application/vnd.rn-realsystem-rjs	.rjt	application/vnd.rn-realsystem-rjt
.rlc	application/x-rlc	.rle	application/x-rle
.rm	application/vnd.rn-realmedia	.rmf	application/ vnd.adobe.rmf
.rmi	audio/mid	.rmj	application/vnd.rn-realsystem-rmj
.rmm	audio/x-pn-realaudio	.rmp	application/vnd.rn-rn_music_package
.rms	application/vnd.rn-realmedia-secure	.rmvb	application/vnd.rn-realmedia-vbr
.rmx	application/vnd.rn-realsystem-rmx	.rnx	application/vnd.rn-realplayer
.rp	image/vnd.rn-realpix	.rpm	audio/x-pn-realaudio-plugin
.rsml	application/vnd.rn-rsml	.rt	text/vnd.rn-realtxt
.rtf	application/msword	.rtf	application/x-rtf
.rv	video/vnd.rn-realvideo	.sam	application/x-sam
.sat	application/x-sat	.sdp	application/sdp
.sdw	application/x-sdw	.sit	application/x-stuffit
.slb	application/x-slb	.sld	application/x-sld
.slk	drawing/x-slk	.smi	application/smil
.smil	application/smil	.smk	application/x-smk
.snd	audio/basic	.sol	text/plain
.sor	text/plain	.spc	application/x-pkcs7-certificates
.spl	application/ futuresplash	.spp	text/xml
.ssm	application/ streamingmedia	.sst	application/vnd.ms-pki.certstore
.stl	application/vnd.ms-pki.stl	.stm	text/html



Extensão de nome de arquivo	Tipo de conteúdo	Extensão de nome de arquivo	Tipo de conteúdo
.sty	application/x-sty	.svg	text/svg+xml
.swf	application/x-shockwave-flash	.tdf	application/x-tdf
.tg4	application/x-tg4	.tga	application/x-tga
.tif	image/tiff	.tif	application/x-tif
.tiff	image/tiff	.tld	text/xml
.top	drawing/x-top	.torrent	application/x-bittorrent
.tsd	text/xml	.txt	text/plain
.uin	application/x-icq	.uls	text/iuls
.vcf	text/x-vcard	.vda	application/x-vda
.vdx	application/vnd.visio	.vml	text/xml
.vpg	application/x-vpeg005	.vsd	application/vnd.visio
.vsd	application/x-vsdx	.vss	application/vnd.visio
.vst	application/vnd.visio	.vst	application/x-vst
.vsw	application/vnd.visio	.vsx	application/vnd.visio
.vtx	application/vnd.visio	.vxml	text/xml
.wav	audio/wav	.wax	audio/x-ms-wax
.wb1	application/x-wb1	.wb2	application/x-wb2
.wb3	application/x-wb3	.wbmp	image/vnd.wap.wbmp
.wiz	application/msword	.wk3	application/x-wk3
.wk4	application/x-wk4	.wkq	application/x-wkq
.wks	application/x-wks	.wm	video/x-ms-wm
.wma	audio/x-ms-wma	.wmd	application/x-ms-wmd
.wmf	application/x-wmf	.wml	text/vnd.wap.wml
.wmv	video/x-ms-wmv	.wmx	video/x-ms-wmx
.wmz	application/x-ms-wmz	.wp6	application/x-wp6
.wpd	application/x-wpd	.wpg	application/x-wpg

Extensão de nome de arquivo	Tipo de conteúdo	Extensão de nome de arquivo	Tipo de conteúdo
.wpl	application/vnd.ms-wpl	.wq1	application/x-wq1
.wr1	application/x-wr1	.wri	application/x-wri
.wrk	application/x-wrk	.ws	application/x-ws
.ws2	application/x-ws	.wsc	text/scriptlet
.wsdl	text/xml	.wvx	video/x-ms-wvx
.xdp	application/vnd.adobe.xdp	.xdr	text/xml
.xfd	application/vnd.adobe.xfd	.xfdf	application/vnd.adobe.xfdf
.xhtml	text/html	.xls	application/vnd.ms-excel
.xls	application/x-xls	.xlw	application/x-xlw
.xml	text/xml	.xpl	audio/scpls
.xq	text/xml	.xql	text/xml
.xquery	text/xml	.xsd	text/xml
.xsl	text/xml	.xslt	text/xml
.xwd	application/x-xwd	.x_b	application/x-x_b
.sis	application/vnd.symbian.install	.sisx	application/vnd.symbian.install
.x_t	application/x-x_t	.ipa	application/vnd.iphone
.apk	application/vnd.android.package-archive	.xap	application/x-silverlight-app
.zip	application/zip	.rar	application/x-rar-compressed
.7z	application/x-7z-compressed	-	-

## Metadados de objeto definidos pelo usuário

Você pode adicionar os metadados definidos pelo usuário cujo nome começa com **x-obs-meta-** para facilitar o gerenciamento de objetos. Quando você recupera ou consulta os metadados do objeto, os metadados definidos pelo usuário adicionados serão retornados no cabeçalho da resposta. Os metadados definidos pelo usuário são limitados a 8 KB de tamanho.

Para medir o tamanho total dos metadados definidos pelo usuário, somar o número de bytes na codificação UTF-8 de cada chave e valor.

As chaves de metadados definidas pelo usuário não diferenciam maiúsculas de minúsculas, mas são armazenadas em minúsculas por OBS. Os valores-chave são sensíveis a maiúsculas e minúsculas.

O seguinte é um exemplo.

```
PUT /key HTTP/1.1
Host: bucket01.obs.myhuaweicloud.com
x-obs-meta-Test1: Test Metal

HEAD /Key HTTP/1.1
Host: bucket01.obs.myhuaweicloud.com
x-obs-meta-test1: Test Metal
```

As chaves de metadados definidas pelo usuário e seus valores devem estar em conformidade com os caracteres US-ASCII. Se caracteres não ASCII ou irreconhecíveis forem necessários, eles devem ser codificados ou decodificados em URL ou Base64 no lado do cliente, porque o lado do servidor não realiza nenhuma decodificação.

Se houver caracteres não-US-ASCII ou irreconhecíveis em um valor e o cliente não codificar o valor, o lado do servidor irá codificar o valor em Base64 e encapsular o valor usando **?UTF-8?B?<base64(str)>?=** Tome **x-obs-meta-nonascii: nonasciiÄÄ** como um exemplo. **nonasciiÄÄ** é codificado como **bm9uYXNjaWnDhMOE** em Base64 e a resposta retornada é **x-obs-meta-nonascii: =?UTF-8?B?bm9uYXNjaWnDhMOE?=**.

O seguinte é um exemplo.

```
PUT /key HTTP/1.1
Host: bucket01.obs.myhuaweicloud.com
x-obs-meta-nonascii: nonasciiÄÄ

HEAD /Key HTTP/1.1
Host: bucket01.obs.myhuaweicloud.com
x-obs-meta-nonascii: =?UTF-8?B?bm9uYXNjaWnDhMOE?=
```

## Como usar essa função

Você pode usar console de OBS, as API, os SDK, OBS Browser+, ou obsutil para personalizar metadados de objetos.

Ferramenta	Referência
Console de OBS	<a href="#">Configuração de metadados de objetos</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Modificação de metadados de objetos</a>
OBS Browser+	-
obsutil	<a href="#">Carregamento de um objeto</a>

## 8.12 Restauração de objetos a partir do armazenamento de arquivos

### Cenário de aplicação

Você precisa restaurar um objeto Arquivo antes de baixá-lo, acessá-lo usando sua URL ou definir permissões de ACL ou metadados de objeto para ele.

Para obter detalhes sobre a duração e os preços da restauração de dados, consulte [Detalhes de preços do produto](#).

### Restrições

- Se um objeto Arquivo estiver sendo restaurado, você não poderá suspender ou excluir a tarefa de restauração.
- Não é possível restaurar novamente um objeto que esteja no estado **Restoring**.
- Depois que um objeto é restaurado, uma cópia do objeto na classe de armazenamento Padrão será gerada. Dessa forma, há um objeto na classe de armazenamento Arquivo e também uma cópia na classe de armazenamento Padrão no bucket. Depois que um objeto Arquivo é restaurado, o status do objeto exibe **Restored**, e a cópia do objeto gerada na classe de armazenamento Padrão não é exibida no bucket.

Durante o período de validade da restauração, você será cobrado pelo espaço de armazenamento ocupado pelo objeto e sua cópia. A cópia será automaticamente excluída após a expiração de seu período de validade.

### Métodos de utilização desta função

Você pode usar console de OBS, as API, os SDK, OBS Browser+, ou obsutil para restaurar objetos do armazenamento Arquivo.

Ferramenta	Referência
Console de OBS	<a href="#">Restauração de um arquivo do armazenamento de Arquivo em OBS</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Restauração de objetos do armazenamento Arquivo</a>
OBS Browser+	-
obsutil	<a href="#">Restauração de objetos do armazenamento Arquivo de OBS</a>

## 8.13 Leitura direta

### Cenário de aplicação

Você pode habilitar a leitura direta se quiser obter objetos do Arquivo imediatamente ou se houver problemas de compatibilidade de interface quando OBS estiver interconectado com outros sistemas.

A leitura direta permite que objetos na classe de armazenamento Arquivo sejam baixados diretamente sem serem restaurados antecipadamente.

A leitura direta é uma função faturável. Para obter detalhes, consulte [Detalhes de preços do produto](#).

Você pode ativar a leitura direta ao criar um bucket ou ativá-lo para um bucket existente.

### Como usar essa função

Você pode usar console de OBS, as API, os SDK, ou OBS Browser+ para definir a leitura direta.

Ferramenta	Referência
Console de OBS	<a href="#">Leitura direta</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Definição da política de leitura direta para objetos de Arquivo em um bucket</a>
OBS Browser+	-

## 8.14 Exclusão de um objeto

### Cenário de aplicação

Você pode excluir arquivos ou pastas indesejados para economizar espaço de armazenamento e reduzir custos.

---

#### AVISO

Se o controle de versão não estiver ativado para um bucket, os objetos excluídos não poderão ser recuperados. Tenha cuidado ao realizar esta operação.

---

## Exclusão manual ou automática de objetos

Você pode excluir objetos manualmente a qualquer momento. Como alternativa, você pode usar [Regras do ciclo de vida](#) para excluir periodicamente objetos de um bucket ou limpar todos os objetos em um bucket de cada vez.

## Exclusão de um objeto ou de um lote de objetos

Você pode excluir um ou mais objetos por vez. Para obter detalhes sobre como excluir objetos em lotes, consulte [OBS suporta exclusão de objetos em lote?](#)

## Mecanismo de exclusão de objetos quando versionamento estiver ativado

Quando o controle de versão está habilitado, OBS usa métodos de exclusão diferentes para objetos diferentes.

- A exclusão de um arquivo ou pasta não o exclui permanentemente. O arquivo ou pasta excluído será mantido na lista **Deleted Objects** e marcado com o **Delete Marker**.
  - Se você quiser excluir o arquivo ou a pasta permanentemente, será necessário excluí-lo da lista de **Deleted Objects**.
  - Para recuperar um arquivo excluído, você pode cancelar a exclusão pela operação **Undelete**. Para mais detalhes, consulte [Cancelamento da exclusão de um objeto](#).
- A exclusão de uma versão de um objeto excluirá permanentemente essa versão. Se a versão excluída for a mais recente, a próxima versão mais recente se tornará a versão mais recente.

Quando o controle de versão está ativado, os arquivos na lista **Deleted Objects** também têm várias versões. Observe os seguintes pontos ao excluir versões diferentes de arquivos:

- Se você excluir uma versão com o **Delete Marker**, ele realmente recupera essa versão específica em vez de excluí-la permanentemente. Para mais detalhes, consulte [Cancelamento da exclusão de um objeto](#).
- Se você excluir uma versão sem o **Delete Marker**, essa versão específica será excluída permanentemente. Mesmo que o objeto seja recuperado mais tarde, esta versão não será recuperada.

Para obter mais informações, consulte [Mecanismo de recuperação de objetos quando versionamento estiver ativado](#).

## Como usar essa função

Você pode usar console de OBS, as API, os SDK, OBS Browser+, obsutil, ou obsfs para excluir objetos.

Ferramenta	Referência
Console de OBS	<a href="#">Exclusão de um arquivo ou uma pasta</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Exclusão de um objeto</a>

Ferramenta	Referência
OBS Browser+	-
Obsutil	<a href="#">Exclusão de um objeto</a>

## 8.15 Cancelamento da exclusão de um objeto

### Cenário de aplicação

Se o versionamento não estiver ativado para um bucket, os objetos excluídos não poderão ser recuperados. Se o versionamento estiver ativado, você poderá recuperar objetos excluídos por meio da operação **Undelete**. Para obter detalhes sobre o versionamento, consulte [Versionamento](#).

### Mecanismo de recuperação de objetos quando versionamento estiver ativado

Quando um bucket tem a função do versionamento ativada, excluir um arquivo da lista **Objects** não o exclui permanentemente. O arquivo excluído será mantido com o **Delete Marker** na lista **Deleted Objects**. Você pode recuperar um objeto excluído pela operação **Undelete**.

Ao executar a operação **Undelete**, observe os seguintes pontos:

1. Você só pode recuperar arquivos excluídos, mas não pastas.  
Depois de restaurar um arquivo excluído, o arquivo é recuperado e aparecerá na lista **Objects**. Em seguida, você pode executar operações básicas no arquivo como normalmente faz em outros objetos. Se o arquivo foi armazenado em uma pasta antes da exclusão, ele será recuperado para seu caminho original depois que você o recuperar.
2. Os arquivos excluídos nos **Deleted Objects** também têm várias versões. Ao excluir versões diferentes de arquivos, observe os seguintes pontos:
  - Se você excluir uma versão com o **Delete Marker**, ele realmente recupera essa versão específica em vez de excluí-la permanentemente.
  - Se você excluir uma versão sem o **Delete Marker**, essa versão específica será excluída permanentemente. Mesmo que o objeto seja recuperado mais tarde, esta versão não será recuperada.
3. Existe pelo menos uma versão sem o **Delete Marker** na lista **Deleted Objects**. Caso contrário, a exclusão não pode ser cancelada.

### Como usar essa função

Você pode usar console de OBS para desfazer a exclusão de objetos.

Ferramenta	Referência
Console de OBS	<a href="#">Cancelamento da exclusão de um arquivo</a>

## 8.16 Gerenciamento de fragmentos

### Cenário de aplicação

Fragmentos referem-se a dados incompletos em um bucket. Os dados são carregados em OBS no modo multiparte. Nos seguintes cenários comuns e outros, um carregamento falha e fragmentos são gerados. Você precisa limpar esses fragmentos para liberar espaço de armazenamento.

- A rede está em más condições, e a conexão com o servidor OBS é interrompida frequentemente.
- A tarefa de carregamento é suspensa manualmente.
- O dispositivo está com falha.
- O dispositivo é desligado repentinamente.

Fragmentos em OBS consomem espaço de armazenamento e são cobrados com base nas taxas de preço do espaço de armazenamento. Se os fragmentos forem gerados devido a interrupções ou falhas de tarefas de carregamento de várias partes, você poderá retomar essas tarefas para que os fragmentos sejam apagados ou limpar diretamente os fragmentos para economizar espaço de armazenamento.

Para obter detalhes sobre como gerenciar fragmentos, consulte [Como gerenciar fragmentos?](#)

#### NOTA

- Se um bucket não for excluído, verifique se todos os fragmentos foram excluídos. Se não, exclua todos os fragmentos do bucket.
- Se não existir nenhum objeto no bucket, mas a taxa ainda estiver sendo deduzida, verifique se há fragmentos no bucket. Se sim, exclua os fragmentos para reduzir os custos de armazenamento.



# 9 Sistema de arquivos paralelos

---

Parallel File System (PFS) é um sistema de arquivos de alto desempenho fornecido pelo OBS, com latência de acesso em milissegundos. O PFS suporta largura de banda de nível TB/s e milhões de IOPS, tornando-o ideal para o processamento de cargas de trabalho de computação de alto desempenho (HPC).

Você pode acessar dados em um sistema de arquivos paralelo usando as API padrão de OBS. Ele também suporta leitura e gravação de dados através de obsfs, um cliente de PFS que suporta POSIX. obsfs pode ser implantado em um ECS e, em seguida, você pode usar obsfs para montar um sistema de arquivos paralelo no SO de Linux em execução no ECS. O sistema de arquivos paralelo montado funciona como um sistema de arquivos local. Você pode gerenciar o sistema de arquivos paralelo montado online, incluindo a criação, exclusão, renomeação de arquivos e pastas, bem como a modificação de arquivos.

Para obter detalhes, consulte [Guia de recursos do sistema de arquivos paralelos](#).

# 10 Processamento de imagens

---

Integrado no OBS, o recurso de processamento de imagem fornece serviços de processamento de imagem estáveis, seguros, eficientes e de baixo custo. Ao usar esse recurso, você pode emagrecer (reduzir), cortar, compactar e imagens de marca d'água, bem como converter os formatos de imagens.

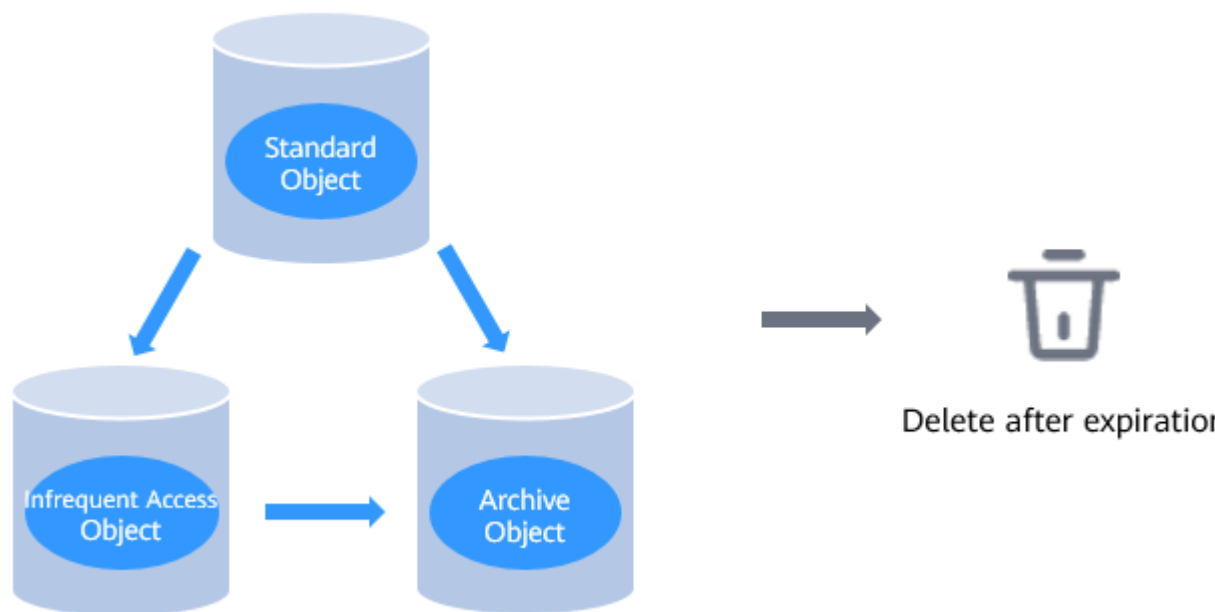
Esse recurso aproveita console de OBS e as API REST. Você pode processar imagens armazenadas no OBS de várias maneiras a qualquer hora e em qualquer lugar, e obter as imagens processadas imediatamente.

