Object Storage Service

User Guide

Issue	01
Date	2024-10-24

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Accessing OBS

OBS Domain Name

The following two concepts are related to OBS domain names:

1. **Endpoint:** OBS provides an endpoint for each region. An endpoint is a domain name to access OBS in a certain region and is used to receive access requests sent from that region. For the mapping between regions and OBS endpoints, see **Regions and Endpoints**.
2. **Bucket domain name**: Each bucket in OBS has a domain name. A domain name is the Internet address of a bucket and can be used to access the bucket over the Internet. It is typically used in cloud application development and data sharing scenarios.

   An OBS bucket domain name is in the *BucketName.Endpoint* format.

   BucketName indicates the name of a bucket, and *Endpoint* indicates the OBS domain name of the region where the bucket is located.

Table 1-1 lists the bucket domain name and other domain names of OBS, including their formats and protocols.

Table 1-1 OBS domain names

Type	Structure	Description	Protocol Type
Region domain name	[Structure] *Endpoint* [Example] obs.ap-southeast-1.myhuaweicloud.com	Each region has an OBS endpoint, which is the OBS service domain name of the region. For a complete mapping between regions and OBS endpoints, see **Regions and Endpoints**.	HTTPS HTTP

Type	Structure	Description	Protocol Type
Bucket domain name	[Structure] *BucketName.Endpoint* [Example] bucketname.obs.ap-southeast-1.myhuaweicloud.com	After a bucket is created, you can use the domain name to access the bucket. You can assemble the bucket domain name by putting the bucket name and endpoint together, or you can obtain it by **viewing the basic bucket information** on OBS Console or OBS Browser+.	HTTPS HTTP
Object domain name	[Structure] *BucketName.Endpoint*/*ObjectName* [Example] bucketname.obs.ap-southeast-1.myhuaweicloud.com/object.txt	After an object is uploaded to a bucket, you can use the object domain name to access the object. You can assemble the object domain name by putting the bucket name, OBS endpoint, and object name together, or you can obtain it by **viewing the object attributes** on OBS Console or OBS Browser+. Alternatively, you can call the GetObjectUrl API in an SDK to obtain the object domain name.	HTTPS HTTP
Static website domain name	[Structure] *BucketName*.**obs-website.***Endpoint* [Example] bucketname.obs-website.ap-southeast-1.myhuaweicloud.com	A static website domain name is a bucket domain name when the bucket is configured to host a static website.	HTTPS HTTP
User-defined domain name	Domain names that have been licensed by the Ministry of Industry and Information Technology (MIIT) of China.	You can configure a user-defined domain name for a bucket so that you can access the bucket with the configured domain name.	HTTP

Endpoints

OBS has an endpoint in each region.

Generally, the endpoint carried in a request for accessing OBS must be the endpoint of the region where the requested resource resides. However, in some special cases, you can use any endpoint.

1. **Scenarios where the endpoint in a request must be the endpoint of the region where the requested resources reside**

   When accessing a bucket or an object, the endpoint in the request must be the endpoint of the region where the bucket or object resides.

   For example, if bucket **mybucket** is in region **ap-southeast-1**, you can list objects in the bucket by sending a request shown in the following example:

   A correct example of request and response for listing objects:

   [Request]

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

   [Response]

```
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: ap-southeast-1
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>b4bf1b36d9ca43d984fbcb9491b6fce9</ID>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
</ListBucketResult>
```

   If the endpoint in the request is not consistent with the endpoint of the requested region, an error message is returned indicating that the bucket does not exist.

   In the example above, bucket **mybucket** is in region **ap-southeast-1**. If you use the endpoint of **cn-south-1** (**mybucket.obs.cn-south-1.myhuaweicloud.com**) to access the bucket, HTTP 404 is returned, indicating that the bucket does not exist. In this case, you can call the API for **obtaining bucket location** to obtain the bucket's region ID, and then send the request again.

   An incorrect example of request and response for listing objects:

   [Request]

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

   [Response]

```
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. **Scenarios where any endpoint can be used in a request**

   You can use the endpoint of any region in the API requests for listing buckets and for obtaining a bucket's region information, because these two APIs search for buckets from all regions.

   An example of request and response for obtaining a bucket's region information:

   For example, if bucket **mybucket** is in region **ap-southeast-1** and the endpoint of the **cn-south-1** region is used in the request, the bucket location information can still be obtained.

   [Request]

```
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
```

   [Response]

```
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/">ap-southeast-1</Location>
```

Accessing OBS over the Internet

Accessing OBS over a public network may generate inbound traffic for write operations (for example, uploading data to OBS), as well as outbound traffic for read operations (for example, downloading data from OBS). Inbound traffic does not incur fees, but outbound traffic does.

For details about the pricing of outbound traffic over the Internet, see **Product Pricing Details**.

If you access OBS over a public network, you can use a URL to specify resources in OBS. An OBS URL is structured as follows:

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

Table 1-2 Parameters in an OBS URL

Parameter	Description	Mandatory or Optional
Protocol	The protocol used for sending requests, which can be either HTTP or HTTPS. HTTPS is a protocol that ensures secure access to resources. OBS supports both HTTP and HTTPS.	Mandatory
BucketName	Name of the requested bucket, which uniquely identifies a bucket in OBS.	Optional
Endpoint	Domain name (endpoint) of the region where the OBS bucket is located. For details about the OBS domain name of each region, see **Regions and Endpoints**.	Mandatory
Port	The port enabled for protocols used for sending requests. The value varies depending on software server deployment. If no port number is specified, the protocol uses the default value. Each transmission protocol has its default port number. In OBS, the default HTTP port number is **80** and that of HTTPS is **443**.	Optional
Object	Path of the requested object resource.	Optional
Param	Specific resource contained by a bucket or object. The default value of this parameter indicates that the bucket or object itself is obtained.	Optional

Example: You have a bucket named **mybucket** in the CN-Hong Kong (**ap-southeast-1**) region. The bucket contains an object named **myfolder/myfile.txt**. The URL for accessing the object over the public network is as follows:

https://mybucket.obs.ap-southeast-1.myhuaweicloud.com/myfolder/myfile.txt

NOTE

All API requests except the one for listing objects must contain the **BucketName**. In consideration of the DNS resolution performance and reliability, OBS requires that the bucket name must precede the **Endpoint** when a request carrying a bucket name is constructed to form a three-level domain name, also mentioned as virtual hosting access domain name.

Accessing OBS over the Intranet

Accessing OBS over a private network refers to accessing OBS through the internal communication network between different Huawei Cloud services. Inbound traffic generated by accessing OBS over an intranet (write operations like uploading data to OBS) and outbound traffic (read operations like downloading data from OBS) are free of charge.

For example, you can access OBS from an Elastic Cloud Server (ECS) over a private network. Such access is not susceptible to public network quality issues, and it also reduces costs.

OBS provides you with a best practice for configuring such access. For details, see **Accessing OBS from an ECS over the Intranet**.

Checking OBS Version (OBS 2.0 or OBS 3.0)

OBS architecture has undergone two generations: OBS 2.0 and OBS 3.0. A newly created bucket is stored in OBS 3.0 by default, and the bucket version is OBS 3.0. However, previously created buckets are still in OBS 2.0.

Basic OBS features and functions are supported by both OBS 3.0 and OBS 2.0. Some new features are supported only by OBS 3.0, such as image processing and cross-region replication.

You can check the bucket version on OBS Console or use the **Head Bucket** API to check whether your bucket is in OBS 2.0 or OBS 3.0. The details are as follows:

Method 1: Log in to OBS Console and check the basic bucket information.

If **Bucket Version** is **3.0**, the bucket is stored in OBS 3.0. If not, the bucket is stored in OBS 2.0.

Method 2: Use the Head Bucket API to check the bucket version.

Sample Request:

```
HEAD / HTTP/1.1
Host: bucketname.obs.ap-southeast-1.myhuaweicloud.com
Accept: */*
Date: WED, 01 Jul 2015 02:23:25 GMT
Authorization: auth string
```

Sample Response:

```
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: 32AAAQAAEAABAAAQAAEAABAAAQAAEAABCS8wS9l00ll4oMWmdniV7XmdAvfewrQq
Date: WED, 01 Jul 2015 02:23:25 GMT
Content-Length: 0
```

In the response message, **x-obs-version: 3.0** indicates that the bucket is stored in OBS 3.0. If this header does not exist or the value of this header is displayed otherwise, the bucket is stored in OBS 2.0.

For details about the **Head Bucket** API, see **Obtaining Bucket Metadata**.

2 Storage Class

2.1 Overview

Scenarios

As Internet develops, data storage scenarios become increasingly diverse. Limited storage classes cannot meet diverse storage and cost management requirements. OBS provides the following storage classes: Standard, Infrequent Access, Deep Archive (under a beta test), and Archive. For more information about billing for different storage classes, see **Storage Space**.

Storage Classes

OBS supports the following storage classes:

Standard

- Scenarios: The Standard storage class features low latency and high throughput. It is therefore good for storing frequently (multiple times a month) accessed files or small files (less than 1 MB). Its application scenarios include big data analytics, mobile apps, hot videos, and social apps.
- Redundancy: The Standard storage class provides two redundancy options. Multi-AZ storage means data is stored in multiple AZs, which enables more reliability. Single-AZ storage means data is stored in a single AZ, which enables more cost-effectiveness.
- Specifications: The minimum billable object size is 64 KB. There is no requirement for the minimum storage duration.

  NOTE

  - The minimum billable object size refers to the minimum object size that is eligible for billing. For example, a 32 KB Standard object will be billed as if it were 64 KB.
  - The minimum storage duration refers to the least time that is eligible for billing. This means that objects will be billed for the minimum storage duration even if they are not stored for that long. For example, if an Infrequent Access object is deleted after being stored in OBS for 20 days, it will be billed for the storage of 30 days (the minimum storage duration).

- Data restoration: N/A

NOTE

- To access objects in the Archive or Deep Archive storage class, including reading or downloading them, accessing them with a URL, or configuring an ACL or metadata for them, you must first restore them. For more information, see **Restoring an Object from Archive or Deep Archive Storage**.

Infrequent Access

- Scenarios: The Infrequent Access storage class is for storing data that is infrequently (less than 12 times a year) accessed, but when needed, the access has to be fast. It can be used for file synchronization, file sharing, enterprise backups, and many other scenarios.
- Redundancy: The Infrequent Access storage class provides two redundancy options. Multi-AZ storage means data is stored in multiple AZs, which enables more reliability. Single-AZ storage means data is stored in a single AZ, which enables more cost-effectiveness.
- Specifications: The minimum billable object size is 64 KB. The minimum storage duration is 30 days.
- Data restoration: Infrequent Access objects can be accessed only after being restored. The restoration process is automatically completed by the system and users are unaware of the restoration. When accessing Infrequent Access objects, you are charged for requesting and restoring objects. There is no billing for temporary file storage.

NOTE

- Data restoration charges involve the request charge, restoration charge (standard or expedited), and temporary file storage charge. For details, see **Data Restoration Pricing**. To calculate an estimated price, use **Price Calculator**.

Archive

- Scenarios: The Archive storage class is ideal for storing data that is rarely (once a year) accessed. Its application scenarios include data archiving and long-term backups. This storage class is secure, durable, and inexpensive, so it is often used to replace tape libraries. To keep cost low, it may take up to hours to restore data from the Archive storage class.
- Redundancy: The Archive storage class only supports single-AZ redundancy.
- Specifications: The minimum billable object size is 64 KB. The minimum storage duration is 90 days.

NOTE

- The minimum billable object size refers to the minimum object size that is eligible for billing. For example, a 32 KB Standard object will be billed as if it were 64 KB.
- The minimum storage duration refers to the least time that is eligible for billing. This means that objects will be billed for the minimum storage duration even if they are not stored for that long. For example, if an Archive object is deleted after being stored in OBS for 80 days, it will be billed for the storage of 90 days.

- Data restoration: Archive objects can be accessed only after being restored. The Archive storage class supports standard and expedited restoration. The standard restoration takes 3 to 5 hours, and the expedited restoration takes 1 to 5 minutes. You can enable **Direct Reading** to access Archive objects without restoration. Direct reading is billable. Decide whether or not to use this function based on your requirements.

NOTE

- Data restoration charges involve the request charge, restoration charge (standard or expedited), and temporary file storage charge. For details, see **Data Restoration Pricing**. To calculate an estimated price, use **Price Calculator**.

Deep Archive (Under a Beta Test)

- Scenarios: The Deep Archive storage class is suitable for storing data that is barely (once every few years) accessed. This storage class costs less than the Archive storage class, but takes longer time (usually several hours) to restore data.
- Redundancy: The Deep Archive storage class only supports single-AZ redundancy.
- Specifications: The minimum billable object size is 64 KB. The minimum storage duration is 180 days.

  NOTE

  - The minimum billable object size refers to the minimum object size that is eligible for billing. For example, a 32 KB Standard object will be billed as if it were 64 KB.
  - The minimum storage duration refers to the least time that is eligible for billing. This means that objects will be billed for the minimum storage duration even if they are not stored for that long. For example, if a Deep Archive object is deleted after being stored in OBS for 170 days, it will be billed for the storage of 180 days.
- Data restoration: Deep Archive objects can be accessed only after being restored. The Deep Archive storage class supports standard and expedited restoration. The standard restoration takes 5 to 12 hours, and the expedited restoration takes 3 to 5 hours.

  NOTE

  - Data restoration charges involve the request charge, restoration charge (standard or expedited), and temporary file storage charge. For details, see **Data Restoration Pricing**. To calculate an estimated price, use **Price Calculator**.

Comparison of Storage Classes

Item	Standard	Infrequent Access	Archive	Deep Archive (Under Limited Beta Testing)
Feature	Top-notch performance, high reliability and availability	Reliable, inexpensive storage with real-time access	Long-term, inexpensive storage for Archive data	Long-term storage for Deep Archive data, with a lower unit price than Archive storage