Para obter detalhes, consulte [Guia de recursos de processamento de imagens](#).

# 11 Gerenciamento do ciclo de vida

O gerenciamento do ciclo de vida do OBS inclui a exclusão periódica de objetos de buckets e a transição entre classes de armazenamento de objetos com base em regras predefinidas.

**Figura 11-1** Gerenciamento do ciclo de vida



## Cenário de aplicação

Você pode configurar regras de gerenciamento do ciclo de vida para executar o seguinte:

- Exclusão periódica dos arquivos que só devem ser retidos por períodos de tempo especificados
- Transição de documentos que raramente são acessados para a classe de armazenamento de Acesso Infrequente ou Arquivo ou excluí-los
- Armazenar vários tipos de dados no OBS para fins de Arquivo, como mídia digital, registros financeiros e médicos, dados de sequência do genoma original, backup de banco de dados de longo prazo e dados que devem ser mantidos para conformidade regulatória.

- Agendamento da exclusão de todos os arquivos de um bucket. A exclusão de objetos manualmente é demorada e somente um número limitado de objetos pode ser excluído por vez.

Você pode definir regras do ciclo de vida para identificar objetos e gerenciar ciclos de vida dos objetos com base nas regras.

Objetos que não são mais acessados com frequência podem ser transferidos para Acesso Infrequente ou Arquivo, reduzindo seus custos. Em suma, transição basicamente significa que a classe de armazenamento do objeto é alterada sem copiar o objeto. Você também pode alterar manualmente a classe de armazenamento de um objeto na página **Objects**. Para obter detalhes, consulte [Carregamento de um arquivo](#).

A configuração do ciclo de vida pode ser adicionada a um bucket com o o versionamento ativado ou desativado. Por padrão, o o versionamento é desativado. Você pode habilitá-lo. Se o versionamento estiver habilitado para um bucket, o bucket manterá a versão atual do objeto e zero ou mais versões não atuais do objeto. Você pode reduzir os custos de armazenamento de objetos usando regras de versionamento e ciclo de vida. As ações predefinidas de configuração do ciclo de vida podem facilitar o gerenciamento ao longo do ciclo de vida das versões atuais do objeto e das versões não atuais do objeto.

## Precauções

- Não há limite para o número de regras do ciclo de vida em um bucket, mas o tamanho total das descrições XML sobre todas as regras do ciclo de vida em um bucket não pode exceder 20 KB.
- A duração mínima do armazenamento do Arquivo é de 90 dias. Após a transição, se o período de armazenamento do objeto Arquivo for menor que a duração mínima de armazenamento, você precisará pagar pelos dias restantes.
- Restrições na transição da classe de armazenamento:
  - As regras do ciclo de vida só podem fazer a transição de objetos da classe de armazenamento Padrão para a classe de armazenamento Acesso Infrequente. Operações manuais são necessárias para fazer a transição de objetos da classe de armazenamento Acesso Infrequente para a classe de armazenamento Padrão.
  - Os objetos só podem ser transferidos da classe de armazenamento Padrão ou Acesso Infrequente para a classe de armazenamento Arquivo. Se quiser fazer a transição de objetos da classe de armazenamento Arquivo para a classe de armazenamento Padrão ou Acesso Infrequente, primeiro será necessário restaurar objetos da classe de armazenamento Arquivo e, em seguida, fazer a transição manual da classe de armazenamento.
- Exclusão de objeto na expiração:

O OBS fornece a ação **Expiration** que permite definir o tempo de expiração para objetos. Além disso, para buckets com o versionamento ativado, o OBS fornece a operação **NoncurrentVersionExpiration** que permite definir o tempo de expiração para versões de objetos não atuais.

  - Para buckets com o versionamento desativado, a ação **Expiration** permite excluir os objetos expirados permanentemente.
  - Se o versionamento estiver ativado (ou suspenso) para um bucket, a ação **Expiration** excluirá logicamente a versão atual de um objeto criando um marcador de exclusão para a versão atual e reconhecendo o marcador de exclusão como a nova versão atual. A ação **NoncurrentVersionExpiration** exclui objetos de versões não atuais permanentemente.

- Além dos buckets, você também pode configurar o tempo de expiração para objetos durante os uploads de objetos. O tempo de expiração dos objetos prevalece contra o dos buckets.

Depois que um objeto expira, o OBS adiciona o objeto à fila de exclusão e o exclui de forma assíncrona. Isso pode fazer com que o tempo de exclusão seja posterior ao tempo de expiração. Depois que um objeto expirar, você não será cobrado pela taxa de duração de armazenamento relacionada.

Para consultar o tempo de expiração planejado de um objeto, você pode chamar a API de objeto GET ou a API de objeto HEAD. Essas API retornam cabeçalhos de resposta que fornecem informações de expiração sobre o objeto.

- Depois que uma regra do ciclo de vida é modificada, a modificação não se aplica aos objetos que já atenderam às condições configuradas. Por exemplo, a regra do ciclo de vida original faz com que os objetos que atendem às condições configuradas expirem e, em seguida, sejam excluídos após serem armazenados por um dia. Sob essa regra, os objetos carregados em 1o de janeiro de 2021 serão excluídos em 3 de janeiro de 2021. Em 3 de janeiro de 2021, se você alterar a regra do ciclo de vida para excluir objetos depois que eles forem armazenados por sete dias, os objetos carregados em 1o de janeiro de 2021 ainda serão excluídos em 3 de janeiro de 2021, mas essa modificação se aplicará aos carregados em e após 2 de janeiro de 2021.

## Regras do ciclo de vida

As regras do ciclo de vida têm os seguintes elementos-chave:

- Política: Configure uma regra do ciclo de vida que tem efeito em objetos especificados.
  - Pelo prefixo: Você pode especificar um prefixo de nome de objeto, para que a regra do ciclo de vida entre em vigor em objetos que compartilham o mesmo prefixo.
  - O bucket inteiro: Você pode especificar um bucket inteiro, para que a regra do ciclo de vida tenha efeito em todos os objetos no bucket.
- Hora: uma hora agendada em que a classe de armazenamento de objetos é transferida. Você pode especificar o número de dias após os quais os objetos que foram atualizados pela última vez e que atendem às condições especificadas são automaticamente transferidos para Acesso pouco frequente ou Arquivo ou após os quais eles expiram e são excluídos automaticamente.
  - Transição para Acesso Infrequente: Você pode especificar o número de dias após os quais os objetos que foram atualizados pela última vez e atendem às condições especificadas são transferidos automaticamente para Acesso Infrequente.
  - Transição para arquivo: Você pode especificar o número de dias após os quais os objetos que foram atualizados pela última vez e atendem às condições especificadas são automaticamente transferidos para Arquivo.
  - Excluído na expiração: Você pode especificar o número de dias após os quais os objetos que foram atualizados pela última vez e atendem às condições especificadas são excluídos automaticamente.

O tempo da regra do ciclo de vida tem as seguintes restrições:

- Se apenas uma transição estiver envolvida, o tempo deve ser pelo menos um dia depois.
- Se várias transições estão envolvidas, a última transição deve ser pelo menos um dia mais longa do que a anterior transição.

Por exemplo, se você quiser que uma regra do ciclo de vida altere a classe de armazenamento de objetos e exclua objetos expirados, defina o tempo de transição para 23 dias depois e o tempo de exclusão para pelo menos 24 dias depois.

#### NOTA

O período mínimo de cobrança é de 30 dias para armazenamento de Acesso Infrequente e 90 dias para armazenamento de Arquivo. Se os objetos forem excluídos por uma regra do ciclo de vida antes de serem armazenados por esse período mínimo, você ainda será cobrado pelo período completo (30 ou 90 dias, dependendo do serviço).

Exemplo: Você carrega um objeto para a classe de armazenamento padrão e nenhuma outra operação é executada no objeto. O bucket, onde o objeto reside, tem uma regra do ciclo de vida que alterará a classe de armazenamento de objeto de Padrão para Acesso Infrequente três dias depois, de Acesso Infrequente para Arquivo e, finalmente, excluirá os objetos à medida que expiram sete dias depois. Nesse caso, você será cobrado por três dias de armazenamento Padrão, 30 dias de armazenamento de Acesso Infrequente e 90 dias de armazenamento de Arquivo.

## Elementos de configuração do ciclo de vida

Você pode definir o formato de configuração do ciclo de vida como XML. A configuração contém uma ou mais regras do ciclo de vida.

Cada regra consiste nos seguintes conteúdos:

- Metadados, especificando o ID da regra e se a regra está ativada ou desativada. Se a regra estiver desativada, o OBS não executará as ações especificadas na regra.
- Critérios de filtragem, identificação de objetos nos quais as regras do ciclo de vida são aplicadas. Você pode definir o prefixo do nome do objeto para ser o critério de filtragem.
- Quando (uma data ou um período de tempo) uma ação de transição ou expiração será executada em objetos no ciclo de vida.

Exemplos de configuração:

### Exemplo 1: Configuração do ciclo de vida de um bucket com versionamento desativado

Por padrão, o versionamento é desativado para buckets. Cada objeto tem apenas uma versão.

Suponha que o versionamento esteja desabilitado para o bucket e você queira que os objetos cujo nome comece com **documents/** transição da classe de armazenamento Padrão para a classe de armazenamento Acesso Infrequente 30 dias após o upload, em seguida, da classe de armazenamento Acesso Infrequente para a classe de armazenamento Arquivo 60 dias após o upload e, finalmente, ser excluído um ano após o upload. Para conseguir isso, você pode adicionar a seguinte configuração do ciclo de vida para o seu bucket. Essa regra do ciclo de vida inclui duas operações (**Transition** e **Expiration**). Essa regra se aplica a objetos cujo prefixo de chave é **documents** (especificado no elemento **Prefix**).

```
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>documents</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
    <Transition>
      <Days>30</Days>
      <StorageClass>WARM</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

```
<Days>60</Days>
  <StorageClass>COLD</StorageClass>
</Transition>
</Rule>
</LifecycleConfiguration>
```

Exemplo 2. Configuração do ciclo de vida de um bucket com o versionamento ativado

Você pode ativar o o versionamento para buckets. Se o o versionamento estiver ativado para um bucket, o bucket reterá a versão atual do objeto e suas versões não atuais do objeto. Para mais detalhes, consulte [Versionamento](#). O versionamento permite manter os registros de histórico de objetos e o gerenciamento do ciclo de vida permite controlar a retenção de versões de objetos, bem como a transição da classe de armazenamento.

Para um bucket com o versionamento ativado, a configuração do ciclo de vida fornece várias ações predefinidas que podem ser usadas para gerenciar versões de objetos não atuais. Neste exemplo, a configuração do ciclo de vida tem uma regra que especifica duas operações (**NoncurrentVersionTransition** e **NoncurrentVersionExpiration**) para objetos cujo prefixo de chave é **logs/**. A ação **NoncurrentVersionTransition** faz a transição de objetos para Acesso Infrequente e para Arquivar 30 dias e 60 dias, respectivamente, após se tornarem versões não atuais. A ação **NoncurrentVersionExpiration** exclui objetos permanentemente 180 dias após eles se tornarem versões não atuais.

```
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>logs</Prefix>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>180</NoncurrentDays>
    </NoncurrentVersionExpiration>
    <NoncurrentVersionTransition>
      <NoncurrentDays>30</NoncurrentDays>
      <StorageClass>WARM</StorageClass>
    </NoncurrentVersionTransition>
    <NoncurrentVersionTransition>
      <NoncurrentDays>60</NoncurrentDays>
      <StorageClass>COLD</StorageClass>
    </NoncurrentVersionTransition>
  </Rule>
</LifecycleConfiguration>
```

Você pode usar as ações predefinidas do **NoncurrentVersionTransition** e **NoncurrentVersionExpiration** para gerenciar versões não atuais em seu bucket.

Geralmente, cada regra do ciclo de vida consiste nas seguintes partes:

- Metadados: especificando o ID da regra (elemento **<ID>**) e se a regra está ativada ou desativada (elemento **<Status>**). Se a regra estiver desativada, o OBS não executará as ações especificadas na regra.
- Prefixo (elemento **<Prefix>**), que identifica objetos aos quais a regra se aplica.
- Ações que você deseja executar nos objetos especificados (como **NoncurrentVersionTransition** e **NoncurrentVersionExpiration** mencionados no exemplo anterior). Cada ação inclui a data ou o período de tempo após o qual a ação será executada.

Os elementos na regra de configuração do ciclo de vida são descritos a seguir:

- Elemento **ID**  
Uma configuração do ciclo de vida pode ter um máximo de 1000 regras. Cada regra tem um ID exclusivo.

- Elemento **Status**

O valor pode ser **Enabled** ou **Disabled**. Se uma regra estiver desativada, o OBS não executará nenhuma ação definida na regra.

- Elemento **Prefix**

Uma regra do ciclo de vida se aplica a um ou mais objetos com o prefixo de nome especificado na regra. Suponha que você tenha os seguintes objetos:

- logs/day1
- logs/day2
- logs/day3
- ExampleObject.jpg

Se você definir **Prefix** como **ExampleObject.jpg**, a regra se aplicará somente ao objeto **ExampleObject.jpg**. Se você definir **Prefix** como **logs/**, a regra se aplicará aos objetos cujo nome comece com **logs/**. Se você deixar **Prefix** null, a regra se aplicará a todos os objetos do bucket.

- Elemento **Action**

Você pode especificar ações predefinidas na regra do ciclo de vida para executá-las em seus buckets no ciclo de vida. As ações predefinidas incluem **Transition**, **Expiration**, **NoncurrentVersionTransition**, e **NoncurrentVersionExpiration**. O efeito da ação depende do status do versionamento de um bucket.

 **NOTA**

Por padrão, o versionamento é desativado para buckets. Você pode ativar o o versionamento do bucket, para que cada objeto tenha uma versão atual e possa ter uma ou mais versões não atuais. Além disso, você pode desativar o o versionamento. Para mais detalhes, consulte [Versionamento](#).

- Ação **Transition**

Esta ação arquiva objetos alterando a classe de armazenamento de objetos para Acesso Infrequente ou Arquivo. Quando chega a data ou período de tempo especificado, o sistema altera a classe de armazenamento de objetos para Acesso Infrequente ou Arquivo.

- Para buckets com o versionamento desativado, a ação **Transition** altera a classe de armazenamento de objetos para Acesso Infrequente ou Arquivo.
- Para buckets com o versionamento ativado e suspenso, a ação **Transition** altera a classe de armazenamento da versão atual do objeto para Acesso Infrequente ou Arquivo. Esta ação não tem impacto em versões de objetos não atuais.

- Ação **Expiration**

Esta ação define a hora em que os objetos expiram. Os objetos ficam indisponíveis quando expiram. Se os objetos expirados serão excluídos permanentemente depende do status de o versionamento do bucket.

 **NOTA**

A ação **Expiration** não excluirá partes carregadas de um upload de várias partes incompleto.

- Para buckets com o versionamento desativado, a ação **Expiration** exclui objetos permanentemente e os objetos excluídos não podem ser restaurados.
- Para buckets com o versionamento ativado, essa ação se aplica somente às versões de objetos atuais, em vez de versões de objetos não atuais. Esta ação não exclui permanentemente as versões atuais do objeto. Ele mantém a versão atual como uma versão não atual adicionando um marcador de exclusão a ela. Essa ação não será



executada em versões de objeto atuais que já tenham os marcadores de exclusão. Se a versão atual do objeto for a única versão do objeto e tiver um marcador de exclusão, o OBS excluirá a versão atual do objeto. Limpeza de um marcador de exclusão pode demorar um pouco, porque o OBS precisa confirmar que o marcador de exclusão é a única versão do objeto.

Se você iniciar uma solicitação GET em um objeto cuja versão atual tenha o marcador de exclusão sem especificar o ID da versão, o OBS identificará o objeto como excluído e retornará o erro **404 Object Not Found**. Mas você pode especificar o ID da versão na solicitação GET para recuperar o objeto excluído.

Por exemplo, você pode definir uma regra para fazer com que o objeto chamado **photo.gif** expire 5 dias após o upload. Se **photo.gif** for criado às 10:30 UTC em 1o de janeiro de 2016, a regra de expiração será executada em um ponto de tempo após as 00:00 UTC (cinco dias após a criação do objeto) em 7 de janeiro de 2016. O horário não será posterior às 23:59 UTC de 7 de janeiro de 2016. Para um bucket com o versionamento desativado, uma operação de exclusão exclui permanentemente **photo.gif**. Para um bucket com versionamento ativado, após a execução da regra de expiração, **photo.gif** (versão **111111**) ainda é armazenado no bucket e pode ser acessado se necessário, mas a versão atual (versão **4857693**) do objeto tem um marcador de exclusão. O objeto original **photo.gif** passa a ser uma versão não-atual. Para obter detalhes sobre como funciona um marcador de exclusão, consulte [Versionamento](#).

Para buckets com o versionamento suspenso, o OBS criará marcadores de exclusão para objetos expirados cujo ID de versão é **null**. Quaisquer versões **null** existentes serão substituídas por novas versões **null**, e os dados associados a esta versão não podem ser restaurados.

#### Ações específicas para buckets com controle de versão ativado (ou suspenso)

As ações do ciclo de vida **Transition** e **Expiration** podem gerenciar o ciclo de vida das versões atuais do objeto. As ações **NoncurrentVersionTransition** e **NoncurrentVersionExpiration** podem gerenciar o ciclo de vida de versões de objetos não atuais.

As seguintes ações de configuração do ciclo de vida só podem ser executadas em buckets com versionamento ativado (ou suspenso). Em um bucket com o versionamento ativado, um objeto pode ter várias versões, incluindo uma versão atual e zero ou mais versões não atuais. Você pode usar essas ações para solicitar que o OBS execute operações específicas em versões de objetos não atuais. Essas ações não afetam as versões atuais do objeto.

**NoncurrentVersionTransition**: Especifica o período de tempo após o qual as versões não atuais transitarão de Padrão para Acesso Infrequente ou Arquivo.

**NoncurrentVersionExpiration**: Especifica o período de tempo após o qual as versões de objetos não atuais serão permanentemente excluídas. Um objeto excluído não pode ser restaurado.

Por exemplo, se você quiser habilitar um período de cinco dias para corrigir qualquer exclusão acidental ou substituição, você pode configurar uma regra de expiração para que o objeto possa ser excluído 5 dias depois de se tornar uma versão não atual.

Em 2016-01-01 10:30 (hora UTC), você criou um objeto chamado **photo.gif** cujo ID de versão é **111111**. Em 2016-01-02 10:30 (hora UTC), você excluiu acidentalmente este objeto e OBS criou um marcador de exclusão para este objeto usando um novo ID de versão, **4857693**. Nos próximos cinco dias, você foi autorizado a restaurar o objeto original **photo.gif** cujo ID de versão é **111111**.

Às 00:00 (horário UTC), a ação **NoncurrentVersionExpiration** excluiu permanentemente o objeto **photo.gif** cujo ID de versão é **111111** (depois de se tornar uma versão não atual por 5 dias)

**Como OBS verifica há quanto tempo um objeto se tornou uma versão não atual?**

Em um bucket com o versionamento ativado, um objeto pode ter várias versões, incluindo uma versão atual e zero ou mais versões não atuais. Toda vez que um objeto é carregado, a versão atual existente é salva como uma versão não atual, enquanto a versão carregada se torna a versão atual. Para verificar os dias em que uma versão de objeto se tornou uma versão não atual, OBS verifica a hora de criação do objeto correspondente da versão atual. OBS usa os dias em que o objeto da versão atual foi criado como os dias em que o objeto de uma versão não atual existiu.

**Restauração de uma versão não atual usando a configuração do ciclo de vida**

Você pode usar um dos seguintes métodos para restaurar um objeto para uma versão não atual:

1. Copiar uma versão de objeto não atual para o mesmo bucket. A versão copiada se tornará a versão atual e todas as versões de objetos serão reservadas.
2. Excluir a versão atual do objeto permanentemente. Depois de excluir a versão atual de um objeto, uma versão não atual do objeto se torna a versão atual.

Recomendamos que você use o primeiro método para copiar a versão do objeto não atual para o mesmo bucket. Devido à sintaxe de consistência do OBS, antes que o modo de comunicação seja alterado, uma versão atual que é excluída permanentemente pode não desaparecer (OBS pode não estar ciente dessa ação de exclusão). Além disso, a ação de expiração para versões não atuais, se você tiver configurado, pode excluir as versões de objeto não atuais permanentemente, incluindo aquelas que você deseja restaurar. Portanto, o método de copiar uma versão não atual é mais seguro.

**Tabela 11-1** lista a relação entre a regra de configuração do ciclo de vida e o status de versão de um bucket.

**Tabela 11-1** Ações de configuração e status de o versionamento de buckets

Operação	Buckets com o versionamento desativado	Bucket com o versionamento ativado	Bucket com o versionamento suspenso
Transição (Executado em um objeto quando a data chega ou o período especificado pelo ciclo de vida se esgota.)	Os objetos podem ser alterados para Acesso Infrequente ou Arquivo.	Se a versão atual não tiver um marcador de exclusão e for a versão mais recente, a versão atual poderá ser alterada para Acesso Infrequente ou Arquivo.	Se a versão atual não tiver um marcador de exclusão e não for a versão mais recente, a versão atual poderá ser alterada para Acesso Infrequente ou Arquivo.

Operação	Buckets com o versionamento desativado	Bucket com o versionamento ativado	Bucket com o versionamento suspenso
Expiração (Executado em um objeto quando a data chega ou o período especificado pelo ciclo de vida se esgota.)	A ação <b>Expiration</b> exclui objetos e os objetos excluídos não podem ser restaurados.	Se a versão atual não tiver um marcador de exclusão, um marcador de exclusão será criado e será considerado como a versão atual. A versão atual existente torna-se uma versão não atual.	Para buckets com o versionamento suspenso, o OBS criará marcadores de exclusão para objetos expirados cujo ID de versão é <b>null</b> . Quaisquer versões <b>null</b> existentes serão substituídas por novas versões <b>null</b> , e os dados associados a esta versão não podem ser restaurados.
NoncurrentVersion-Transition (Executado em um objeto depois que ele se torna uma versão não atual por um número especificado de dias.)	A ação NoncurrentVersion-Transition é inválida.	Se a versão atual não for um marcador de exclusão e não for a versão mais recente, a versão atual poderá ser transitada para Acesso Infrequente ou Arquivo.	Se a versão atual não tiver um marcador de exclusão e não for a versão mais recente, a versão atual poderá ser alterada para Acesso Infrequente ou Arquivo.
NoncurrentVersionExpiration (Executado em um objeto depois que ele se torna uma versão não atual por um número especificado de dias.)	A ação <b>NoncurrentVersionExpiration</b> é inválida.	Objetos não atuais são excluídos e não podem ser restaurados depois de excluídos.	Se a versão atual não tiver um marcador de exclusão e não for a versão mais recente, a versão atual poderá ser alterada para Acesso Infrequente ou Arquivo.

### Regras de ciclo de vida baseadas em data

Você pode especificar as datas de execução das ações de **Transition** e **Expiration**. As datas devem estar em conformidade com os padrões ISO 8601 e a hora exata é sempre 00:00 (hora UTC). Se você especificar a hora para uma data passada, todos os objetos aplicáveis serão executados imediatamente.

Não é possível criar uma regra do ciclo de vida baseada em data no console de OBS.

### ATENÇÃO

Uma ação do ciclo de vida com uma data especificada não é uma ação única. Mesmo que a data tenha passado, OBS adotará essa ação desde que o ciclo de vida esteja ativado.

Suponha que você tenha especificado uma data para executar a ação **Expiration** para excluir todos os objetos (sem definir nenhum critério de filtragem). Na data especificada, OBS faz com que todos os objetos no bucket expirem. Além disso, OBS continua a fazer com que todos os novos objetos criados no bucket expirem. Para encerrar a ação **Expiration**, é necessário excluir essa ação da configuração do ciclo de vida, desabilitar a regra ou excluí-la da configuração do ciclo de vida.

### Regra de ciclo de vida baseada em período de tempo

Você pode especificar quantos dias após a criação de um objeto a ação **Transition** ou **Expiration** será executada no objeto. Depois que o número de dias é especificado, o OBS começa a calcular a hora a partir das 00:00 (hora UTC) do dia seguinte. Por exemplo, você criou um objeto em 2016-01-15 10:30 (hora UTC) e especificou que os objetos seriam alterados 3 dias após a criação, o objeto seria transitado em 2016-01-19 00:00 (hora UTC).

### AVISO

O OBS registra apenas a data da última modificação para cada objeto. No console de OBS, você pode exibir o horário da última modificação (**LastModified**) de um objeto na página de propriedades do objeto. Depois que um objeto é criado, a data é a data de criação. Se o objeto for substituído, a data também será alterada.

Ao usar a ação **NoncurrentVersionTransition** ou **NoncurrentVersionExpiration**, você pode especificar quantos dias depois que um objeto mudar para uma versão não atual (devido a substituição ou exclusão) a ação será executada no objeto.

Quando você usa as operações **NoncurrentVersionTransition** e **NoncurrentVersionExpiration** para especificar o número de dias, OBS adiciona o número de dias especificado na regra ao momento em que a versão do objeto se torna uma versão não atual. Em seguida, o tempo de operação é calculado a partir das 00:00 (hora UTC) do dia seguinte. Por exemplo, a versão atual de um objeto em um bucket é criada às 10:30 UTC em 10 de janeiro de 2016. Se a versão do objeto se tornar uma versão não atual às 10:30 UTC de 15 de janeiro de 2016 e você especificar três dias na regra de conversão, a data para alterar a classe de armazenamento do objeto será calculada como 00:00 UTC de 19 de janeiro de 2016.

### NOTA

Ao configurar as regras do ciclo de vida, dentro de uma regra e para regras cujos prefixos tenham relação de inclusão, **Date** ou **Days** de **Transition** e **Expiration** devem ser consistentes.

## Como usar essa função

Você pode usar console de OBS, as API ou os SDK para configurar regras do ciclo de vida.

Ferramenta	Referência
Console de OBS	<a href="#">Configuração de uma regra do ciclo de vida</a>

Ferramenta	Referência
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Configuração de regras do ciclo de vida do bucket</a> <a href="#">Obtenção da configuração do ciclo de vida do bucket</a> <a href="#">Exclusão de regras do ciclo de vida</a>

# 12 Replicação entre regiões

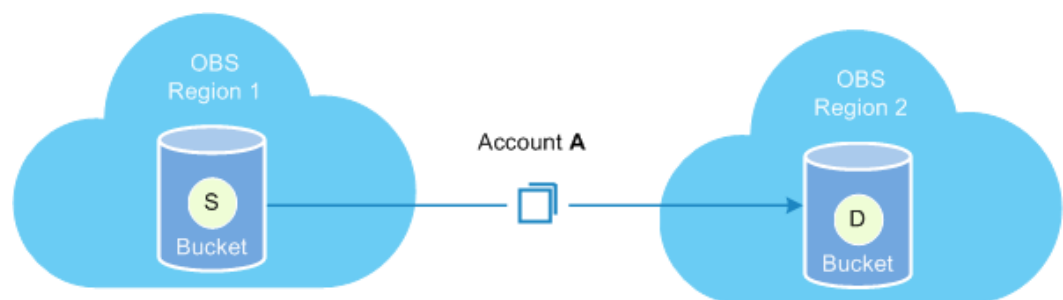
## Cenário de aplicação

A replicação entre regiões oferece a capacidade de recuperação de desastres entre regiões, permitindo que você configure uma solução de backup remoto.

A replicação entre regiões refere-se ao processo de replicação automática e assíncrona de dados de um bucket (bucket de origem) para outro bucket (bucket de destino) entre regiões, criando uma regra de replicação entre regiões. O bucket de origem e o bucket de destino devem pertencer à mesma conta. A replicação entre contas não é suportada.

Para uma regra de replicação entre regiões, você pode configurá-la para corresponder a um prefixo de objeto predefinido para que todos os objetos com esse prefixo sejam replicados. Como alternativa, você pode configurar a regra para aplicar ao bucket inteiro para que todos os objetos no bucket sejam replicados. Os objetos replicados para o bucket de destino são cópias precisas de objetos no bucket de origem. Eles têm os mesmos nomes de objetos, metadados, conteúdo, tamanho, hora da última modificação, criador, ID de versão, metadados definidos pelo usuário e as ACL. Por padrão, a classe de armazenamento de uma cópia de objeto é a mesma do objeto de origem. Você também pode especificar uma classe de armazenamento diferente para uma cópia de objeto.

**Figura 12-1** Replicação entre regiões



- Conformidade regulamentar  
OBS armazena dados nas AZ que estão relativamente distantes umas das outras. No entanto, a conformidade regulatória pode exigir distâncias maiores. A replicação entre regiões permite que OBS replique dados entre regiões para conformidade regulamentar.
- Latências minimizadas

Os mesmos recursos de OBS podem precisar ser acessados de locais diferentes. Para minimizar a latência de acesso, você pode usar a replicação entre regiões para criar cópias de objetos na região mais próxima dos usuários finais.

- Replicação de dados  
A replicação entre regiões permite migrar facilmente os dados armazenados em OBS de uma região para outra.
- Backup de dados e recuperação de desastres  
Para garantir a segurança e a disponibilidade dos dados, você precisa criar backups explícitos para todos os dados gravados em OBS no centro de dados de outra região, para que o backup seguro esteja disponível caso os dados de origem sejam danificados irrevogavelmente.
- Manutenção facilitada  
Você tem um cluster de computação entre regiões para analisar a mesma coleção de objetos. Você precisa manter réplicas de objetos nas duas regiões.

## Conteúdo replicado

Após a ativação da regra de replicação entre regiões, os objetos que atendem às seguintes condições são copiados para o bucket de destino:

- Objetos carregados recentemente (exclusão de objetos na classe de armazenamento Arquivo)
- Objetos atualizados. Por exemplo, o conteúdo do objeto é atualizado ou as informações de ACL de um objeto copiado são atualizadas.
- Objetos históricos em um bucket se **Synchronizing Existing Objects** estiver ativado (exclusão de objetos na classe de armazenamento Arquivo)

### NOTA

A replicação entre regiões não replica objetos criptografados usando SSE-C.

## Restrições

- Uma regra de replicação entre regiões pode não ter efeito imediatamente após sua configuração. Assim, os objetos aos quais essa regra é aplicada podem não ser replicados imediatamente após a configuração da regra.
- Atualmente, apenas os buckets da versão 3.0 oferecem suporte à replicação entre regiões. Para verificar a versão do bucket, vá para a página **Overview** do bucket no console de OBS. Em seguida, você pode visualizar a versão do bucket na área **Basic Information**.
- Por padrão, os objetos carregados antes da ativação da replicação entre regiões não são copiados para o bucket de destino, a menos que a função de sincronização de objetos existentes esteja ativada.
- O bucket de origem e o bucket de destino devem pertencer a regiões diferentes separadamente. Os dados não podem ser copiados entre buckets na mesma região.
- Os objetos não podem ser copiados do bucket de origem para o bucket de destino se estiverem na classe de armazenamento Arquivo.
- Se a região em que o intervalo de destino reside não oferecer suporte a classes de armazenamento diferentes para dados, as cópias do objeto serão armazenadas na classe de armazenamento Padrão.
- O status de controle de versão do bucket de origem deve ser o mesmo do bucket de destino.

- Os objetos em um bucket de origem podem ser copiados para somente um bucket de destino e não podem ser copiados novamente do bucket de destino para outro bucket. Por exemplo, o bucket A e o bucket B estão em duas regiões diferentes. Você pode copiar dados do bucket A para o bucket B ou vice-versa. No entanto, as cópias de dados no bucket A ou no bucket B não podem mais ser replicadas.
- Se o controle de versão estiver ativado ou suspenso para os buckets de origem e de destino e a replicação entre regiões também estiver ativada para ambos os buckets, a exclusão de um objeto sem especificar sua versão no bucket de origem também excluirá o objeto no bucket de destino.
- Se você alterar o status de controle de versão do bucket de destino quando a replicação entre regiões estiver ativada, a replicação dos objetos falhará. Se você quiser alterar o status de controle de versão do bucket de origem, desative a replicação entre regiões primeiro e, em seguida, faça a alteração.
- Certifique-se de que os proprietários dos buckets de origem e destino tenham as permissões de leitura e gravação nos dois buckets. Caso contrário, os dados não podem ser sincronizados. Se o sistema não tiver as permissões para ler o bucket de origem ou gravar o bucket de destino devido a erros de permissão de leitura/gravação, os objetos não poderão ser copiados com êxito e essa replicação não será retomada mesmo que o erro de permissão seja corrigido.
- Para um bucket de origem, você pode criar somente uma regra de replicação entre regiões que se aplica a todo o bucket para replicação de todos os objetos no bucket. No entanto, você pode criar um máximo de 100 regras de replicação entre regiões com base em prefixos de objeto para a replicação de objetos que correspondam aos prefixos.
- Atualmente, OBS suporta apenas a replicação entre um bucket de origem e um bucket de destino. A replicação de um bucket de origem para vários buckets de destino não é suportada. O bucket de destino pode ser modificado. No entanto, modificar o bucket de destino alterará o bucket de destino de todas as regras existentes.
- Se eliminar a configuração de agência de OBS numa replicação entre regiões, o estado de replicação torna-se **Failed**.
- Não exclua, substitua réplicas de objeto no bucket de destino ou modifique suas ACL, o que pode causar inconsistência das últimas versões de objeto ou configurações de controle de permissão entre o bucket de destino e o bucket de origem.
- Se a função para sincronizar objetos existentes estiver habilitada, modificar a configuração de replicação entre regiões pode causar falhas na sincronização de objetos existentes. Por conseguinte, não modifique a configuração de replicação entre regiões antes da conclusão da sincronização.
- Se a replicação entre regiões estiver ativada, os dados não poderão ser adicionados ao final dos objetos no intervalo de origem.
- Após a conclusão de uma replicação com **Synchronize Existing Objects** ativada, se a diretiva de replicação se mantiver inalterada, quaisquer alterações de ACL de objetos de origem serão sincronizadas com as cópias de objetos. No entanto, as alterações ACL de objetos históricos de origem não serão sincronizadas com as cópias de objetos históricos.
- Antes de replicar os dados, certifique-se de que as regiões de origem e de destino possam ter seus dados replicados uma da outra. **Figura 12-2** lista as regiões suportadas. ✓ definição indica que os dados podem ser replicados entre regiões. x indica que os dados não podem ser replicados entre regiões.



**Figura 12-2** Replicação entre regiões

	CN North-Beijing4	CN East-Shanghai1	CN East-Shanghai2	CN North-Beijing1	CN North-Beijing2	CN North-Ulanqab1	CN Southwest-Guiyang1	CN South-Guangzhou	CN-Hong Kong	AP-Bangkok	LA-Sao Paulo1	AF-Johannesburg	LA-Mexico City1	AP-Singapore
CN North-Beijing4	/	√	√	√	√	√	√	√	x	x	x	x	x	x
CN East-Shanghai1	√	/	√	√	√	√	√	√	x	x	x	x	x	x
CN East-Shanghai2	√	√	/	√	√	√	√	√	√	x	x	√	x	x
CN North-Beijing1	√	√	√	/	√	√	√	√	x	x	x	√	x	x
CN North-Beijing2	√	√	√	√	/	√	√	√	x	x	x	x	x	x
CN North-Ulanqab1	√	√	√	√	√	/	√	√	x	x	x	x	x	x
CN Southwest-Guiyang1	√	√	√	√	√	√	/	√	x	x	x	x	x	x
CN South-Guangzhou	√	√	√	√	√	√	√	/	x	x	x	x	x	x
CN-Hong Kong	x	x	√	x	x	x	x	x	/	x	x	√	√	x
AP-Bangkok	x	x	x	x	x	x	x	x	x	/	x	x	x	x
LA-Sao Paulo1	x	x	x	x	x	x	x	x	x	x	/	x	x	x
AF-Johannesburg	x	x	√	√	x	x	x	x	√	x	x	/	x	x
LA-Mexico City1	x	x	x	x	x	x	x	x	√	x	x	x	/	x
AP-Singapore	x	x	x	x	x	x	x	x	x	x	x	x	x	/

## Como usar essa função

Você pode usar console de OBS, as API, os SDK, ou obsutil para configurar a replicação entre regiões.

Ferramenta	Referência
Console de OBS	<a href="#">Configuração da replicação entre regiões</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Configuração da replicação entre regiões para um bucket</a>
obsutil	<a href="#">Cópia de objetos</a>

# 13 Encriptação do lado do servidor

---

## Cenário de aplicação

Depois que a encriptação do lado do servidor for ativada, os objetos carregados no OBS serão criptografados e armazenados no servidor. Quando você baixar os objetos criptografados, os dados criptografados serão descriptografados no servidor e exibidos em texto simples para você.

OBS suporta os seguintes dois modos de encriptação do lado do servidor. Ambos os modos usam o algoritmo de encriptação AES-256 padrão da indústria.

- **Criptografia do lado do servidor usando chaves hospedadas pelo KMS (SSE-KMS)**

Você precisa criar uma chave usando o Key Management Service (KMS) ou usar a chave padrão fornecida pelo KMS. A chave KMS é então usada para executar a encriptação do lado do servidor quando você carrega objetos para OBS.

No modo SSE-KMS, você pode ativar a encriptação padrão ao criar um intervalo. Em seguida, todos os objetos carregados no bucket são criptografados. Você também pode ativar a encriptação padrão para um bucket após sua criação e, em seguida, os objetos recém-carregados são criptografados.

OBS criptografa somente os objetos carregados após a função de encriptação padrão ser ativada. O status de encriptação dos objetos existentes no bucket permanece inalterado. Desabilitar a encriptação padrão não altera o status de encriptação dos objetos existentes em um bucket. Depois que essa função for desabilitada, você ainda poderá criptografar objetos manualmente após o upload.

Você pode usar o Console do OBS, as API, os SDK ou OBS Browser+ para configurar o SSE-KMS.
- **Criptografia do lado do servidor com chaves fornecidas pelo cliente (SSE-C)**

OBS usa as chaves e os valores MD5 fornecidos pelos clientes para encriptação do lado do servidor.

Você pode usar as API ou os SDK para configurar o SSE-C.

## Informações de plano de fundo

No modo SSE-KMS, o KMS usa um módulo de segurança de hardware (HSM) para proteger a segurança da chave, ajudando você a criar e controlar facilmente chaves de encriptação. As chaves não são exibidas em texto sem formatação fora dos HSM, o que impede a sua divulgação. Todas as operações realizadas nas chaves são controladas usando permissões de acesso e registradas, atendendo aos requisitos de conformidade regulatória.

## Como usar essa função

Você pode usar Console do OBS, as API, os SDK, ou OBS Browser+ para definir a encriptação do lado do servidor.

Ferramenta	Referência
Console de OBS	<a href="#">Carregamento de um arquivo com encriptação do lado do servidor</a> <a href="#">Encriptação padrão de um bucket</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Encriptação do lado do servidor (SSE-KMS)</a> <a href="#">Encriptação do lado do servidor (SSE-C)</a> <a href="#">Configuração da encriptação de bucket</a>
OBS Browser+	-

# 14 Hospedagem de site estático

---

## Precauções

Para fins de segurança e conformidade, usando a hospedagem de sites estáticos por meio do nome de domínio padrão do OBS (**nome de domínio do bucket ou nome de domínio do site estático**) serão proibidos por OBS. Quando você usa esse nome de domínio para acessar páginas da web por meio de um navegador, nenhum conteúdo será exibido; em vez disso, o conteúdo será baixado como um anexo.

Esta proibição entrará em vigor em diferentes regiões nos dois momentos seguintes:

1 de janeiro de 2022: CN North-Beijing1, CN North-Beijing4, CN East-Shanghai1, CN East-Shanghai2, e CN South-Guangzhou

25 de março de 2022: CN-Hong Kong, AP-Bangkok, AP-Singapore, AF-Johannesburg, LA-Mexico City1, LA-Mexico City2, LA-Sao Paulo1, e LA-Santiago

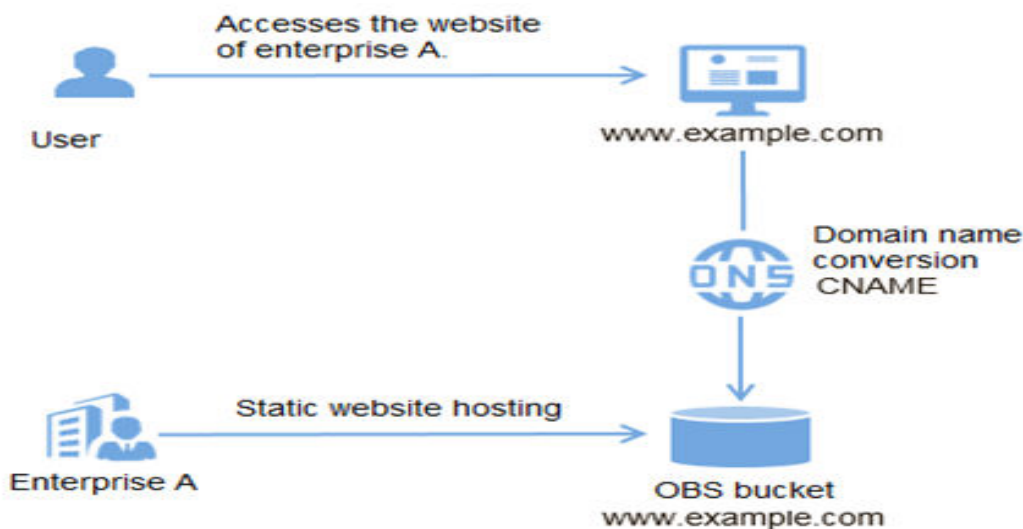
Você ainda pode usar a hospedagem de sites estáticos por meio de um nome de domínio definido pelo usuário. Dessa forma, o conteúdo ainda pode ser visualizado. Para obter detalhes, consulte [Como faço para visualizar objetos no OBS através de um navegador?](#)

## Cenário de aplicação

Você pode usar OBS para hospedar seu site estático, para o qual também pode configurar um documento de índice, um documento de erro e a direção da página. Você pode carregar os arquivos de conteúdo do site estático para o bucket no OBS e configurar uma permissão de leitura para usuários anônimos para esses arquivos e, em seguida, configurar o modo de hospedagem de site estático para o bucket hospedar seus sites estáticos no OBS.

Sites estáticos contêm páginas da web estáticas e alguns scripts que podem ser executados em clientes, como JavaScript e Flash. Diferente de sites estáticos, sites dinâmicos contam com servidores para processar scripts, incluindo PHP, JSP e ASP.NET. OBS não suporta scripts executados em servidores.

**Figura 14-1** Hospedagem de site estático



Depois que a hospedagem de sites estáticos estiver ativada, você poderá acessar objetos em um bucket usando um dos seguintes métodos:

- **Gestão de recursos:** Você pode acessar objetos em um bucket por meio de um nome de domínio comum. Você pode usar as API ou os SDK e nomes de domínio padrão do OBS com endpoints contidos para executar operações comuns em buckets e objetos, como upload, download, exclusão e listagem.
- **Site estático:** Você pode acessar objetos em um bucket usando um nome de domínio específico. Neste modo, a exibição da página de índice, páginas de erro e páginas solicitadas é implementada de acordo com as regras de hospedagem de nomes de domínio estáticos configuradas.

A configuração da hospedagem estática do site entra em vigor em dois minutos. Depois que ele entrar em vigor, você poderá acessar recursos estáticos usando os seguintes nomes de domínio de acesso:

```
https://bucketname.OBS static website hosting domain name/Object name
http://bucketname.OBS static website hosting domain name/Object name
```

**NOTA**

Evite usar (.) de períodos no nome do bucket de destino. Caso contrário, falhas no certificado de autenticação do cliente podem ocorrer quando os usuários usam HTTPS para acesso.

Para permitir que seus clientes acessem o conteúdo no nó do terminal do site, você deve tornar todo o seu conteúdo público e acessível. Você pode usar políticas de bucket ou ACL em objetos para conceder permissões.

A tabela a seguir lista as diferenças entre o modo de gerenciamento de recursos e o modo de site estático.

Diferença principal	Gestão de recursos	Site estático
Controle de acesso	Tanto o conteúdo público quanto o conteúdo privado são suportados.	Somente o conteúdo público é suportado.

Diferença principal	Gestão de recursos	Site estático
Processamento de mensagem de erro	Uma resposta de erro no formato XML é retornada.	Um documento HTML é retornado.
Suporte ao redirecionamento	N/A	Tanto o redirecionamento no nível do objeto quanto o no nível do bucket são suportados.
Solicitação suportada	Operações em todos os buckets e objetos são suportadas.	Somente solicitações GET e HEAD em objetos são suportadas.
Resposta a solicitações GET e HEAD do nível raiz do bucket	A lista de chaves de objeto em um bucket é retornada.	O arquivo de índice especificado na configuração é retornado.

## Visão geral do redirecionamento

Ao usar a hospedagem de sites estáticos, você pode configurar o redirecionamento para redirecionar solicitações específicas ou todas.

Se a estrutura, endereço ou extensão de nome de arquivo de um site for alterada, os usuários não conseguirão acessar o site usando o endereço antigo. (como o endereço salvo na pasta de favoritos) e a mensagem de erro 404 será retornada. Nesse caso, você pode configurar o redirecionamento do site para redirecionar as solicitações de acesso do usuário para a página especificada, em vez de retornar a página de erro 404.

As configurações típicas incluem:

- Redirecionamento de todas as solicitações para outro site.
- Redirecionamento de solicitações específicas com base em regras de redirecionamento.

## Configuração da hospedagem de site estático

### Visão geral

Se você quiser usar um bucket para hospedar sites estáticos, adicione a configuração do site ao bucket. A configuração inclui as seguintes informações:

#### Documento de índice

Quando você insere uma URL como **http://example.com**, você não está solicitando uma página específica. Nesse caso, o servidor web fornecerá uma página padrão que contém o diretório que armazena o conteúdo do site solicitado. Essa página padrão é chamada de documento de índice e, na maioria dos casos, é nomeada como **index.html**. Ao configurar um bucket para hospedagem de sites, você deve especificar um documento de índice. Quando uma solicitação é enviada para o domínio raiz ou qualquer subpasta, OBS retorna esse documento de índice.

#### Documento de erro

Se ocorrer um erro, OBS retornará um documento de erro de HTML. Para erros 4XX, você pode fornecer seu próprio documento de erro personalizado ou fornecer outros guias para seus usuários neste documento.

### Redirecionamento de todas as solicitações

Se o domínio raiz for **example.com** e você precisar responder a solicitações de **http://example.com** e **http://www.example.com**, poderá criar dois buckets denominados **example.com** e **www.example.com**. Em seguida, você pode manter o conteúdo do site em apenas um bucket (como **example.com**) e configurar o outro bucket para redirecionar todas as solicitações para o bucket **example.com**.

### Redirecionamento baseado em condições avançadas

Você pode redirecionar solicitações com base no nome ou prefixo do objeto específico na solicitação ou com base no código de resposta. Por exemplo, suponha que você exclua ou renomeie um objeto em um bucket. Você pode adicionar uma regra de roteamento que redireciona solicitações para outros objetos.

O formato de sintaxe é o seguinte, definindo regras de redirecionamento de solicitações específicas:

```
<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>index.html</Suffix></IndexDocument>
  <ErrorDocument>
    <Key>SomeErrorDocument.html</Key>
  </ErrorDocument>
  <RoutingRules>
    RoutingRules.....
  </RoutingRules>
</WebsiteConfiguration>
```

O formato de sintaxe do **RoutingRules** é o seguinte. O conteúdo entre colchetes é opcional.

```
<RoutingRules> =
  <RoutingRules>
    <RoutingRule>...</RoutingRule>
    [ <RoutingRule>...</RoutingRule> ... ]
  </RoutingRules>

<RoutingRule> =
  <RoutingRule>
    [ <Condition>...</Condition> ]
    <Redirect>...</Redirect>
  </RoutingRule>

<Condition> =
  <Condition>
    [ <KeyPrefixEquals>...</KeyPrefixEquals> ]
    [ <HttpErrorCodeReturnedEquals>...</HttpErrorCodeReturnedEquals> ]
  </Condition>

<Redirect> =
  <Redirect>
    [ <HostName>...</HostName> ]
    [ <Protocol>...</Protocol> ]
    [ <ReplaceKeyPrefixWith>...</ReplaceKeyPrefixWith> ]
    [ <ReplaceKeyWith>...</ReplaceKeyWith> ] [
    <HttpRedirectCode>...</HttpRedirectCode> ]
  </Redirect>
```

Para elementos de solicitação necessários para redirecionar todas as solicitações enviadas a um site especificado e para definir regras de redirecionamento, consulte [Configuração da hospedagem de sites estáticos para um bucket – elementos de solicitação](#)

Exemplos:

### Exemplo 1: Modificação do prefixo do nome do objeto para redirecionamento

Suponha que o bucket contém os seguintes objetos:

index.html

docs/article1.html

docs/article2.html

Você decide mudar o nome da pasta de **docs/** para **documents/**. Após a modificação, a solicitação de um objeto com prefixo **/docs** precisa ser redirecionada para outro com **documents/**. Por exemplo, a solicitação para **docs/article1.html** precisa ser redirecionada para **documents/article1.html**.

Nesse caso, você pode adicionar as seguintes regras de roteamento à configuração do site:

```
<RoutingRules>
  <RoutingRule>
    <Condition>
      <KeyPrefixEquals>docs/</KeyPrefixEquals>
    </Condition>
    <Redirect>
      <ReplaceKeyPrefixWith>documents/</ReplaceKeyPrefixWith>
    </Redirect>
  </RoutingRule>
</RoutingRules>
```

### Exemplo 2: Redirecionamento de solicitações enviadas para pastas excluídas para uma página especificada

Suponha que você tenha excluído a pasta **images/**, ou seja, você tenha excluído todos os objetos cujo prefixo do nome do objeto é **images/**. Você pode adicionar uma regra de roteamento que redireciona as solicitações de todos os objetos cujo prefixo é **images/** para a página chamada **folderdeleted.html**.

```
<RoutingRules>
  <RoutingRule>
    <Condition>
      <KeyPrefixEquals>images/</KeyPrefixEquals>
    </Condition>
    <Redirect>
      <ReplaceKeyWith>folderdeleted.html</ReplaceKeyWith>
    </Redirect>
  </RoutingRule>
</RoutingRules>
```

### Exemplo 3: Redirecionamento quando ocorrem erros HTTP

Suponha que o objeto solicitado não é encontrado, e a solicitação precisa ser redirecionada para **www.example.com**. Você pode adicionar regras de redirecionamento para que os visitantes possam redirecionar para **www.example.com** quando o código de status de HTTP 404 (não encontrado) for retornado. O exemplo a seguir também insere o prefixo de nome de objeto **report-404/** no redirecionamento. Por exemplo, se você solicitar a página **ExamplePage.html** e causar um erro HTTP 404, a solicitação será redirecionada para a página **report-404/ExamplePage.html** no **www.example.com**. Se não houver nenhuma regra de roteamento e ocorrer um erro de HTTP 404, o documento de erro especificado na configuração será retornado.

```
<RoutingRules>
  <RoutingRule>
    <Condition>
```



```
<HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals >  
</Condition>  
<Redirect>  
  <HostName>www.example.com</HostName>  
  <ReplaceKeyPrefixWith>report-404</ReplaceKeyPrefixWith>  
</Redirect>  
</RoutingRule>  
</RoutingRules>
```

## Suporte a documentos de índice

Um documento de índice é uma página da web retornada quando uma solicitação é enviada para a raiz de um site ou qualquer subpasta. Por exemplo, se o usuário inserir **http://www.example.com** no navegador, o usuário não solicitará nenhuma página específica. Nesse caso, OBS fornece documentos de índice, que às vezes são chamados de páginas padrão.

Ao configurar seu bucket como um site, forneça o nome do documento de índice. Você deve carregar o objeto com esse nome e configurá-lo para ser público e acessível.

A barra de rastreamento no URL raiz é opcional. Por exemplo, se você configurar um site **index.html** como um documento de índice, **index.html** será retornado para qualquer um dos seguintes os URL:

```
http://bucketname.obs-website.example.com/  
http://bucketname.obs-website.example.com
```

No OBS, os objetos são armazenados horizontalmente em buckets. Ele não fornece nenhuma organização hierárquica como um sistema de arquivos em um computador. Você pode criar uma hierarquia lógica usando o nome do objeto que representa a estrutura de pastas. Por exemplo, considere um bucket com três objetos e os seguintes nomes de objetos.

- sample1.jpg
- photos/2006/Jan/sample2.jpg
- photos/2006/Feb/sample3.jpg

Embora eles não sejam armazenados como qualquer organização hierárquica física, você pode inferir a seguinte estrutura de pasta lógica a partir do nome do objeto.

- O objeto **sample1.jpg** está localizado no nível raiz do bucket.
- O objeto **sample2.jpg** está localizado na subpasta **photos/2006/Jan**.
- O objeto **sample3.jpg** está localizado na subpasta **photos/2006/Feb**.

## Suporte a documentos de erros personalizados

**Tabela 14-1** lista o subconjunto do código de resposta de HTTP retornado pelo OBS quando ocorre um erro.

**Tabela 14-1** Lista de códigos de erro de OBS

Código de erro de HTTP	Descrição
301 Moved Permanently	Quando um usuário envia uma solicitação para um endpoint de OBS, uma resposta <b>301 Moved Permanently</b> é retornada.

Código de erro de HTTP	Descrição
302 Found	Quando o OBS recebe uma solicitação para a chave <i>x</i> que não termina com uma barra, ele procura o objeto cujo nome da chave é <i>x</i> . Se o objeto não for encontrado, o OBS determina que a solicitação é enviada para a subpasta <i>x</i> . Em seguida, o OBS redireciona a solicitação adicionando uma barra no final da solicitação e retorna <b>302 Found</b> .
304 Not Modified	Os usuários do OBS solicitam os cabeçalhos <b>If-Modified-Since</b> e <b>If-Unmodified-Since</b> para determinar se o objeto solicitado é o mesmo que a cópia armazenada em cache no cliente. Se os objetos forem os mesmos, o nó do terminal do site retorna uma resposta <b>304 Not Modified</b> .
403 Forbidden	Quando a solicitação do usuário é convertida em um objeto que não pode ser lido publicamente, a resposta do nó do terminal do site inclui <b>403 Forbidden</b> . O proprietário do objeto deve usar políticas de bucket ou ACL para tornar o objeto público e acessível.
404 Not Found	A resposta do nó do terminal do site contém <b>404 Not Found</b> devido aos seguintes motivos: <ul style="list-style-type: none"> <li>● O OBS determina que uma chave de objeto inválida é referenciada no URL do site.</li> <li>● O OBS determina que a solicitação é para um documento de índice que não existe.</li> <li>● O bucket especificado no URL não existe.</li> <li>● O bucket especificado no URL existe, mas não está configurado como um site.</li> </ul> Você pode criar documentos personalizados retornados para <b>404 Not Found</b> . Certifique-se de que o documento foi carregado no bucket configurado como um site e que a configuração de hospedagem do site foi definida para usar o documento.
<b>500 Service Error</b>	Quando ocorre um erro de servidor interno, a resposta do nó do terminal do site contém <b>500 Service Error</b> .

Código de erro de HTTP	Descrição
503 Service Unavailable	Quando o OBS determina que você precisa reduzir a frequência de solicitação, a resposta do nó do terminal do site contém <b>503 Service Unavailable</b> .

## Permissões necessárias para acesso ao site

Ao configurar um bucket como um site, você deve tornar o objeto a ser fornecido público e acessível. Para executar essa operação, você precisa escrever uma política de bucket que conceda permissões de GetObject a todos. Se o objeto solicitado pelo usuário não existir no nó terminal do site, OBS retornará o código de resposta de HTTP **404 Not Found**. Se o objeto existir, mas você não conceder permissão de leitura para o objeto, o nó do terminal do site retornará o código de resposta HTTP **403 Access Denied**. Você pode usar esse código de resposta para inferir se um objeto específico existe. Se você não precisar dessa função, não habilite o suporte ao site do bucket.

### NOTA

No cenário de hospedagem de sites estáticos, os usuários anônimos devem ter acesso ao arquivo de site estático hospedado. Durante seu acesso ao arquivo hospedado, taxas para o tráfego de Internet de saída e solicitações serão incorridas.

A política de bucket de exemplo a seguir concede a cada pessoa a permissão para acessar objetos em uma pasta especificada. Para obter mais informações sobre políticas de bucket, consulte [Política de bucket](#).