Item	Standard	Infrequent Access	Archive	Deep Archive (Under Limited Beta Testing)
Use cases	Cloud applications, data sharing, content sharing, and hot data storage	Web disk applications, enterprise backups, active archiving, and data monitoring	Archiving, medical image storage, video material storage, and replacement of tape libraries	Archive data that is barely accessed
Designed durability (single-AZ)	99.999999999% (11 nines)	99.999999999% (11 nines)	99.999999999% (11 nines)	99.999999999% (11 nines)
Designed durability (multi-AZ)	99.9999999999% (12 nines)	99.9999999999% (12 nines)	Not supported	Not supported
Designed availability (single-AZ)	99.99%	99%	99%	99%
Designed availability (multi-AZ)	99.995%	99.5%	Not supported	Not supported
Minimum billable object size	64 KB	64 KB	64 KB	64 KB
Minimum storage duration	N/A	30 days	90 days	180 days
Data access speed	Real-time access	Real-time access	● If direct reading is disabled, objects must be restored before you can access them. Standard restoration: 3–5 hours; Expedited restoration: 1–5 minutes ● If direct reading is enabled, objects can be accessed in real time.	Objects must be restored before you can access them. Standard restoration: 5–12 hours; Expedited restoration: 3–5 hours

Item	Standard	Infrequent Access	Archive	Deep Archive (Under Limited Beta Testing)
Data restoration charges	N/A	Billed for each GB restored.	Billed for each GB restored. Pricing differs depending on standard or expedited restoration.	Billed for each GB restored. Pricing differs depending on standard or expedited restoration.
Image processing	Supported	Supported	Not supported	Not supported

2.2 Configuring a Storage Class

Scenarios

This section describes how to configure storage classes for buckets or objects when creating buckets or uploading objects. If you want to change the storage class of a bucket or object, see **Changing the Storage Classes of Buckets and Objects**.

Bucket and Object Storage Classes

When creating a bucket, you can specify a storage class for it. You can also change the storage class after the bucket has been created.

You can specify a storage class for an object when uploading it, or you can change the object storage class after the object has been uploaded.

Changing the storage class of a bucket does not change the storage class of existing objects in the bucket. However, any new objects uploaded to the bucket will inherit the bucket's new storage class.

Configuring a Storage Class During Bucket Creation

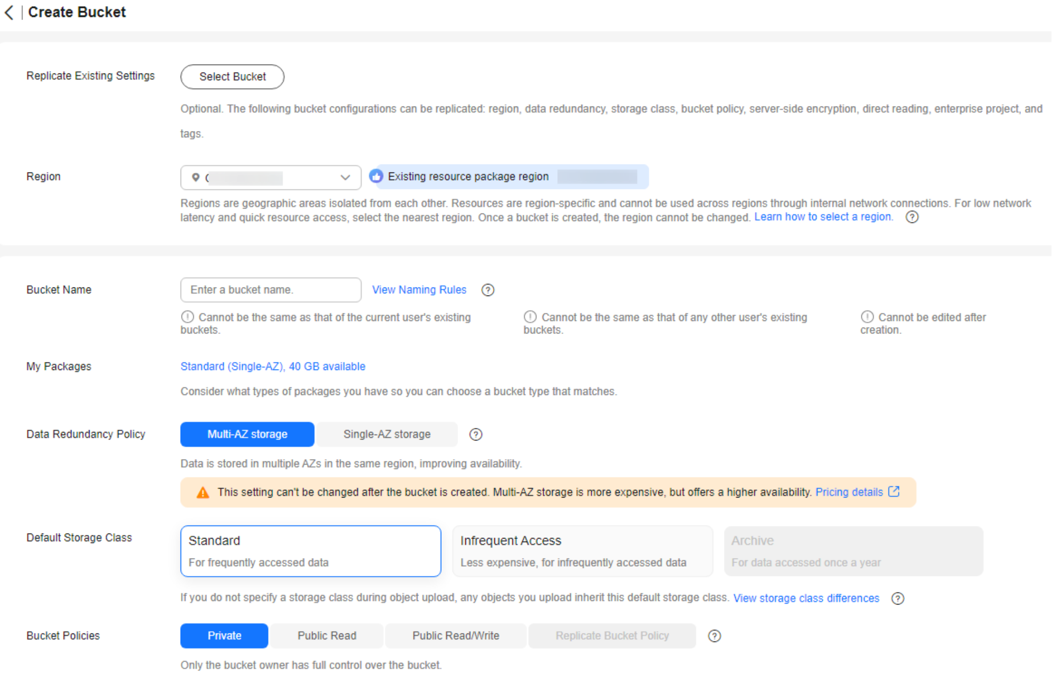
You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to configure storage classes for buckets during creation.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the upper right corner, click **Create Bucket**. The **Create Bucket** page is displayed.

Figure 2-1 Creating a bucket



< | Create Bucket

Replicate Existing Settings: Select Bucket

Optional. The following bucket configurations can be replicated: region, data redundancy, storage class, bucket policy, server-side encryption, direct reading, enterprise project, and tags.

Region: Existing resource package region

Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region. Once a bucket is created, the region cannot be changed. Learn how to select a region.

Bucket Name: Enter a bucket name. View Naming Rules

Cannot be the same as that of the current user's existing buckets.

Cannot be the same as that of any other user's existing buckets.

Cannot be edited after creation.

My Packages: Standard (Single-AZ), 40 GB available

Consider what types of packages you have so you can choose a bucket type that matches.

Data Redundancy Policy: Multi-AZ storage | Single-AZ storage

Data is stored in multiple AZs in the same region, improving availability.

This setting can't be changed after the bucket is created. Multi-AZ storage is more expensive, but offers a higher availability. Pricing details

Default Storage Class:
- Standard: For frequently accessed data
- Infrequent Access: Less expensive, for infrequently accessed data
- Archive: For data accessed once a year

If you do not specify a storage class during object upload, any objects you upload inherit this default storage class. View storage class differences

Bucket Policies: Private | Public Read | Public Read/Write | Replicate Bucket Policy

Only the bucket owner has full control over the bucket.

Step 3 Configure bucket parameters.

Table 2-1 Bucket parameters

Parameter	Description
Replicate Existing Settings	Optional. To use this function, click **Select Bucket** and select a bucket from the list as the replication source. After the replication source is selected, the following settings are replicated to the bucket you are creating: region, data redundancy policy, storage class, bucket policy, server-side encryption, direct reading, enterprise project, and tags. You can still change some or all of the replicated settings as needed.
Region	Geographic area where a bucket resides. For low latency and fast access, select the region nearest to you. Once the bucket is created, its region cannot be changed. Most OBS features are available in all regions, but some are only available for certain regions. Consider the feature availability in each region when you select a region for a bucket. For details, see **Function Overview**. If your ECS needs to access an OBS bucket over an intranet, ensure that the bucket and the ECS are in the same region. For details, see **Accessing OBS over an Intranet**.

Parameter	Description
Bucket Name	Name of the bucket. A bucket name must be unique across all accounts and regions. Once a bucket is created, its name cannot be changed. According to the globally applied DNS naming rules, an OBS bucket name: ● Must be unique across all accounts and regions. The name of a deleted bucket can be reused for another bucket or a parallel file system at least 30 minutes after the deletion. ● Must be 3 to 63 characters long. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. ● Cannot start or end with a period (.) or hyphen (-), and cannot contain two consecutive periods (..) or contain a period (.) and a hyphen (-) adjacent to each other. ● Cannot be formatted as an IP address. **NOTE** When you access OBS through HTTPS using virtual hosted-style URLs, if the bucket name contains a period (.), the certificate verification will fail. To work around this issue, you are advised not to use periods (.) in bucket names.
Data Redundancy Policy	● **Multi-AZ storage**: Data is stored in multiple AZs to achieve higher reliability. ● **Single-AZ storage**: Data is stored in a single AZ, with lower costs. For details about the performance comparison between multi-AZ and single-AZ storage, see **Comparison of Storage Classes**. Once a bucket is created, the data redundancy policy cannot be changed, so choose the policy that can meet your needs. ● Multi-AZ storage is not available for buckets in the Archive storage class. ● Multi-AZ storage is not available for buckets in the Deep Archive storage class.

Parameter	Description