```
{
  "Statement": [{
    "Sid": "PublicReadGetObject",
    "Effect": "Allow",
    "Principal": {"ID": "*"},
    "Action": ["GetObject"],
    "Resource": ["example-bucket/*" ]
  }
]
```

### AVISO

Uma política de bucket se aplica somente a objetos de propriedade de um proprietário de bucket. Se o bucket contiver objetos que não são de propriedade do proprietário do bucket, use ACL de objeto para conceder a permissão de leitura pública aos objetos.

Você pode usar políticas de bucket ou ACL de objeto para conceder permissões de leitura pública aos seus objetos. Para usar ACL para tornar objetos públicos e acessíveis, você pode conceder a permissão de leitura a todos, conforme mostrado nos seguintes elementos de autorização. Você pode adicionar o elemento de autorização às ACL de objeto. Para obter mais informações sobre como gerenciar os ACL, consulte [ACL](#).

```
<Grant>
  <Grantee>
    <Canned>Everyone</Canned>
  </Grantee>
```

```
<Permission>READ</Permission>  
</Grant>
```

## Funções relacionadas

Função	Descrição do relacionamento	Referência
CORS	Por padrão, os sites estáticos hospedados em um bucket do OBS só podem responder a solicitações de acesso de sites no mesmo domínio. Você pode configurar o compartilhamento de recursos de origem cruzada (CORS) para o bucket para que os sites estáticos possam ser acessados por solicitações entre domínios.	<a href="#">CORS</a>
Vinculação de nome de domínio definido pelo usuário	OBS permite que você use um nome de domínio definido pelo usuário para acessar sites estáticos hospedados em um bucket do OBS. Você pode usar o nome de domínio original para acessar o site sem modificar o código do site.	<a href="#">Vinculação de nome de domínio definido pelo usuário</a> <a href="#">Uso de um nome de domínio definido pelo usuário para hospedar um site estático</a>
Validação de URL	OBS usa validação de URL para impedir que seus sites hospedados em um bucket de OBS sejam roubados. OBS verifica os URL com base no campo de referência no cabeçalho de HTTP.	<a href="#">Validação de URL</a>

## Como usar essa função

Você pode usar console do OBS, as API, ou os SDK para configurar CORS.

Ferramenta	Referência
Console de OBS	<a href="#">Configuração da hospedagem de website estático</a>
SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
API	<a href="#">Configuração da hospedagem de website estático para um bucket</a>

# 15 CORS

O compartilhamento de recursos de origem cruzada (CORS) é um mecanismo padrão de navegador fornecido pelo World Wide web Consortium (W3C). Ele define os métodos de interação entre aplicações web do lado do cliente em uma origem e recursos em outra. Para solicitações gerais de páginas da web, os scripts e conteúdos de sites em uma origem não podem interagir com os de outra devido às Políticas de Mesma Origem (as SOP).

OBS é compatível com as regras de CORS e permite que seus recursos sejam acessados entre origens. A configuração de CORS entra em vigor em dois minutos.

OBS suporta **Hospedagem de site estático**. Os sites estáticos armazenados em OBS podem responder a solicitações de sites de outra origem somente quando CORS estiver configurado para o bucket.

## AVISO

Por padrão, o sistema OBS é configurado para oferecer suporte ao acesso entre domínios usando o nome de domínio raiz. Isso permite o acesso de todos os domínios e os clientes provavelmente serão atacados.

Para resolver esse problema, crie um arquivo **crossdomain.xml** com regras específicas no bucket para cada cliente e adicione **Security.loadPolicyFile("https://bucket.obs.cn-north-4.myhuaweicloud.com/crossdomain.xml")** no código flash do arquivo para evitar ataques. **bucket.obs.cn-north-4.myhuaweicloud.com** precisa ser substituído pelo nome de domínio de acesso real do seu bucket.

## Informações de plano de fundo

Cross-domain refere-se ao acesso entre diferentes domínios.

Restringir o acesso entre domínios é uma política do navegador para fins de segurança, ou seja, a política de mesma origem. Devido a esta política JavaScript da mesma origem, o JavaScript sob o domínio A não pode operar objetos sob o domínio B ou C.

O mesmo protocolo, nome de domínio (ou endereço IP) e porta são considerados como o mesmo domínio. Se os protocolos, nomes de domínio e portas (se especificado) das duas páginas da web forem os mesmos, as duas páginas da web terão a mesma origem. Para entender melhor a política de mesma origem, você pode ver a análise sobre como acessar o exemplo de endereço **https://support.huaweicloud.com/dir/test.html** em **Tabela 15-1**.

**Tabela 15-1** Exemplo de análise

URL	Resultado de acesso	Causa
https://support.huaweicloud.com/dir/other.html	Bem-sucedido	Mesmo protocolo, nome de domínio e porta
https://support.huaweicloud.com/dir/inner/other.html	Bem-sucedido	Mesmo protocolo, nome de domínio e porta
http://support.huaweicloud.com/dir/test.html	Falha	Mesmo nome de domínio e porta, mas protocolos diferentes
https://support.huaweicloud.com:81/dir/test.html	Falha	Mesmo protocolo e nome de domínio, mas portas diferentes
https://help.huaweicloud.com/dir/test.html	Falha	Mesmo protocolo e porta, mas nomes de domínio diferentes

## Cenários de aplicação

Os cenários de aplicação típicos do CORS são os seguintes:

- Permite que JavaScript e HTML5 sejam usados para estabelecer aplicativos da web que podem acessar diretamente recursos em OBS. Nenhum servidor proxy é necessário para a transferência.
- Permite que a função de arrastar do HTML5 seja usada para carregar arquivos para OBS (com o andamento do carregamento exibido) ou atualizar o conteúdo de OBS usando aplicativos da web.
- Páginas externas da web, folhas de estilo e aplicativos HTML5 hospedados em diferentes origens podem acessar fontes da web ou imagens armazenadas em OBS, implementando o compartilhamento de recursos.

## Como usar essa função

Você pode configurar CORS usando console de OBS, as API, ou os SDK.

Ferramenta	Referência
Console de OBS	<a href="#">Configuração de CORS</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Configuração de CORS de bucket</a>

# 16 Vinculação de nome de domínio definido pelo usuário

---

## Cenário de aplicação

Se você precisar migrar arquivos de um site para OBS e não quiser modificar o código da página da web, ou seja, deseja manter o endereço do site inalterado, use a vinculação de nome de domínio definida pelo usuário.

Por exemplo, se o nome de domínio do seu site for **www.example.com** e o arquivo do site for **abc.html**, a URL para acessar esse arquivo será **http://www.example.com/abc.html**. O procedimento de configuração é o seguinte:

1. Crie um bucket em OBS e carregue o arquivo do site **abc.html** para o bucket.
2. No console de OBS, vincule o nome de domínio do site **www.example.com** ao bucket de o criado.
3. No servidor de DNS, adicione **uma regra de CNAME** e mapeie **www.example.com** para o nome de domínio do bucket.
4. Depois que a solicitação de **http://www.example.com/abc.html** chega a OBS, OBS encontra o mapeamento entre o **www.example.com** e o nome de domínio do bucket de o e o redireciona para o arquivo **abc.html** armazenado no bucket. Ou seja, OBS redireciona a solicitação para acessar **http://www.example.com/abc.html** para **http://bucket domain name/abc.html**.

## Restrições

- Somente os buckets da versão 3.0 ou posterior oferecem suporte à vinculação de nome de domínio definida pelo usuário. Para verificar a versão do bucket, vá para a página **Overview** do bucket no console de OBS. Em seguida, você pode visualizar a versão do bucket na área **Basic Information**.
- Por padrão, um máximo de 20 nomes de domínio definidos pelo usuário pode ser vinculado a um bucket.
- Atualmente, os nomes de domínio de usuário vinculados a OBS só permitem solicitações de acesso via de HTTP.

Se você quiser usar um nome de domínio de usuário para acessar OBS por HTTPS, será necessário habilitar a CDN para gerenciar certificados de HTTPS.

Para obter detalhes sobre como gerenciar certificados de HTTPS no console de gerenciamento de CDN, consulte [Configurações de HTTPS](#).

- Um nome de domínio definido pelo usuário pode ser associado a apenas um bucket.
- Atualmente, o sufixo de um nome de domínio definido pelo usuário pode conter de 2 a 6 letras maiúsculas e minúsculas.
- Conforme exigido pelo MIIT, você deve concluir o Arquivamento de ICP, se o bucket de o ao qual seu nome de domínio definido pelo usuário está vinculado estiver em qualquer uma das seguintes regiões:  
CN North-Beijing1, CN North-Beijing4, CN East-Shanghai1, CN East-Shanghai2, e CN South-Guangzhou

 **NOTA**

Se a aceleração de CDN for usada, para evitar que objetos em buckets de OBS sejam baixados diretamente após o acesso, você precisará executar as operações necessárias depois que um nome de domínio definido pelo usuário for associado. Para obter detalhes, consulte [Com a aceleração da CDN ativada, Por que os objetos no meu bucket de OBS são baixados diretamente quando eu acessá-los?](#)

## Como usar essa função

Você pode configurar a vinculação de nome de domínio definida pelo usuário usando console de OBS, as API, ou os SDK.

Ferramenta	Referência
Console de OBS	<a href="#">Configuração da vinculação de nome de domínio definido pelo usuário</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Configuração de um nome de domínio personalizado para um bucket</a>



# 17 Volta à fonte

Quando um cliente não acessa os dados solicitados no OBS, o erro 404 é retornado. No entanto, o OBS fornece a função volta à fonte para ajudá-lo a obter os dados solicitados de seu site de origem, se ele não for encontrado no OBS.

## Volta à fonte por espelhamento

Se uma regra de volta à fonte de espelhamento for configurada para um bucket de OBS e os dados solicitados não forem encontrados no bucket, o sistema recuperará os dados, quando a regra de volta à fonte se aplicar aos dados, do servidor de origem, carregá-los-á no bucket, e depois devolvê-lo ao cliente solicitante. Este processo não interrompe os serviços. Portanto, você pode usar essa função para migrar dados sem problemas do servidor de origem para OBS ou migrar serviços para o OBS sem ser detectado pelos usuários, a baixo custo. [Figura 17-1](#) ilustra o processo de espelhamento volta à fonte.

**Figura 17-1** Volta à fonte por espelhamento



### Restrições

- Atualmente, esse recurso está disponível apenas nas regiões CN North-Beijing4, CN East-Shanghai1, CN East-Shanghai2, CN South-Guangzhou, CN-Hong Kong, e AP-Singapore regions.
- Usuários anônimos não podem configurar regras de espelhamento volta à fonte para um bucket.
- Sistemas de arquivos paralelos não oferecem suporte a regras de espelhamento volta à fonte.
- Uma regra de espelhamento volta à fonte não é compatível com a função de hospedagem de sites estáticos. Especificamente, se um erro 404 ocorrer quando os objetos forem

baixados de um domínio de site estático hospedado pelo OBS, ele não acionará o processo de espelhamento de retorno à origem.

- O bucket, para o qual uma regra de retorno à origem é configurada, não pode ser especificado como o site de origem.
- Um bucket privado não pode ser especificado como o site de origem em uma regra de espelhamento volta à fonte. Altere a política do bucket para Leitura pública, se você precisar usar o bucket como o site de origem.
- O servidor de origem não pode transferir dados no modo **Transfer-Encoding: chunked**. Ou seja, a resposta à solicitação de download de um objeto do servidor de origem deve conter o cabeçalho **Content-Length** para especificar o tamanho do objeto de origem.
- Um objeto não pode corresponder a duas regras de espelhamento volta à fonte diferentes.
- Quando as regras de espelhamento e redirecionamento volta à fonte são configuradas e os dados solicitados correspondem a ambas as regras, o OBS primeiro procede de acordo com a regra de espelhamento. Se os dados solicitados não forem encontrados no site de origem, o OBS processa com a regra de redirecionamento.
- Somente os buckets da versão 3.0 ou posterior suportam a função de espelhamento volta à fonte.
- Uma regra de espelhamento volta à fonte entra em vigor cinco minutos depois de qualquer alteração na regra.
- Um máximo de 10 regras de espelhamento volta à fonte podem ser configuradas para um bucket.
- A função de espelhamento volta à fonte é oferecida gratuitamente.

#### Criação de uma regra de volta à fonte por espelhamento

Você pode criar regras de espelhamento de retorno à origem no console de OBS ou por chamar as API.

Ferramenta	Referência
Console de OBS	<a href="#">Criação de uma regra de volta à fonte</a>
API	<a href="#">Configuração de regras de volta à fonte por espelhamento</a>

# 18 Validação de URL

## Cenário de aplicação

Alguns sites desonestos podem roubar links de outros sites para enriquecer seu conteúdo sem nenhum custo. O roubo de links prejudica os interesses dos sites originais e também é uma pressão em seus servidores. OBS fornece validação de URL para resolver esse problema.

No HTTP, o campo **Referer** permite que sites e servidores da Web identifiquem de onde as pessoas estão os visitando. A validação de URL de OBS utiliza esse campo **Referer**. A ideia é que, uma vez que você descobrir que uma solicitação para o seu recurso não é originada de uma fonte autorizada (por exemplo, uma URL), você pode ter a solicitação bloqueada ou redirecionada para uma página da Web específica. Dessa forma, OBS impede o acesso não autorizado a dados armazenados em buckets.

Tal autorização é controlada usando uma lista branca e uma lista negra.

### Referer setting rules:

- O comprimento de uma lista branca ou lista negra não pode exceder caracteres 1.024.
- Formato de referência:
  - Você pode inserir vários referenciadores, cada um em uma linha.
  - O parâmetro referer suporta asteriscos (\*) e pontos de interrogação (?). Um asterisco funciona como um curinga que pode substituir zero ou vários caracteres, e um ponto de interrogação (?) pode substituir um único caractere.
  - Se o campo de cabeçalho referer contém **http** ou **https** durante o download, o referer deve conter **http** ou **https**.
- Se **Whitelisted Referers** for deixado em branco, mas **Blacklisted Referers** não for, todos os sites, exceto os especificados na lista negra, terão permissão para acessar os dados no bucket de destino.
- Se **Whitelisted Referers** não forem deixados em branco, somente os sites especificados na lista branca terão permissão para acessar o bucket de destino, independentemente de **Blacklisted Referers** serem deixados em branco ou não.

### NOTA

Se **Whitelisted Referers** estiver configurado da mesma forma que **Blacklisted Referers**, a lista negra entrará em vigor. Por exemplo, se ambas **Whitelisted Referers** e **Blacklisted Referers** estiverem definidas para **https://www.example.com**, as solicitações de acesso desse endereço serão bloqueadas.

- Se **Whitelisted Referers** e **Blacklisted Referers** forem deixadas em branco, todos os sites terão permissão para acessar dados no bucket de destino por padrão.
- Antes de determinar se um usuário tem os quatro tipos de permissões (**Read**, **Write**, **ACL View**, e **ACL Edit**) para um bucket ou objetos no bucket, verifique se esse usuário está em conformidade com os princípios de validação de URL do campo **Referer**.

#### Whitelist and blacklist setting methods:

- Configurações da lista branca

Ao definir uma lista branca, você pode permitir solicitações dos sites na lista branca, mas negar as dos sites que não estão na lista branca.

Para as solicitações que são inicializadas a partir da caixa de endereço de um navegador, você pode adicionar o campo **`\${null}** a **Referer** de **Condition** para especificar se deseja permitir as solicitações HTTP com uma **referer** em branco.

Para configurar uma lista branca, consulte a seguinte política:

```
"Statement": [
  {
    "Sid": "1",
    "Effect": "Allow",
    "Principal": {"ID": ["*"]},
    "Action": "*",
    "Resource": ["bucket/*"],
  },
  {
    "Sid": "2",
    "Effect": "Deny",
    "Principal": {"ID": ["*"]},
    "Action": ["*"],
    "Resource": ["bucket/*"],
    "Condition": {
      "StringNotEquals": {
        "Referer": ["http://www.example01.com", "${null}"]
      }
    }
  }
]
```

Se você definir uma lista de permissões dessa maneira, apenas as solicitações cuja **referer** está definida como **www.example01.com** ou em branco podem trabalhar em recursos em buckets.

- Configurações da lista negra

Para configurar uma lista negra, consulte a seguinte política:

```
"Statement": [
  {
    "Sid": "1",
    "Effect": "Deny",
    "Principal": {"ID": ["*"]},
    "Action": ["*"],
    "Resource": ["bucket/*"],
    "Condition": {
      "StringEquals": {
        "Referer": ["http://www.example01.com", "http://www.example02.com"]
      }
    }
  }
]
```

Se você definir uma lista negra dessa forma, as solicitações cuja **referer** está definida como **www.example01.com** ou **www.example02.com** não podem trabalhar em recursos em buckets.

## Como usar essa função

Você pode usar console de OBS ou as API para configurar a validação de URL.

Ferramenta	Referência
Console	<a href="#">Configuração da validação de URL</a>
As API	<a href="#">Configuração de uma lista branca da validação de URL</a>

# 19 Tags

## Cenário de aplicação

Se o seu sistema de serviços usa vários serviços em nuvem da HUAWEI CLOUD, você pode definir tags para identificar melhor as instâncias de recursos para diferentes serviços em nuvem. Para OBS, uma instância de recurso é um bucket. Essas tags serão incluídas nos registros de dados de cobrança (os CDR) gerados para esses serviços e suas instâncias. Se o seu sistema de serviço for composto de vários aplicativos, definir a mesma tag para todas as instâncias de recurso usadas para cada aplicativo ajuda a analisar facilmente o uso e os custos de recursos.

Em OBS, as tags são usadas para identificar e classificar buckets. Se você adicionar tags a um bucket, os CDR gerados para as solicitações enviadas a esse bucket incluirão essas tags, para que você possa usar as tags para classificar os CDR para análise de custo detalhada. Por exemplo, se você tiver um aplicativo que faz upload de dados de tempo de execução para um bucket, poderá marcar o bucket com o nome do aplicativo. Desta forma, os custos na aplicação podem ser analisados usando tags nos CDR.

Uma tag é descrita usando um par de chave/valor. Um bucket pode ter no máximo 10 tags. Cada tag possui apenas uma chave e um valor. A chave e o valor podem existir em qualquer sequência em uma tag. Cada chave é única entre todas as tags de um bucket, enquanto os valores podem ser repetitivos ou em branco. Demora aproximadamente três minutos para que a tag entre em vigor.

## Como usar essa função

Você pode configurar tags usando console de OBS, as API, ou os SDK.

Ferramenta	Referência
Console de OBS	<a href="#">Configuração de tags</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Configuração de tags para um bucket</a>

# 20 Inventário de bucket

---

## Cenário de aplicação

Um inventário de bucket pode listar objetos em um bucket, salvar as informações do objeto relacionado em arquivos CSV, e entregar os arquivos CSV ao bucket especificado para armazenar arquivos de inventário de bucket. Desta forma, você pode facilmente gerenciar objetos em um bucket. Um bucket de origem também pode ser o bucket de destino.

- Um arquivo de inventário de bucket pode conter as seguintes informações relacionadas a objetos: versões, tamanhos, classes de armazenamento tags, status de encriptação, e hora da última modificação.
- Você pode encriptar os arquivos de inventário do bucket no modo SSE-KMS.
- Você pode definir a frequência (diariamente ou semanalmente) para gerar arquivos de inventário de bucket.
- Você também pode especificar um bucket para armazenar os arquivos de inventário de bucket gerados.

## Restrições

- Os estoques de períodos só podem ser gerados para períodos de OBS 3.0, mas podem ser armazenados em OBS 3.0 ou em OBS 2.0.
- Um bucket pode ter no máximo 10 regras de inventário.
- O bucket de origem (para o qual uma regra de inventário de bucket é configurada) e o bucket de destino (onde os arquivos de inventário gerados são armazenados) devem pertencer à mesma conta.
- O bucket de origem e o bucket de destino devem estar na mesma região.
- Os arquivos de inventário devem estar no formato CSV.
- OBS pode gerar arquivos de inventário para todos os objetos em um bucket ou um grupo de objetos cujos nomes começam com o mesmo prefixo.
- Se um bucket tiver várias regras de inventário, essas regras não devem se sobrepor.
  - Se um bucket já tiver uma regra de inventário para todo o bucket, novas regras de inventário que filtram objetos por prefixos não poderão ser criadas. Se você precisar de uma regra de inventário que cubra apenas um subconjunto de objetos no bucket, primeiro exclua a regra de inventário configurada para o bucket inteiro.
  - Se já existir uma regra de inventário que filtra objetos por um prefixo especificado, não será possível criar uma regra de inventário para o bucket inteiro. Para criar uma

- regra de inventário para o bucket inteiro, certifique-se de que o bucket não tenha outras regras de inventário que filtrem objetos por prefixos especificados.
- Se um bucket já tiver uma regra de inventário que filtra objetos pelo prefixo de nome de objeto **ab**, o filtro de uma nova regra de inventário não poderá começar com **a** ou **abc**. Ou, você pode excluir a regra de inventário existente e criar uma nova que filtre objetos de acordo com suas necessidades.
  - Os arquivos de inventário de bucket podem ser encriptados somente no modo SSE-KMS.
  - A função de inventário de bucket é oferecida gratuitamente, mas os arquivos de inventário são cobrados pelo espaço de armazenamento usado.
  - A encriptação padrão não pode ser ativada para o bucket de destino configurado para armazenar arquivos de inventário.
  - Os arquivos de inventário são entregues ao bucket de destino por um usuário do sistema de OBS. Portanto, você precisa autorizar o usuário do sistema a permissão para gravar o bucket de destino.

## Como um inventário de bucket é configurado?

Antes da configuração, você precisa entender brevemente o que é um bucket de origem ou um bucket de destino.

- Bucket de origem: Um bucket de origem é o bucket para o qual um inventário é configurado. O inventário lista os objetos armazenados no bucket de origem.
- Bucket de destino: Um bucket de destino é onde os arquivos de inventário gerados são armazenados. Um bucket de origem também pode ser o bucket de destino. Você pode especificar um prefixo de nome para um inventário. Em seguida, os arquivos de inventário gerados serão nomeados com o prefixo e salvos no diretório com o prefixo. Se você não especificar nenhum prefixo de nome para o inventário, os arquivos de inventário gerados serão armazenados no diretório raiz do bucket.
  - Restrições no bucket de destino
    - O bucket de destino e o bucket de origem devem pertencer ao mesmo localatário.
    - O bucket de destino e o bucket de origem devem estar na mesma região.
    - A política do bucket de destino deve conceder aos usuários do sistema de OBS a permissão para gravar objetos no bucket. Para obter detalhes sobre como autorizar essa permissão, consulte [1](#).
  - O bucket de destino contém os seguintes arquivos:
    - Uma lista de arquivos de inventário
    - O arquivo **Manifest**, que contém a lista de todos os arquivos de inventário sob uma determinada configuração de inventário. Para obter detalhes sobre o arquivo **Manifest**, consulte [Arquivo de manifesto](#).

### Configuring a Bucket Inventory

Use console de OBS ou chame a API para configurar um inventário de bucket. Se você configurar um inventário de bucket no console de OBS, uma política de bucket com a configuração de permissão necessária será gerada automaticamente para o bucket de destino. Se você chamar a API para configurar o inventário do bucket, precisará configurar manualmente a política do bucket para o bucket de destino.

1. Adicione uma política de bucket para o bucket de destino.



Uma política de bucket deve ser configurada para o bucket de destino para conceder aos usuários do sistema de OBS a permissão para gravar objetos no bucket de destino. O formato da política de bucket é o seguinte. Substitua **destbucket** pelo nome real do bucket de destino.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Sid": "1",
      "Principal": {"Service": "obs"},
      "Resource": ["destbucket/*"],
      "Action": ["PutObject"]
    }
  ]
}
```

## 2. Configure um inventário de buckets.

Fornecemos várias ferramentas para configurar um inventário de buckets. Para mais detalhes, consulte [Como usar essa função](#).

## Conteúdo em um arquivo de inventário

O conteúdo em um arquivo de inventário pode ser configurado ao criar o inventário. Para obter detalhes sobre todos os campos possíveis, consulte [Tabela 20-1](#).

**Tabela 20-1** Metadados do objeto listados em um arquivo de inventário

Metadados	Descrição
Bucket	Nome do bucket de origem
Key	O nome de um objeto. Cada objeto em um bucket possui uma chave exclusiva. (Nomes de objetos no arquivo de inventário são codificados por URL usando o conjunto de caracteres UTF-8 e podem ser usados somente após serem decodificados.)
VersionId	ID de versão de um objeto. Se o valor de <b>IncludedObjectVersions</b> na configuração de inventário for <b>Current</b> , esse campo não será incluído no arquivo de inventário.
IsLatest	Se a versão do objeto for a mais recente, este parâmetro é <b>True</b> . (Se o valor de <b>IncludedObjectVersions</b> na configuração de inventário for <b>Current</b> , esse campo não será incluído no arquivo de inventário.)
IsDeleteMarker	Quando o controle de versão estiver ativado para o bucket de origem, se um objeto for excluído, um novo metadado de objeto será gerado para o objeto e o <b>IsDeleteMarker</b> dos metadados será definido como <b>true</b> . (Se o valor de <b>IncludedObjectVersions</b> na configuração de inventário for <b>Current</b> , esse campo não será incluído no arquivo de inventário.)
Size	Tamanho do objeto, em bytes.
LastModifiedDate	Data de criação do objeto ou data da última modificação

Metadados	Descrição
ETag	Resumo hexadecimal do objeto MD5. ETag é o identificador exclusivo do conteúdo do objeto. Ele pode ser usado para identificar se o conteúdo do objeto é alterado. Por exemplo, se o valor de ETag for <b>A</b> quando um objeto for carregado e o valor de ETag for alterado para <b>B</b> quando o objeto for baixado, isso indicará que o conteúdo do objeto foi alterado.
StorageClass	Classe de armazenamento de um objeto
IsMultipartUploaded	Indica se um objeto é carregado no modo multipart.
ReplicationStatus	Status de replicação entre regiões de um objeto
EncryptionStatus	Status da encriptação de um objeto

## Nome do arquivo de inventário

O nome de um arquivo de inventário está no seguinte formato:

```
destinationPrefix /sourceBucketName / inventoryId/ yyyy-MM-dd'T'HH-mm'Z' /files/  
UUID_index .csv
```

- **destinationPrefix**: O prefixo do nome do arquivo de inventário configurado ao criar a regra de inventário. Arquivos de inventário gerados sob a regra são nomeados após o prefixo, o que pode facilitar a classificação de arquivos de inventário. Se nenhum prefixo for especificado, o prefixo padrão será **BucketInventory**.
- **sourceBucketName**: Nome do bucket de origem para o qual um inventário está configurado. Esse campo pode ser usado para diferenciar arquivos de inventário de buckets de origem diferentes, se esses arquivos de inventário forem salvos no mesmo bucket de destino.
- **inventoryId**: Se um bucket de origem tiver várias regras de inventário cujos arquivos de inventário sejam salvos no mesmo bucket de destino, esse campo poderá ser usado para identificar regras de inventário diferentes.
- **yyyy-MM-dd'T'HH-mm'Z'**: Hora e data de início para a verificação do bucket de destino quando um arquivo de inventário é gerado. Os objetos carregados no bucket de origem após esse período podem não ser listados no arquivo de inventário.
- **UUID\_index.csv**: um dos arquivos de inventário

## Arquivo de manifesto

Se houver um grande número de objetos em um bucket, vários arquivos de inventário podem ser gerados para uma única configuração de inventário. Leva algum tempo para gerar esses arquivos. Por exemplo, se houver objetos 200.000 em um bucket, levará cerca de 1,5 minutos para gerar todos os arquivos de inventário. Uma ou duas horas após todos os arquivos de inventário serem gerados, um arquivo **manifest.json** será gerado. O arquivo **manifest.json** contém informações sobre todos os arquivos de inventário gerados desta vez, incluindo:

- **sourceBucket**: nome do bucket de origem
- **destinationBucket**: nome do bucket de destino

- **version**: versão do inventário
- **fileFormat**: formato dos arquivos de inventário
- **fileSchema**: campos de metadados do objeto contidos nos arquivos de inventário
- **files**: lista de todos os arquivos de inventário
- **key**: nome do arquivo de inventário
- **size**: tamanho de um arquivo de inventário, em bytes
- **inventoriedRecord**: número de registros contidos em um arquivo de inventário

O seguinte é um exemplo de um arquivo **manifest.json** simples.

```
{
  "sourceBucket": "user001",
  "destinationBucket": "bucket001",
  "version": "2019-01-03",
  "fileFormat": "CSV",

  "fileSchema": "Bucket, Key, Size, LastModifiedDate, ETag, StorageClass, IsMultipartUpload
ed, ReplicationStatus, EncryptionStatus",
  "files": [
    {
      "key": "inventory/user001/test_id/2019-01-03T12-28Z/files/
0000016813AF58E66806C1E2D7F15155_1.csv",
      "size": 6705647390,
      "inventoriedRecord": 70585762,
    }
  ]
}
```

O nome de um arquivo de **manifest** é o seguinte (para obter detalhes sobre cada campo, consulte [Nome do arquivo de inventário](#)):

```
destinationPrefix /sourceBucketName /inventoryId /yyyy-MM-dd'T'HH-mm'Z' /
manifest.json
```

## Como um usuário é notificado quando os arquivos de inventário são gerados?

Você pode habilitar SMN (o serviço de notificação de mensagens) para o bucket de destino. Ao fazer isso, você pode receber mensagens SMS ou e-mails sempre que os arquivos de inventário e o arquivo de **manifest** são gerados. Para obter mais informações sobre SMN, consulte [Notificação de evento](#).

A seguir, um exemplo simples de configuração de SMN. **destinationPrefix/sourceBucketName** indica o prefixo do arquivo de **manifest**. **destinationPrefix** é o prefixo de nome configurado para arquivos de inventário, e **sourceBucketName** é o bucket de origem para o qual o arquivo de inventário está configurado. O **manifest.json** é o sufixo do arquivo **manifest**.

```
<NotificationConfiguration>
  <TopicConfiguration>
    <Id>01</Id>
    <Filter>
      <Object>
        <FilterRule>
          <Name>prefix</Name>
          <Value>destination-prefix/source-bucket</Value>
        </FilterRule>
        <FilterRule>
          <Name>suffix</Name>
          <Value>manifest.json</Value>
        </FilterRule>
      </Object>
    </Filter>
  </TopicConfiguration>
</NotificationConfiguration>
```

```
<Topic>urn:smn:southchina:11aa22bb:notification</Topic>  
<Event>ObjectCreated:Put</Event>  
</TopicConfiguration>  
</NotificationConfiguration>
```

## Como usar essa função

Você pode configurar inventários de buckets usando console de OBS, as API, ou os SDK.

Ferramenta	Referência
Console de OBS	<a href="#">Configurar inventário de um bucket</a>
Os SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
As API	<a href="#">Configuração de inventários de bucket</a>

# 21 Notificação de evento

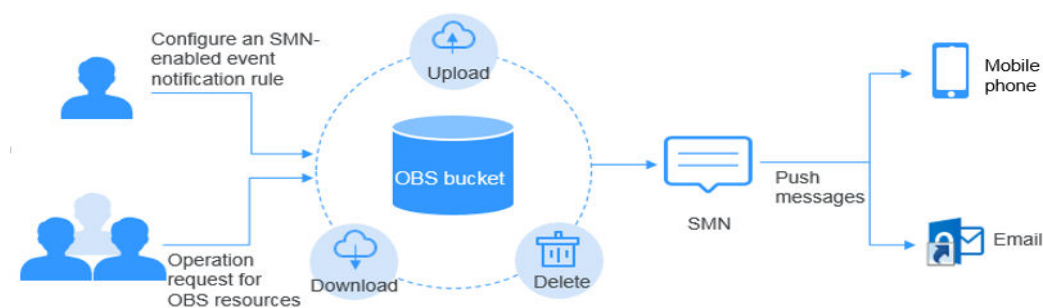
## Notificação de evento de SMN habilitado

Simple Message Notification (SMN) é um serviço de notificação de mensagens confiável e extensível que pode lidar com um grande número de mensagens. SMN simplifica significativamente o acoplamento do sistema. Ele pode enviar mensagens automaticamente para os assinantes através de e-mails e mensagens de texto.

OBS usa SMN para fornecer a função de notificação de eventos. Você pode usar SMN para enviar notificações de eventos aos assinantes especificados para informá-los em tempo real de operações críticas (como carregamento e exclusão) que ocorrem em buckets especificados. Por exemplo, você pode configurar uma regra de notificação de eventos para enviar mensagens por meio de SMN para o endereço de e-mail especificado sempre que ocorrer uma operação de carregamento no bucket especificado.

Você pode configurar uma regra de notificação de eventos para filtrar objetos pelo prefixo ou sufixo do nome do objeto. Por exemplo, você pode adicionar uma regra de notificação de eventos para enviar notificações sempre que um objeto com o sufixo **.jpg** for carregado no bucket especificado. Você também pode adicionar uma regra de notificação de eventos para enviar notificações sempre que um objeto com o prefixo **images/** for carregado no bucket especificado.

Figura 21-1 Notificação de evento de SMN habilitado



## Tipos de eventos suportados por OBS

OBS pode publicar eventos dos seguintes tipos. Você precisa especificar esses tipos de evento nas configurações de notificação.

**Tabela 21-1** Tipos de eventos suportados por OBS

Tipo de evento	Descrição
<p><b>ObjectCreated:*</b> (todas as operações de carregamento)</p> <p><b>ObjectCreated:Put</b> (carregamento de objeto)</p> <p><b>ObjectCreated:Post</b> (envio de objetos através de um navegador web)</p> <p><b>ObjectCreated:Copy</b> (cópia de objetos)</p> <p><b>ObjectCreated:CompleteMultipartUpload</b> (mescla de peças carregadas)</p>	<p>OBS pode usar as API como PUT, POST e COPY para criar objetos e configurar tipos de eventos correspondentes. Você receberá uma notificação quando um objeto for criado usando uma API específica. Além disso, você pode usar o tipo de evento <b>ObjectCreated:*</b> para solicitar todas as notificações de criação de objetos.</p> <p><b>NOTA</b>                  Você não receberá notificações de eventos de operações com falha.</p>
<p><b>ObjectRemoved:*</b> (todas as operações de eliminação)</p> <p><b>ObjectRemoved&gt;Delete</b> (excluindo objetos)</p> <p><b>ObjectRemoved&gt;DeleteMarkerCreated</b> (Um objeto deletemarker é criado.)</p>	<p>Usando os tipos de evento <b>ObjectRemoved</b>, você pode ativar a notificação quando um objeto ou um lote de objetos são removidos de um bucket.</p> <p>Você pode solicitar uma notificação quando um objeto é excluído ou um objeto versionado é excluído permanentemente usando o tipo de evento <b>ObjectRemoved&gt;Delete</b>. Como alternativa, você pode solicitar uma notificação quando um marcador de exclusão for criado para um objeto versionado usando <b>ObjectRemoved&gt;Delete</b>. Você também pode usar <b>ObjectRemoved:*</b> para solicitar uma notificação cada vez que um objeto é excluído.</p> <p><b>NOTA</b>                  Você não receberá notificações de eventos de exclusões automáticas de políticas de ciclo de vida ou de operações com falha.</p>

## Restrições

- Uma configuração sem atributos de filtragem contidos corresponde a todos os prefixos e sufixos por padrão.
- Quando há dois sufixos configurados, se uma dada cadeia de caracteres pode terminar com ambos os sufixos, esta configuração é inválida devido à sobreposição de sufixos (por exemplo, sufixos jpg e pg). Isso também funciona para prefixos.