Default Storage Class	The storage class of a bucket. Different storage classes meet different requirements for storage performance and costs. ● The Standard storage class is for storing a large number of hot files or small files that are frequently accessed (multiple times per month on average) and require quick retrieval. ● The Infrequent Access storage class is for storing data that is less frequently accessed (less than 12 times per year on average) but requires quick retrieval. ● The Archive storage class is for archiving data that is rarely accessed (once a year on average) and has less demanding requirements for quick retrieval. ● The Deep Archive storage class is for storing data that is rarely accessed (a lower frequency than the archived data) and has less demanding requirements for quick retrieval. For details, see **Storage Classes**.
Bucket Policy	Controls read and write permissions for buckets. ● **Private**: Only users granted permissions by the bucket ACL can access the bucket. ● **Public Read**: Anyone can read objects in the bucket. ● **Public Read and Write**: Anyone can read, write, or delete objects in the bucket.
Server-Side Encryption	Choose **SSE-KMS**. For the encryption key type, you can choose **Default** or **Custom**. If **Default** is used, the default key of the current region will be used to encrypt your objects. If there is no such a default key, OBS creates one the first time you upload an object. If **Custom** is used, you can choose a custom key you created on the KMS console to encrypt your objects. If **SSE-OBS** is chosen, the keys created and managed by OBS are used for encryption. After you enable server-side encryption for the bucket, any object you upload to it will inherit encryption from the bucket by default. You can also configure new encryption with SSE-KMS or SSE-OBS for the object.
WORM	When you enable write-once-read-many (WORM), you can configure a retention policy for the current bucket. The object version which the retention policy is applied to cannot be deleted within a specified period. You can only enable WORM when you create a bucket. Once enabled for a bucket, WORM cannot be disabled. When you enable WORM, OBS automatically enables versioning for the bucket, and versioning cannot be suspended later for that bucket.

Parameter	Description
Direct Reading	Direct reading allows you to directly download objects from the Archive storage class without restoring them first. Direct reading is a billable function. For details, see **Product Pricing Details**. No matter which default storage class you select, you can enable direct reading for your bucket. For example, if you select the Standard storage class and enable direct reading for your bucket, you can directly download objects stored in the Archive storage class from your bucket.
Enterprise Project	You can add a bucket to an enterprise project for unified management. Create an enterprise project by referring to **Creating an Enterprise Project**. The default enterprise project is named **default**. On the **Enterprise Project Management** page, **create an enterprise project**, and **add a user group to the enterprise project**. By doing so, users in this user group obtain the operation permissions for the buckets and objects in the enterprise project. **NOTE** Only an enterprise account can configure enterprise projects. OBS ReadOnlyAccess and OBS OperateAccess are the fine-grained authorizations of the enterprise project user group in OBS.
Tags	Optional. Tags are used to identify and classify buckets in OBS. Each tag is represented by a key-value pair. For more information, see **Tags**.

Step 4 Click **Create Now**.

----End

Using the API

Specifying a Storage Class During Bucket Creation (adding the **x-obs-storage-class** header)

Using SDKs

Java	**Python**	**C**	**Go**	**BrowserJS**	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**

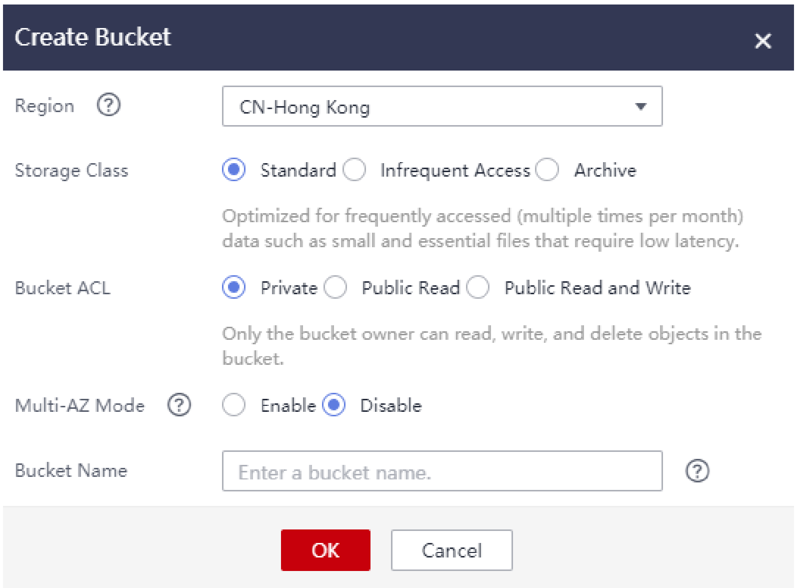
Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 In the upper part of the page, click **Create Bucket**.

Step 3 In the displayed dialog box, configure bucket parameters, as shown in **Figure 2-2**.

Figure 2-2 Creating buckets



Create Bucket

Region: CN-Hong Kong

Storage Class: ● Standard ○ Infrequent Access ○ Archive

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

Bucket ACL: ● Private ○ Public Read ○ Public Read and Write

Only the bucket owner can read, write, and delete objects in the bucket.

Multi-AZ Mode: ○ Enable ● Disable

Bucket Name: Enter a bucket name.

Table 2-2 Creating buckets

Parameter	Description
Region	Enter the region where you want to create a bucket. Once the bucket is created, its region cannot be changed.
Storage Class	The storage class of a bucket. Different storage classes meet customers' needs for storage performance and costs. ● The Standard storage class is for storing a large number of hot files or small files that are frequently accessed (multiple times per month on average) and require quick retrieval. ● The Infrequent Access storage class is for storing data that is less frequently accessed (less than 12 times per year on average) but requires quick retrieval. ● The Archive storage class is for archiving data that is rarely accessed (once a year on average) and has less demanding requirements for quick retrieval. For details, see **Storage Classes**.

Parameter	Description
Bucket ACL	Controls read and write permissions for buckets. ● **Private**: Only users granted permissions by the bucket ACL can access the bucket. ● **Public Read**: Anyone can read objects in the bucket. ● **Public Read and Write**: Anyone can read, write, or delete objects in the bucket.
Multi-AZ Mode	If multi-AZ storage is enabled, data will be stored in multiple AZs. ● Once a bucket is created, its multi-AZ status cannot be changed. So, plan in advance and determine whether to enable the multi-AZ function during bucket creation.
Bucket Name	Name of the bucket you want to create, which must be globally unique. A bucket name: ● Must be 3 to 63 characters long and start with a digit or letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. ● Cannot be formatted as an IP address. ● Cannot start or end with a hyphen (-) or period (.). ● Cannot contain two consecutive periods (..), for example, **my..bucket**. ● Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, **my-.bucket** or **my.-bucket**.

You can click ? next to the bucket name to learn about the bucket naming rules. A user can create a maximum of 100 buckets in OBS.

NOTE

- When a URL is used to access a bucket, the bucket name will become a part of the URL. According to the DNS rule, URLs do not support uppercase letters and cannot be used to access a bucket whose name contains uppercase letters. Therefore, a bucket name can contain only lowercase letters, digits, hyphens (-), and periods (.) For example, if you attempt to access bucket **MyBucket** using a URL, **MyBucket** will be parsed as **mybucket**. This results in an access error.
- DNS naming rules standardize bucket names globally, facilitating the resolution during bucket access. With the DNS naming rules used, you can benefit from new functions and optimized features, and configure static website hosting for buckets.
- Once you have created a bucket, you cannot change the name of it. Make sure that the bucket name you set is appropriate.

Step 4 Click **OK**. If the bucket is successfully created, it is displayed in the bucket list. If the creation fails, an error message will be displayed.

----End

Using the CLI Tool - obsutil

Command Line Structure

- Windows:

```
obsutil mb obs://bucket [-fs] [-az=xxx] [-acl=xxx] [-sc=xxx] [-location=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

- Linux or macOS:

```
./obsutil mb obs://bucket [-fs] [-az=xxx] [-acl=xxx] [-sc=xxx] [-location=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

Examples

- Take the Windows OS as an example. Run the **obsutil mb obs://bucket-test** command to create a bucket. The example bucket was created.

```
obsutil mb obs://bucket-test

Create bucket [bucket-test] successfully, request id [0000016979E1D2EA860BB5E80A6B8FCC]
```

- Take the Windows OS as an example. Run the **obsutil mb obs://bucket001** command to create a namesake bucket. The example bucket failed to be created.

```
obsutil mb obs://bucket001

Create bucket [bucket001] failed, http status [409], error code [BucketAlreadyExists], error message
[The requested bucket name is not available. The bucket namespace is shared by all users of the
system. Please select a different name and try again.], request id
[04030000016757F31A0333281A6B1E92]
```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name **NOTE** A bucket name: ● Must be 3 to 63 characters long and start with a digit or letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. ● Cannot be formatted as an IP address. ● Cannot start or end with a hyphen (-) or period (.). ● Cannot contain two consecutive periods (..), for example, **my..bucket**. ● Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, **my-.bucket** or **my.-bucket**.
fs	Optional (additional parameter)	Creates a parallel file system.

Parameter	Optional or Mandatory	Description
az	Optional (additional parameter)	Specifies a bucket's data redundancy policy. Possible values are: ● multi-az **NOTE** If **multi-az** is used, a bucket with the multi-AZ storage policy will be created. If this parameter is not specified, a bucket with the single-AZ storage policy will be created.
acl	Optional (additional parameter)	Access control policies that can be specified when creating a bucket. Possible values are: ● private ● public-read ● public-read-write **NOTE** The three values above indicate, respectively, private read and write, public read, and public read and write.
sc	Optional (additional parameter)	Default bucket storage class that can be specified when creating a bucket. Possible values are: ● **standard**: Standard storage class. It features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month) or data that is smaller than 1 MB. ● **warm**: Infrequent Access storage class. It is ideal for storing infrequently accessed (less than 12 times a year) data, but when needed, the access has to be fast. ● **cold**: Archive storage class. It provides secure, durable, and inexpensive storage for rarely-accessed (once a year) data. ● **deep-archive**: Deep Archive storage class (under a beta test). It is suitable for storing data that is barely (once every few years) accessed. This storage class costs less than the Archive storage class, but takes longer time (usually several hours) to restore data.
location	Mandatory unless the region where a bucket will be created is the default region (additional parameter)	Region where the bucket will be located **NOTE** This parameter indicates the region where a bucket will be created. It is mandatory if the endpoint belongs to any region other than CN North-Beijing1 (cn-north-1). To view the currently valid regions, see **Regions and Endpoints**.

Parameter	Optional or Mandatory	Description
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **Parameter Description**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

Configuring a Storage Class During Object Uploads

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to configure storage classes for objects during uploads.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

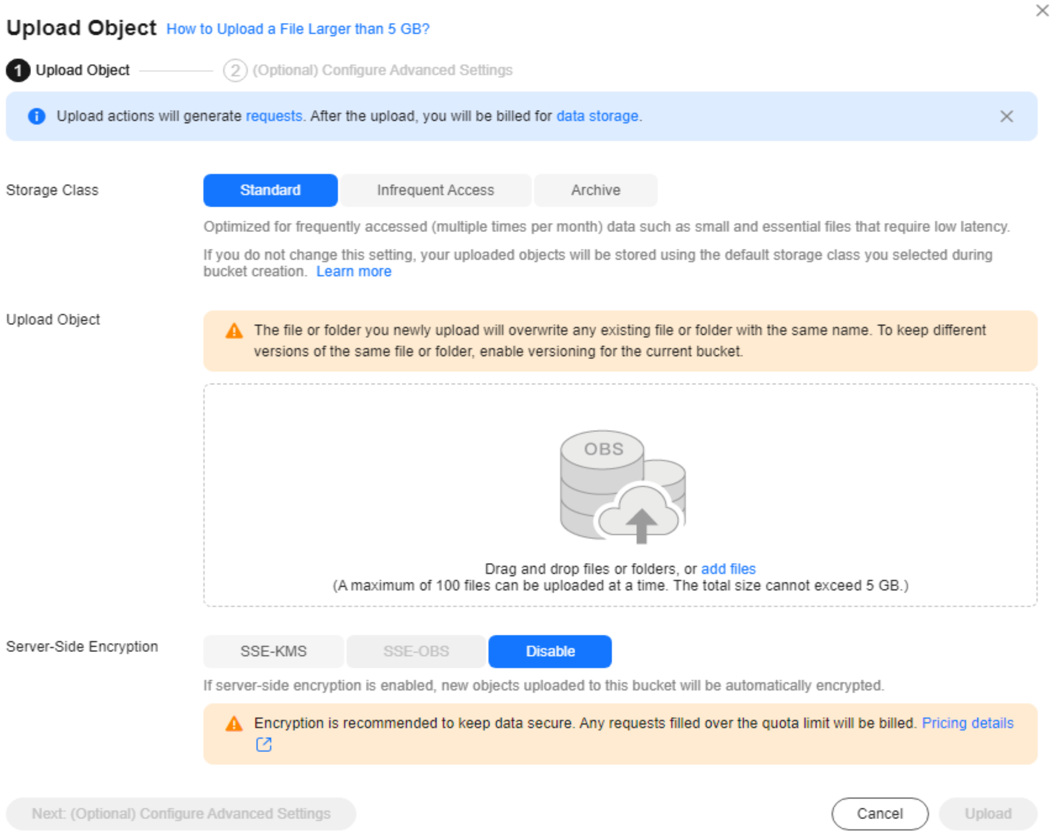
Step 3 Go to the folder where you want to upload files and click **Upload Object**. The **Upload Object** dialog box is displayed.

Batch upload is used as an example here. If the region you are using does not support batch upload, perform operations as instructed.

NOTE

If the files that you want to upload to OBS are stored in Microsoft OneDrive, it is recommended that the names of these files contain a maximum of 32 characters to ensure compatibility.

Figure 2-3 Uploading objects



Upload Object How to Upload a File Larger than 5 GB?

1 Upload Object — 2 (Optional) Configure Advanced Settings

Upload actions will generate requests. After the upload, you will be billed for data storage.

Storage Class: Standard | Infrequent Access | Archive

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

If you do not change this setting, your uploaded objects will be stored using the default storage class you selected during bucket creation. Learn more

Upload Object

The file or folder you newly upload will overwrite any existing file or folder with the same name. To keep different versions of the same file or folder, enable versioning for the current bucket.

Drag and drop files or folders, or add files
(A maximum of 100 files can be uploaded at a time. The total size cannot exceed 5 GB.)

Server-Side Encryption: SSE-KMS | SSE-OBS | Disable

If server-side encryption is enabled, new objects uploaded to this bucket will be automatically encrypted.

Encryption is recommended to keep data secure. Any requests filled over the quota limit will be billed. Pricing details

Step 4 Select a storage class. If you do not specify a storage class, the objects you upload inherit the storage class of the bucket.

NOTE

An object can have a different storage class from its bucket. You can specify a storage class for an object when uploading it, or you can change the object storage class after the object has been uploaded.

Step 5 In the **Upload Object** area, drag and drop the files or folders you want to upload.

You can also click **add files** in the **Upload Object** area to select files.

Step 6 **Server-Side Encryption**: Choose **Disable**, **SSE-KMS**, or **SSE-OBS**. For details, see **Uploading a File with Server-Side Encryption**.

NOTE

If the bucket has server-side encryption configured, the object you upload will inherit encryption from the bucket by default.

Step 7 (Optional) To configure metadata or WORM retention policies, click **Next: (Optional) Configure Advanced Settings**.

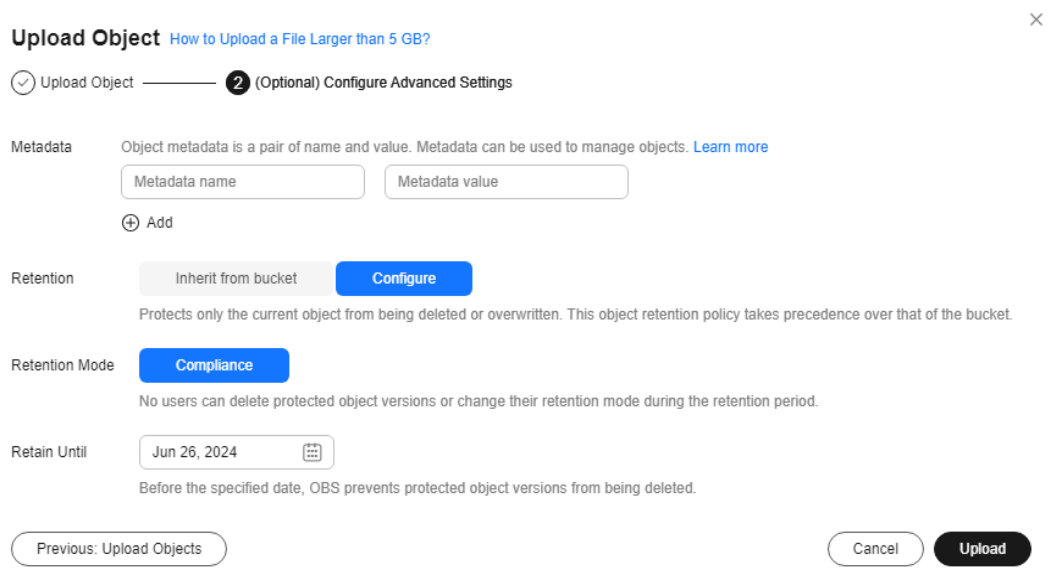
NOTE

WORM retention policies can be configured in the advanced settings only when WORM is enabled for the bucket.

Configuring metadata: Add metadata ContentDisposition, ContentLanguage, WebsiteRedirectLocation, ContentEncoding, or ContentType as needed. For more information, see **OBS Object Metadata**. Metadata is a set of name-value pairs. The metadata value cannot be left blank. You can add two or more metadata entries by clicking **Add**.

Configuring WORM retention: Choose **Inherit from bucket**, or choose **Configure** and then specify a retention period, to automatically protect new objects uploaded to the bucket from being deleted.

Figure 2-4 Configuring metadata or WORM retention



Upload Object How to Upload a File Larger than 5 GB?

Metadata: Object metadata is a pair of name and value. Metadata can be used to manage objects. Learn more

Retention: Configure

Protects only the current object from being deleted or overwritten. This object retention policy takes precedence over that of the bucket.

Retention Mode: Compliance

No users can delete protected object versions or change their retention mode during the retention period.

Retain Until: Jun 26, 2024

Before the specified date, OBS prevents protected object versions from being deleted.

Step 8 Click **Upload**.

----End

Using the API

Specifying a Storage Class During Object Uploads with PUT (adding the **x-obs-storage-class** header)

Specifying a Storage Class During Object Uploads with POST (adding the **x-obs-storage-class** header)

Initiating a Multipart Upload (adding the **x-obs-storage-class** header during the initiation process if a multipart upload is used)

Using SDKs

Uploading objects	Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

Multipart upload	Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

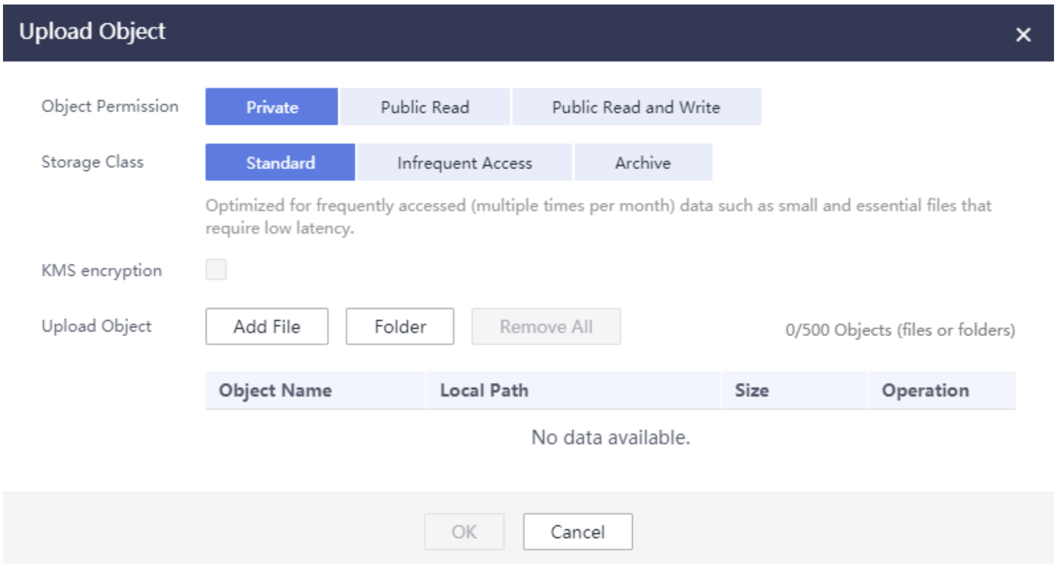
Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Click the bucket where you want to upload files or folders.