## Destinos suportados por OBS

OBS pode enviar mensagens de notificação de eventos para tópicos de [SMN](#). Tem de conceder a OBS as permissões para enviar mensagens para estes destinos. Além disso, você precisa especificar os valores URN desses destinos na configuração de notificação.

## Como ativar as notificações de eventos

A ativação de notificações é uma operação no nível do bucket. OBS armazena sua configuração de notificação de eventos em sub-recursos de bucket no formato XML. Por padrão, a notificação não está habilitada para nenhum tipo de evento. Ou seja, a configuração inicial de notificação de evento de cada bucket está vazia.

Para ativar notificações para eventos de tipos específicos, você deve adicionar a configuração XML correspondente que identifica os tipos de evento que você deseja que OBS publique e o destino onde você deseja que as notificações sejam publicadas.

- Neste exemplo, você precisa publicar mensagens de evento em um tópico de SMN. Para definir um tópico SMN como o destino de notificação para tipos de evento específicos, adicione o **TopicConfiguration**.

```
<NotificationConfiguration>
  <TopicConfiguration>
    <Id>optional-id-string</Id>
    <Topic>topic-urn</Topic>
    <Event>event-type</Event>
    <Event>event-type</Event>
    ...
  </TopicConfiguration>
  ...
</NotificationConfiguration>
```

Para remover todas as configurações de notificação de um bucket, defina o elemento **<NotificationConfiguration>** como nulo.

## Uso de nomes de chave de objeto para filtrar notificações de evento

Você pode definir a notificação de eventos pelo prefixo e sufixo de um nome de chave de objeto. Por exemplo, você pode definir uma configuração para que uma notificação seja publicada somente quando objetos com uma extensão **.jpg** forem adicionados a um bucket.

OBS armazena a configuração de notificação como XML. Você pode usar o elemento **Filter** na estrutura XML para definir as regras para que as notificações sejam filtradas pelo prefixo e/ou sufixo de um nome de chave de objeto. As configurações de notificação que usam o filtro não podem definir regras de filtragem com prefixos sobrepostos, sufixos sobrepostos ou sobreposição de prefixos e sufixos. A seguir estão exemplos de configurações de notificação com filtragem de nome de chave de objeto:

- **Exemplo de configuração de notificação válida com filtragem de nome de chave de objeto**

As informações a seguir contêm uma configuração que identifica um tópico de SMN no qual OBS publica eventos do tipo **ObjectCreated:Put**. Os eventos serão publicados cada vez que um objeto que tem um prefixo de **image** e um sufixo **jpg** for PUT para um bucket.

```
<NotificationConfiguration>
  <TopicConfiguration>
    <Id>01</Id>
    <Filter>
      <Object>
        <FilterRule>
          <Name>prefix</Name>
          <Value>image</Value>
        </FilterRule>
        <FilterRule>
          <Name>suffix</Name>
          <Value>jpg</Value>
        </FilterRule>
      </Object>
    </Filter>
  </TopicConfiguration>
</NotificationConfiguration>
```

```
</Object>
</Filter>
<Topic>urn:smn:southchina:11aa22bb:notification</Topic>
<Event>ObjectCreated:Put</Event>
</TopicConfiguration>
</NotificationConfiguration>
```

A configuração de notificação a seguir tem vários prefixos não sobrepostos. A configuração define o seguinte: Quando os objetos que têm um prefixo de **images** são carregados em buckets, as notificações de eventos são publicadas no tópico-A; quando os objetos que têm um prefixo de **videos** são carregados em buckets, as notificações de eventos são publicadas no tópico-B.

```
<NotificationConfiguration>
  <TopicConfiguration>
    <Id>01</Id>
    <Filter>
      <Object>
        <FilterRule>
          <Name>prefix</Name>
          <Value>images</Value>
        </FilterRule>
      </Object>
    </Filter>
    <Topic>urn:smn:southchina:11aa22bb:topic-A</Topic>
    <Event>ObjectCreated:Put</Event>
  </TopicConfiguration>
  <TopicConfiguration>
    <Id>02</Id>
    <Filter>
      <Object>
        <FilterRule>
          <Name>prefix</Name>
          <Value>videos</Value>
        </FilterRule>
      </Object>
    </Filter>
    <Topic>urn:smn:southchina:11aa22bb:topic-B</Topic>
    <Event>ObjectCreated:Put</Event>
  </TopicConfiguration>
</NotificationConfiguration>
```

A configuração de notificação a seguir tem vários sufixos não sobrepostos. A configuração define o seguinte: As notificações serão publicadas no tópico-A para todos os objetos **.jpg** PUT para buckets e as notificações serão publicadas no tópico-B para todos os objetos **.png**. Os sufixos **.png** e **.jpg** não são sobrepostos, apesar de terem a mesma última letra.

```
<NotificationConfiguration>
  <TopicConfiguration>
    <Id>01</Id>
    <Filter>
      <Object>
        <FilterRule>
          <Name>suffix</Name>
          <Value>.jpg</Value>
        </FilterRule>
      </Object>
    </Filter>
    <Topic>urn:smn:southchina:11aa22bb:topic-A</Topic>
    <Event>ObjectCreated:Put</Event>
  </TopicConfiguration>
  <TopicConfiguration>
    <Id>02</Id>
    <Filter>
      <Object>
        <FilterRule>
          <Name>suffix</Name>
          <Value>.png</Value>
        </FilterRule>
      </Object>
    </Filter>
    <Topic>urn:smn:southchina:11aa22bb:topic-B</Topic>
    <Event>ObjectCreated:Put</Event>
  </TopicConfiguration>
</NotificationConfiguration>
```



```
</FilterRule>
</Object>
</Filter>
<Topic>urn:smn:southchina:11aa22bb:topic-B</Topic>
<Event>ObjectCreated:Put</Event>
</TopicConfiguration>
</NotificationConfiguration>
```

- **Exemplos de configuração de notificação com sobreposição de prefixo/sufixo inválido**

Na maioria dos casos, as configurações de notificação que usam o filtro não podem definir regras de filtragem com prefixos sobrepostos, sufixos sobrepostos ou combinações sobrepostas de prefixos e sufixos para os mesmos tipos de evento. (Você pode ter prefixos sobrepostos se os sufixos não se sobrepõem.) Você pode usar regras de filtragem de nomes de chave de objeto sobrepostos com diferentes tipos de evento. Por exemplo, você pode criar uma configuração de notificação que usa o prefixo de **image** para o tipo de evento **ObjectCreated:Put** e o prefixo de **image** para o tipo de evento **ObjectDeleted:\***.

Uma configuração sem atributos de filtragem contidos corresponde a todos os prefixos e sufixos por padrão. A configuração de notificação a seguir é inválida porque contém prefixos sobrepostos. (A mesma coisa seria verdadeira se **suffix** em vez de **prefix** é usado neste exemplo.)

```
<NotificationConfiguration>
  <TopicConfiguration>
    <Topic>urn:smn:southchina:11aa22bb:topic-A</Topic>
    <Event>ObjectCreated:*</Event>
  </TopicConfiguration>
  <TopicConfiguration>
    <Filter>
      <Object>
        <FilterRule>
          <Name>prefix</Name>
          <Value>abc</Value>
        </FilterRule>
      </Object>
    </Filter>
    <Topic>urn:smn:southchina:11aa22bb:topic-B</Topic>
    <Event>ObjectCreated:*</Event>
  </TopicConfiguration>
</NotificationConfiguration>
```

A seguinte configuração de notificação é inválida porque contém sufixos sobrepostos. Dois sufixos são considerados sobrepostos se uma dada cadeia pode terminar com ambos os sufixos. Uma cadeia pode terminar com **jpg** e **pg**. Portanto, os sufixos são sobrepostos. (O mesmo vale para os prefixos.)

```
<NotificationConfiguration>
  <TopicConfiguration>
    <Filter>
      <Object>
        <FilterRule>
          <Name>suffix</Name>
          <Value>jpg</Value>
        </FilterRule>
      </Object>
    </Filter>
    <Topic>urn:smn:southchina:11aa22bb:topic-A</Topic>
    <Event>ObjectCreated:*</Event>
  </TopicConfiguration>
  <TopicConfiguration>
    <Filter>
      <Object>
        <FilterRule>
          <Name>suffix</Name>
          <Value>pg</Value>
        </FilterRule>
      </Object>
    </Filter>
    <Topic>urn:smn:southchina:11aa22bb:topic-B</Topic>
    <Event>ObjectCreated:*</Event>
  </TopicConfiguration>
</NotificationConfiguration>
```

```
</FilterRule>
</Object>
</Filter>
<Topic>urn:smn:southchina:11aa22bb:topic-B</Topic>
<Event>ObjectCreated:Put</Event>
</TopicConfiguration>
</NotificationConfiguration>
```

## Estrutura da mensagem de evento

Uma mensagem de notificação enviada por OBS para publicar um evento é uma mensagem JSON com a seguinte estrutura.

```
{
  "Records": [
    {
      "eventVersion": "", //Version number. The current version is 3.0.
      "eventSource": "", //Message source. The value is fixed to OBS.
      "eventRegion": "", //Region where the event occurs
      "eventTime": "", //Time when an event occurs, in the ISO-8601 format,
      for example, 2020-07-10T09:24:11.418Z
      "eventName": "", //Name of the event that triggers the notification
      "userIdentity": {
        "ID": "" //Billing ID of the user who triggers the event
      },
      "requestParameters": {
        "sourceIPAddress": "" //Source IP address of the request
      },
      "responseElements": {
        "x-obs-request-id": "", //ID of the request
        "x-obs-id-2": "" ///Special characters for locating problems
      },
      "obs": {
        "Version": "1.0",
        "configurationId": "", //Name of the event notification rule in
        OBS that matches the event
        "bucket": {
          "name": "examplebucket",
          "ownerIdentity": {
            "ID": "" //Account ID of the bucket owner
          },
          "bucket": "" //Bucket name
        },
        "object": {
          "key": "", //Object name
          "eTag": "", //ETag of the object
          "size": , //Object size
          "versionId": "nulo", //Version ID of the object
          "sequencer": "" //Identifier that defines the event sequence
        }
      }
    }
  ]
}
```

Observe o seguinte:

- Você pode usar o valor-chave **responseElements** para rastrear solicitações de OBS. Ambos **x-obs-request-id** e **x-obs-id-2** podem ser usados para rastrear uma única requisição. Seu valor é aquele retornado por OBS na resposta à solicitação.
- O **obs** chave-valor contém informações sobre o bucket e o objeto envolvidos no evento. Observe que o nome da chave do objeto é codificado por URL. Por exemplo, **TEST/Chinese.jpg** é alterado para **TEST%2F%E4%B8%AD+%E6%96%87%2F.jpg**.

**AVISO**

Durante o desenvolvimento secundário, se você usar o SDK de OBS para baixar o objeto, precisará decodificar o nome do objeto codificado por URL e chamar a API para download. Se um navegador da web for usado para acessar o objeto, a decodificação não será necessária.

- O **sequencer** chave-valor determina a seqüência de eventos. Geralmente, as notificações de eventos não chegam na ordem em que os eventos ocorreram. No entanto, as notificações de eventos para criação de objetos (PUT) e exclusão de objetos (DELETE) contêm um **sequencer**, que pode ser usado para determinar a ordem dos eventos de uma determinada chave de objeto. Se você comparar as cadeias de sequenciador hexadecimal de duas notificações de evento na mesma chave de objeto, a notificação de evento com o maior valor de sequenciador é o evento que ocorreu mais tarde.

**AVISO**

1. O sequenciador não pode ser usado para determinar a seqüência de eventos em diferentes chaves de objeto.
2. A seqüência de eventos indicada pelo **sequencer** é apenas para referência, o que não funciona para sistemas altamente confiáveis

Exemplo de mensagem:

- Test message - When you configure an event notification on a bucket, OBS sends the following test message:

```
{
  "Service": "OBS",
  "Event": "TestEvent",
  "Time": "1970-01-01T00:00:00.000Z",
  "Bucket": "examplebucketname",
  "RequestId": "0002F4BCF6000001563B064B17B2094D",
  "HostId": "2Zf+b9AmbaBgNQ+YE8XU2j87DZaBNxu4TaMiOCTqpmkC2SA9ouf8TpB2SY5j3i4"
}
```

- Example message when an object is created using the PUT request - The following is an example of a message sent by OBS to publish an **ObjectCreated:Put** event:

```
{
  "Records": [
    {
      "eventVersion": "3.0",
      "eventSource": "OBS",
      "eventRegion": "cn-east-3",
      "eventRegion": "region",
      "eventTime": "2018-06-26T14:37:05.468Z",
      "eventName": "ObjectCreated:Put",
      "userIdentity": {
        "ID": "71f3901173514e6988115ea2c26d1999"
      },
      "requestParameters": {
        "sourceIPAddress": "104.55.173.69"
      },
      "responseElements": {
        "x-obs-request-id": "9006000001643C86D03C300BE8860FA7",
        "x-obs-id-2": "2+/Ucr6uinCJAbUejWyQ+rhxkuF/K/9uoaXuewIi/SE9j4tU5LwaXTTlD1gvMv2o"
      },
      "obs": {
        "Version": "1.0",

```

```
"configurationId": "ConfigurationId",
"bucket": {
  "name": "examplebucket",
  "ownerIdentity": {
    "ID": "b4bf1b36d9ca43d984fbc9491b6fce9"
  },
  "bucket": "examplebucket"
},
"object": {
  "key": "object0001.txt",
  "eTag": "3b9680702b9a12733c5490d1b15c7607",
  "size": 538,
  "versionId": "null",
  "sequencer": "000000001643C86D06576F5320000000"
}
}
]
```

 **NOTA**

Devido a problemas de Internet ou restrições no envio de e-mail, as notificações podem não ser enviadas para terminais HTTP ou de e-mail.

## Como usar essa função

Você pode configurar notificações de SMN usando console de OBS, as API, ou os SDK.

Ferramenta	Referência
SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
API	<a href="#">Configuração da notificação de evento para um bucket</a>

# 22 Registro

## Cenário de aplicação

Você pode habilitar o registro em registro para facilitar a análise ou a auditoria, conforme necessário. Os logs de acesso permitem que o proprietário de um bucket analise a propriedade, o tipo ou a tendência das solicitações para o bucket em profundidade. Com o registro ativado, OBS registra automaticamente as solicitações de acesso para o bucket e grava os arquivos de registro gerados no bucket especificado.

Você precisa especificar um bucket para armazenar arquivos de registro ao ativar o registro para um bucket. Os arquivos de registro podem ser armazenados em qualquer bucket na região onde o bucket registrado reside, incluindo o próprio bucket registrado. Para gerenciar melhor os logs, é aconselhável armazenar os arquivos de registro em um bucket diferente do bucket registrado. Se os arquivos de registro forem armazenados no bucket de registro, OBS criará logs adicionais para gravar arquivos de registro no bucket, o que ocupará espaço de armazenamento extra, o que aumentará seus custos e dificultará a localização dos logs necessários.

### AVISO

O upload de logs do bucket para o bucket de destino incorre em custos para solicitações PUT. Para obter detalhes sobre os preços de solicitação PUT, consulte [Cobrança de OBS](#).

OBS pode registrar solicitações de acesso ao bucket em logs para análise de solicitações e auditoria de logs.

Logs ocupam algum espaço de armazenamento OBS alugado pelos usuários, incorrendo em taxas extras. Por esse motivo, a política padrão é que OBS não coleta logs de acesso ao bucket.

The log files are generated and uploaded by OBS to the bucket where the logs are stored. Therefore, OBS requires the authorization to upload the generated log files. Therefore, before configuring logging for a bucket, you need to create an IAM agency for OBS and add this IAM agency when configuring logging for the bucket. By default, when configuring permissions for an IAM agency, you only need to grant the IAM agency the permission to upload log files to the bucket where log files are stored. In the following example, **mybucketlogs** is the name of the bucket for storing log files. If the default encryption function is enabled for the log storing bucket, the IAM agency also requires the KMS Administrator permissions in the region where the log storing bucket resides.

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "obs:object:PutObject"
      ],
      "Resource": [
        "OBS:*:*:object:mybucketlogs/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Depois que o registro em registro é configurado, você pode ver os registros de operação no bucket que armazena os registros em aproximadamente quinze minutos.

O seguinte mostra um registro de registro de acesso de exemplo:

```
787f2f92b20943998a4fe2ab75eb09b8 bucket [13/Aug/2015:01:43:42 +0000] xx.xx.xx.xx
787f2f92b20943998a4fe2ab75eb09b8 281599BACAD9376ECE141B842B94535B
REST.GET.BUCKET.LOCATION
- "GET /bucket?location HTTP/1.1" 200 - 211 - 6 6 "-" "HttpClient" - -
```

O registro de acesso de cada bucket contém as seguintes informações.

**Tabela 22-1** Formato dos logs de acesso ao bucket

Nome	Exemplo	Descrição
BucketOwner	787f2f92b20943998a4fe2ab75eb09b8	ID da conta do proprietário do bucket
Bucket	Bucket	Nome do bucket
Horário	[13/Ago/2015:01:43:42 +0000]	Carimbo de data e hora da solicitação (UTC)
IP remoto	xx.xx.xx.xx	Solicitar endereço IP
Solicitante	787f2f92b20943998a4fe2ab75eb09b8	ID do solicitante <ul style="list-style-type: none"> <li>● Quando uma conta inicia uma solicitação, essa é a ID da conta. Quando um usuário do IAM inicia uma solicitação, esse é a ID da conta à qual o usuário do IAM pertence.</li> <li>● Quando uma solicitação é iniciada por um usuário anônimo, o valor desse parâmetro é <b>Anonymous</b>.</li> </ul>
SolicitaçãoID	281599BACAD9376ECE141B842B94535B	ID de Solicitação
Operação	REST.GET.BUCKET.LOCATION	Operação

Nome	Exemplo	Descrição
Chave	-	Nome do Objeto
Solicitação-URI	GET /bucket?localização HTTP/1.1	Solicitar URI
HTTPStatus	200	Código de resposta
ErrorCode	-	Código de erro
BytesSent	211	Tamanho da resposta HTTP, expresso em bytes
ObjectSize	-	Tamanho do objeto (bytes)
TotalTime	6	Tempo de processamento no servidor (ms)
Turn-AroundTime	6	Tempo total para processar a solicitação (ms)
Encaminhado por	-	Cabeçalho de referência da solicitação
Usuário-Agente	HttpClient	Cabeçalho User-Agent da solicitação
ID da versão	-	ID de versão contida em uma solicitação
STSLogUrn	-	Autenticação federada e informações de agência
StorageClass	STANDARD_IA	Classe de armazenamento atual do objeto
TargetStorageClass	GLACIER	Classe de armazenamento para a qual o objeto será transitado
DentryName	12456/file.txt	<ul style="list-style-type: none"><li>● Para um sistema de arquivos paralelo, este campo indica um identificador interno de um arquivo ou diretório. Seu valor consiste em um número de inode de diretório pai e um nome de arquivo ou diretório.</li><li>● Para um bucket, o valor deste campo é -.</li></ul>

## Como usar essa função

Você pode configurar o registro em registro no console de OBS, usando as API, ou os SDK.