Step 3 Click **Upload** and then **Add File** or **Folder**, as shown in **Figure 2-5**.

Figure 2-5 Uploading a file or folder



Upload Object

Object Permission: Private

Storage Class: Standard

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

KMS encryption □

Upload Object

0/500 Objects (files or folders)

Object Name	Local Path	Size	Operation

No data available.

For better experience when using the **Add File** function, you are advised to upload a maximum of 100 files at a time. If you need to upload more, place all the files in a folder and upload them by adding a folder.

NOTE

1. If message "Service Unavailable" is displayed when files are being uploaded, try again later.
2. If an access deny message is displayed when you are uploading a file or folder, possible causes are as follows:

- Access to the bucket is restricted by an ACL. For example, you do not have the write permission for the bucket.
- Access to the bucket is restricted by a bucket policy. For example, you do not have the write permission for the bucket, or write operations cannot be performed on the bucket during the current period.

  If such message is displayed, check ACL and policy settings of the bucket and resolve the problem accordingly.

⚠ CAUTION

You must have access to the file you want to upload, or the file upload will fail.

Step 4 In the displayed dialog box, select the file or folder you want to upload and click **Open**.

You can upload one folder or multiple files at a time. To upload multiple files, hold down **Ctrl** or **Shift** to select multiple files and batch upload them. You can also press **Ctrl+A** to select all files. The operations are consistent with those in Windows operating systems.

Step 5 Click **OK**.

----End

Using the CLI Tool - obsutil

Command Line Structure

- Windows:
  - Uploading a file
    ```
    obsutil cp file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-u] [-vlength] [-vmd5] [-
    p=1] [-threshold=5248800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-
    cpd=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading a folder
    ```
    obsutil cp folder_url obs://bucket[/key] -r [-arcDir=xxx] [-dryRun] [-link] [-f] [-flat] [-u] [-
    vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-
    meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2]
    [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading multiple files/folders
    ```
    obsutil cp file1_url,folder1_url|filelist_url obs://bucket[/prefix] -msm=1 [-r] [-arcDir=xxx] [-
    dryRun] [-link] [-f] [-u] [-vlength] [-vmd5] [-flat] [-j=1] [-p=1] [-threshold=52428800] [-
    acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx][-
    timeRange=time1-time2] [-at] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-
    k=xxx] [-t=xxx]
    ```

    NOTE

    In this command, **/prefix** is the name prefix for uploading folders.
- Linux or macOS:
  - Uploading a file
    ```
    ./obsutil cp file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-u] [-vlength] [-vmd5] [-
    p=1] [-threshold=5248800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-
    cpd=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading a folder
    ```
    ./obsutil cp folder_url obs://bucket[/key] -r [-arcDir=xxx] [-dryRun] [-link] [-f] [-flat] [-u] [-
    vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-
    meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2]
    [-at] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading multiple files/folders
    ```
    ./obsutil cp file1_url,folder1_url|filelist_url obs://bucket[/prefix] -msm=1 [-r] [-arcDir=xxx] [-
    dryRun] [-link] [-f] [-u] [-vlength] [-vmd5] [-flat] [-j=1] [-p=1] [-threshold=52428800] [-
    acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx][-
    timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-
    t=xxx]
    ```

NOTE

In this command, **/prefix** is the name prefix for uploading folders.

Examples

- Take the Windows OS as an example. Run the **obsutil cp d:\temp\test.txt obs://bucket-test/key** command to upload the **test.txt** file in the **temp** directory in the **D:** drive to bucket **bucket-test** and rename the file as **key**.