Ferramenta	Referência
Console de OBS	<a href="#">Configuração da registro de acesso para um bucket</a>
SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
API	<a href="#">Configuração da registro para um bucket</a>



# 23 Versionamento

## Cenário de aplicação

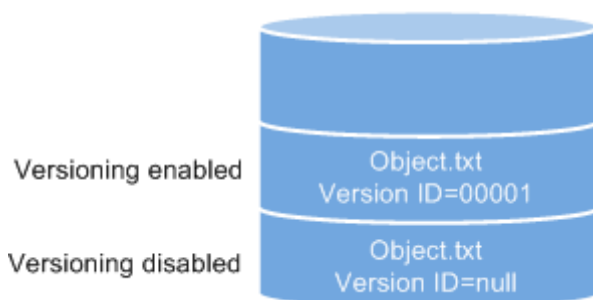
OBS consegue armazenar várias versões de um objeto. Você pode pesquisar e restaurar versões diferentes rapidamente ou restaurar dados em caso de exclusões acidentais ou falhas de aplicações.

Por padrão, o versionamento é desativado para novos buckets de OBS. Novos objetos irão sobrescrever objetos existentes no caso de terem os mesmos nomes.

## Ativação de versionamento

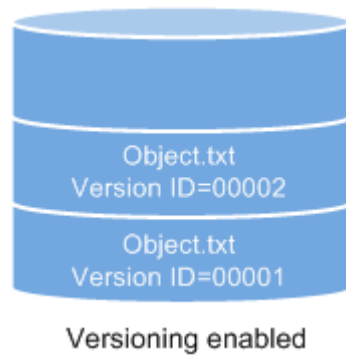
- A ativação de versionamento não altera as versões ou o conteúdo de objetos existentes no bucket. O ID de versão de um objeto é nulo antes da ativação de versionamento. Se um objeto homônimo for carregado após a ativação de versionamento, um ID de versão será atribuído ao objeto. Para mais detalhes, consulte [Figura 23-1](#).

**Figura 23-1** Controle de versões (habilitado vs. desabilitado)



- Com versionamento ativado, o OBS aloca automaticamente um ID de versão exclusivo para um objeto recém-carregado. Quando um objeto com o mesmo nome de um objeto existente é carregado novamente, ambos os objetos são armazenados em OBS com o mesmo nome, mas os ID de versões diferentes. Para mais detalhes, consulte [Figura 23-2](#).

**Figura 23-2** Versionamento (os ID de versões diferentes para objetos homônimos)



**Tabela 23-1** Descrição da versão

Versão	Descrição
Última versão	Depois que versionamento for ativado, cada operação em um objeto resultará no salvamento do objeto com um novo ID de versão. O ID de versão gerado na operação mais recente é chamado de versão mais recente.
Versão do histórico	Depois que versionamento for ativado, cada operação em um objeto resultará no salvamento do objeto com um novo ID de versão. Os ID de versão gerados em operações diferentes da operação mais recente são chamados de versões históricas.

- Os objetos mais recentes em um bucket são retornados por padrão após uma solicitação GET object.
- Os objetos podem ser baixados por os ID de versão. Por padrão, o último objeto será baixado se o ID da versão não for especificado.
- Você pode selecionar um objeto e clicar em **Delete** à direita para excluir o objeto. Após a exclusão do objeto, OBS gera a **Delete Marker** com um ID de versão exclusivo para o objeto excluído, e o objeto excluído é exibido na lista **Deleted Objects**. Se você tentar acessar o objeto excluído, um erro 404 será retornado.

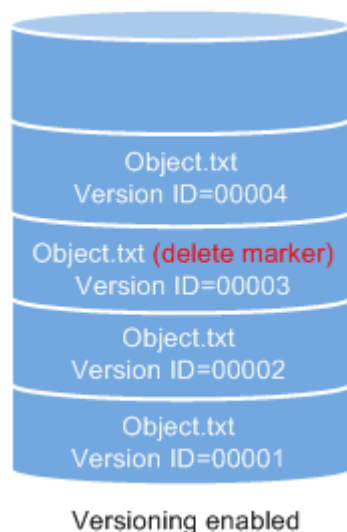
**Figura 23-3** Objeto com um marcador de exclusão



- Você pode recuperar um objeto excluído, excluindo a versão do objeto que possui o **Delete Marker**.
- Depois que um objeto é excluído, você pode especificar o número da versão em **Deleted Objects** para excluir permanentemente o objeto da versão especificada.
- Um objeto é exibido na lista de objetos ou na lista de objetos excluídos. Ele nunca será exibido nas duas listas ao mesmo tempo.

Por exemplo, após o carregamento e a exclusão do objeto **A**, ele será exibido na lista **Deleted Objects**. Se você enviar um objeto chamado **A** novamente, o objeto **A** será exibido na lista **Objects**, e o objeto **A** excluído anteriormente não será mais exibido na lista **Deleted Objects**. Para mais detalhes, consulte [Figura 23-4](#).

**Figura 23-4** Carregando um objeto homônimo após a exclusão do original

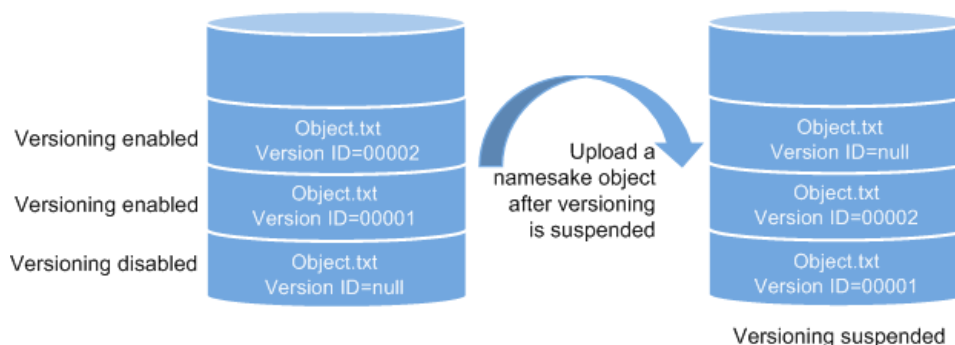


- Todas as versões de objetos, exceto aquelas com o **Delete Marker** armazenado em OBS, são cobradas.

## Suspensão de versionamento

Uma vez ativada a função de versionamento, ela pode ser suspensa, mas não pode ser desativada. Depois que o versionamento for suspenso, os IDs de versão não serão mais alocados a objetos recém-carregados. Se um objeto com o mesmo nome já existir e não tiver um ID de versão, ele será substituído.

**Figura 23-5** Versões de objeto no cenário quando o versionamento é suspenso



Se as versões de objetos em um bucket não precisarem ser controladas, você poderá suspender a função de versionamento.

- As versões históricas serão mantidas em OBS. Se você não precisar dessas versões históricas, exclua-as manualmente.
- Os objetos podem ser baixados por os ID de versão. Por padrão, o último objeto será baixado se o ID da versão não for especificado.
- Todas as versões de objetos históricos, exceto aquelas com o **Delete Marker** armazenado em OBS, são cobradas.

## Diferenças entre cenários quando versionamento é suspenso e desativado

Se você excluir um objeto quando versionamento estiver suspenso, será gerada uma versão nula com o **Delete Marker**, independentemente de o objeto ter versões históricas. Porém, se versionamento não estiver ativado, a mesma operação não gerará uma versão com o **Delete Marker**.

### NOTA

Depois que versionamento é ativado, cada versão histórica de um objeto é armazenada e ocupa espaço de armazenamento. OBS cobra taxas de armazenamento para todas as versões. Tenha cuidado para evitar taxas extras de armazenamento.

## Como usar essa função

Você pode configurar versionamento por meio do console de OBS, as API, e os SDK.

Ferramenta	Referência
Console de OBS	<a href="#">Configuração de versionamento</a>
SDK	OBS suporta kits de desenvolvimento de software (os SDK) em vários idiomas. Para obter detalhes, consulte o guia do desenvolvedor correspondente na página <a href="#">Visão geral de SDK</a> .
API	<a href="#">Configuração de versionamento para um bucket</a>

# 24 Monitoramento

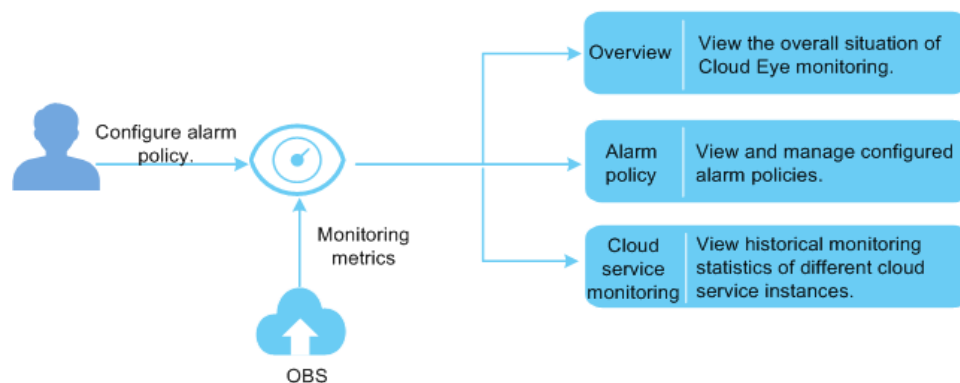
## Cenário de aplicação

Você pode enviar solicitações PUT e GET continuamente ao usar OBS, o que gera tráfego de carregamento e download. Você também pode receber respostas de erro do servidor. HUAWEI CLOUD fornece o serviço Cloud Eye para ajudá-lo a monitorar OBS e entender melhor os status do seu bucket. O Cloud Eye pode realizar monitoramento automático e em tempo real sobre seus buckets. Ele aciona alarmes e notificações sobre as operações do usuário com base em políticas predefinidas, permitindo que você fique de olho nas solicitações de acesso ao bucket, no tráfego e nas respostas a erros.

Você não precisa assinar separadamente o serviço Cloud Eye. Ele é iniciado automaticamente quando você cria um recurso (um bucket, por exemplo) em OBS.

Para obter mais informações sobre o Cloud Eye, consulte [O que é o Cloud Eye](#).

**Figura 24-1** Monitoramento do Cloud Eye



## Configuração de regras de alarme

Além do monitoramento automático e em tempo real, você pode configurar regras de alarme no Cloud Eye para que notificações de alarme sejam enviadas quando situações específicas ocorrem.

Para obter detalhes sobre como configurar uma regra de alarme para o monitoramento do Cloud Eye por OBS, consulte [Criação de uma regra de alarme](#).

## Visualização de métricas de monitoramento de OBS

O Cloud Eye monitora o uso de OBS em tempo real. Você pode ver estatísticas detalhadas de cada métrica no Console do Cloud Eye.

Para obter detalhes sobre como exibir métricas de monitoramento de OBS, consulte [Consulta de métricas de monitoramento de serviço Cloud](#).

## Métricas de monitoramento

Para obter detalhes, consulte [Métricas de monitoramento de OBS](#).

# 25 Auditoria

## Cenário de aplicação

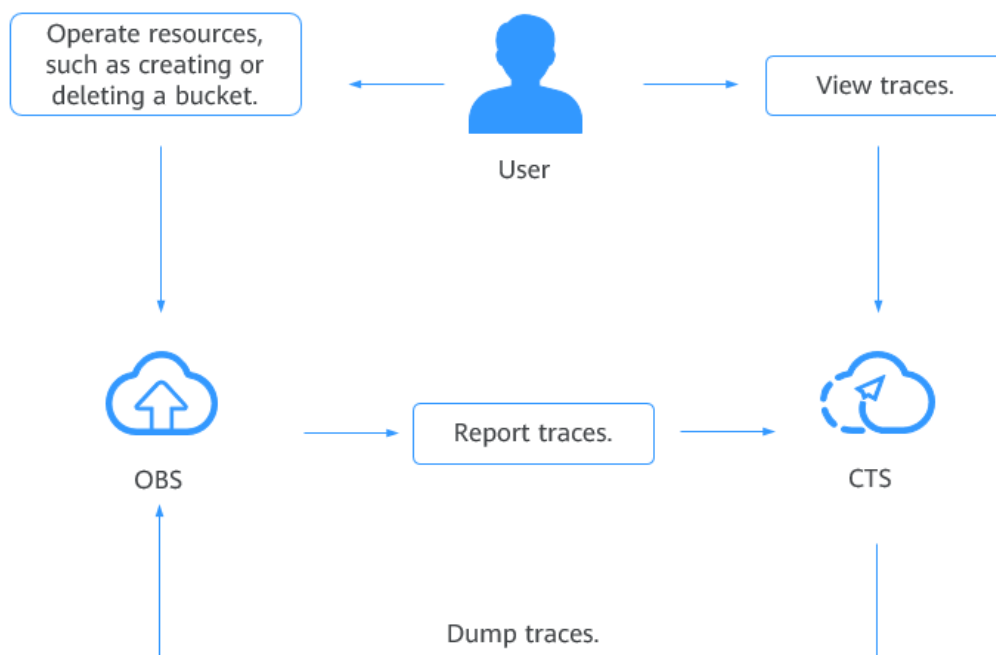
Cloud Trace Service (CTS) registra as operações nos recursos da nuvem em sua conta. Você pode usar os logs gerados por CTS para realizar análises de segurança, rastrear alterações de recursos, auditar conformidade, e localizar falhas.

Depois de ativar CTS e configurar um rastreador, CTS pode registrar rastreamentos de gerenciamento e dados de OBS para auditoria.

Para obter detalhes sobre como habilitar e configurar CTS, consulte [Habilitação de CTS](#).

Para obter detalhes sobre o gerenciamento de OBS e os rastreamentos de dados que podem ser rastreados por CTS, consulte [Cloud Trace Service](#).

**Figura 25-1** Cloud Trace Service



# 26 Descompressão online (OBT)

---

Você pode compactar vários arquivos em um pacote ZIP e enviá-lo para o OBS.

O OBS descompacta automaticamente os pacotes ZIP depois que eles são carregados.

## Cenário de aplicação

- Carregue um pacote ZIP em um bucket do OBS e faça com que o pacote seja automaticamente descompactado e armazenado em um diretório específico.
- Faça upload de um grande número de pequenos arquivos por vez usando um pacote ZIP para economizar tempo e esforços. Para restrições de upload em lote, consulte [Carregamento em lote de OBS](#).

## Pré-requisitos

Você foi atribuída a função **Tenant Administrator**.

## Restrições

- Atualmente, a descompressão online está disponível apenas nas regiões CN North-Beijing4, CN South-Guangzhou, e CN East-Shanghai2.
- Atualmente, apenas pacotes ZIP são suportados.
- Um único pacote ZIP não pode exceder 1 GB de tamanho.
- Um único pacote ZIP pode conter um máximo de arquivos 65.536.
- Um único arquivo descompactado não pode exceder 40 GB de tamanho.
- A descompactação de um pacote ZIP leva no máximo 10 minutos.
- Os nomes de pacotes ZIP não podem conter sinais de pontuação em chinês, caracteres especiais ou códigos especiais.
- Para descompactar o pacote ZIP que contém outros pacotes ZIP, o tipo de evento da política de descompactação online deve ser **ObjectCreated:\*** ou **ObjectCreated:CompleteMultipartUpload**.
- No momento, nenhuma notificação será enviada aos usuários após a conclusão das tarefas de descompactação.
- Atualmente, somente pacotes ZIP deflate-comprimidos podem ser descomprimidos. Deflate é diferente dos algoritmos de compressão (como Store e Normal) fornecidos pelo WinRAR.



- O comprimento total do caminho de descompactação mais o nome de qualquer arquivo descompactado não pode exceder 1024 caracteres.
- Atualmente, os pacotes ZIP criptografados não podem ser descompactados.

## Precauções

- É aconselhável definir um prefixo preciso para uma política de descompressão. No mesmo bucket, as condições de disparo (incluindo eventos, prefixos ou sufixos) de diferentes políticas de descompressão não podem se sobrepor.  
Por exemplo, há duas políticas de descompactação **event-0001** e **event-0002** em um bucket. Se o prefixo de **event-0001** for **aa**, o prefixo de **event-0002** não pode ser **aaaa**, porque **aa** está contido em **aaaa**.
- Se o prefixo for deixado em branco, a política de descompactação se aplicará a todos os pacotes ZIP no bucket por padrão. Isso pode desencadear a descompressão cíclica se um pacote contiver outros pacotes.  
Por exemplo, o pacote **AA.zip** contém outro pacote **BB.zip**. Se o prefixo for deixado em branco, o sistema continuará a descompactar **BB.zip** após a descompactação do **AA.zip**. Esse problema não acontecerá se um prefixo for definido na política.
- Você deve definir um diretório para armazenar os arquivos descompactados. Se o diretório não estiver definido, os arquivos descompactados serão armazenados no diretório base do bucket atual.
- É aconselhável codificar nomes de arquivos ou pastas usando UTF-8. Caso contrário, os nomes dos arquivos ou pastas descompactados podem conter caracteres ilegíveis ou a descompactação pode ser interrompida.
- Se desejar descompactar pacotes ZIP na classe de armazenamento Arquivo, verifique se a Leitura direta foi ativada para o bucket.
- Uma descompactação de pacotes ZIP falhará se demorar mais de 10 minutos.

## Como usar essa função

Configure as políticas para descompactar pacotes ZIP online por meio do console de OBS ou das API.

Ferramenta	Referência
Console de OBS	<a href="#">Criação de uma política de descompressão online</a>
API	<a href="#">Configuração de uma política de descompactação online</a>

---

# 27 Histórico de alterações

---

Data	Alteração
31/08/2020	Este é o primeiro lançamento oficial.