```
obsutil cp d:\temp\test.txt obs://bucket-test/key

Parallel:      5                   Jobs:          5
Threshold:     50.00MB             PartSize:      auto
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: C:\Users\Administrator\.obsutil_checkpoint

[==================================================] 100.00% 1.68 MB/s 8.46MB/
8.46MB 5s
Upload successfully, 8.46MB, n/a, d:\temp\test.txt --> obs://bucket-test/key, cost [5], status [200],
request id [0000016979E1D2B2860BB5181229C72C]
```

- Take the Windows OS as an example. Run the **obsutil cp d:\temp obs://bucket-test -f -r** command to recursively upload all files and subfolders in the **temp** directory in the **D:** drive to the **temp** folder in bucket **bucket-test**.

```
obsutil cp d:\temp obs://bucket-test -f -r

Parallel:      5                   Jobs:          5
Threshold:     50.00MB             PartSize:      auto
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: C:\Users\Administrator\.obsutil_checkpoint

Task id: 104786c8-27c2-48fc-bc6a-5886596fb0ed
OutputDir: C:\Users\Administrator\.obsutil_output

[=====================================================] 100.00% tps:35.71 2.02 KB/s
7.20MB/7.20MB 0s
Succeed count is:   5         Failed count is:    0
Metrics [max cost:90 ms, min cost:45 ms, average cost:63.80 ms, average tps:35.71]

Task id: 104786c8-27c2-48fc-bc6a-5886596fb0ed
```

Parameter Description

Parameter	Optional or Mandatory	Description
file_url	Optional for uploading multiple files/folders Mandatory for uploading a file	Local file path **NOTE** ● Do not nest paths when uploading multiple files/folders. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time. ● If this parameter is configured when uploading multiple files/folders, **msm** must be set to **1**. In this case, use commas (,) to separate multiple file paths, for example, **file_url1,file_url2**. ● Files and folders can both be included when uploading multiple files/folders. For example, **file_url1,folder_url1,file_url2,folder_url2**.

Parameter	Optional or Mandatory	Description
folder_url	Optional for uploading multiple files/ folders Mandatory for uploading a folder	Local folder path **NOTE** - If **flat** is not configured when uploading a folder, the entire folder is uploaded. If **flat** is configured, all files in the folder are uploaded. - Do not nest paths when uploading multiple files/ folders. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time. - If this parameter is configured when uploading multiple files/folders, **msm** must be set to **1**. In this case, use commas (,) to separate multiple folder paths, for example, **folder_url1,folder_url2**. - Files and folders can both be included when uploading multiple files/folders. For example, **file_url1,folder_url1,file_url2,folder_url2**.
filelist_url	Optional for uploading multiple files/ folders	Indicates the path of the file that contains the list of files/folders to be uploaded. If this parameter is configured, **msm** must be set to **2**. **NOTE** - The list file is in common text file formats, such as TXT and CSV. Each line in the file indicates a file or folder to be uploaded. For example: file_url1 file_url2 folder_url1 folder_url2 - Do not nest paths in the list file. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time.
bucket	Mandatory	Bucket name

Parameter	Optional or Mandatory	Description
key	Optional	Indicates the object name or object name prefix specified when uploading a file, or the object name prefix specified when uploading a folder. The rules are as follows: ● If this parameter is left blank when uploading a file, the file is uploaded to the root directory of the bucket and the object name is the file name. If the value ends with a slash (/), the value is used as the object name prefix when the file is uploaded, and the object name is the value plus the file name. If the value does not end with a slash (/), the file is uploaded with the value as the object name. ● If this parameter is left blank when uploading a folder, the folder is uploaded to the root directory of the bucket. If the value ends with a slash (/), the value is used as the object name prefix of the folder to be uploaded. If the value does not end with a slash (/), the folder to be uploaded is prefixed with the value plus a slash (/).
fr	Optional for uploading a file (additional parameter)	Generates an operation result list when uploading a file.
flat	Optional for uploading a folder or multiple files/ folders (additional parameter)	Uploads all files in a folder but excludes the folder itself.
arcDir	Optional (additional parameter)	Path to which the uploaded files are archived
dryRun	Optional (additional parameter)	Conducts a dry run.

Parameter	Optional or Mandatory	Description
link	Optional (additional parameter)	The actual path of the symbolic link of a file/folder **NOTICE** ● If this parameter is not specified and the file to be uploaded is a symbolic-link file whose target file does not exist, the exception message "The system cannot find the file specified" will be displayed in Windows, while the exception message "No such file or directory" will be displayed in macOS or Linux. ● Avoid the symbolic link loop of a folder, or the upload will exit due to panic. If you do not want the system to panic, set **panicForSymbolicLinkCircle** to **false** in the configuration file.
u	Optional (additional parameter)	Indicates incremental upload. If this parameter is configured, each file can be uploaded only when it does not exist in the bucket, its size is different from the corresponding object in the bucket, or its latest modification time is later than that of the corresponding object in the bucket. **CAUTION** When you compare each local file with data in the bucket, a billable HEAD request is generated. For details, see **Requests**.
vlength	Optional (additional parameter)	After the upload completes, check whether the sizes of the objects in the bucket are the same as those of the local files.
vmd5	Optional (additional parameter)	After the upload completes, check whether the MD5 values of the objects in the bucket are the same as those of the local files. **NOTE** ● If the size of the file or folder to be uploaded is too large, using this parameter will degrade the overall performance due to MD5 calculation. ● After the MD5 verification is successful, this parameter value is used as the metadata **x-obs-meta-md5chksum** for later MD5 verification during download or copy.
p	Optional (additional parameter)	Indicates the maximum number of concurrent part uploads for a multipart upload. The default value is the value of **defaultParallels** in the configuration file.

Parameter	Optional or Mandatory	Description
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart upload, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of a file to be uploaded is smaller than the threshold, the entire file is uploaded directly. Otherwise, a multipart upload is used. ● If the entire file is uploaded directly, resumable upload is not supported. ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes).
acl	Optional (additional parameter)	Access control policies that can be specified when uploading files. Possible values are: ● private ● public-read ● public-read-write ● bucket-owner-full-control **NOTE** These four values indicate, respectively, private read and write, public read, public read and write, and bucket owner full control.
sc	Optional (additional parameter)	Indicates the storage classes of objects that can be specified when uploading files. Possible values are: ● **standard**: Standard storage class. It features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month) or data that is smaller than 1 MB. ● **warm**: Infrequent Access storage class. It is ideal for storing infrequently accessed (less than 12 times a year) data, but when needed, the access has to be fast. ● **cold**: Archive storage class. It provides secure, durable, and inexpensive storage for rarely-accessed (once a year) data. ● **deep-archive**: Deep Archive storage class (under a beta test). It is suitable for storing data that is barely (once every few years) accessed. This storage class costs less than the Archive storage class, but takes longer time (usually several hours) to restore data.

Parameter	Optional or Mandatory	Description
meta	Optional (additional parameter)	Indicates the standard and custom metadata that can be specified during file upload. This parameter should be configured in the following format: *key1*:*value1*#*key2*:*value2*#*key3*:*value3*. **NOTE** 1. The format example above indicates that the destination object contains three groups of custom metadata: *key1:value1*, *key2:value2*, and *key3:value3*. 2. Standard metadata headers include **Content-Type**, **Content-Encoding**, **Cache-Control**, **Content-Disposition**, **Content-Language** and **Expires**.
ps	Optional (additional parameter)	Indicates the size of each part in a multipart upload, in bytes. The value ranges from **100KB** to **5GB**. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes). ● The parameter can be set to **auto**. In this case, obsutil automatically configures the part size for each multipart based on the source file size.
cpd	Optional (additional parameter)	Indicates the folder where the checkpoint file is stored. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A checkpoint file is generated for each multipart upload and saved to the **upload** subfolder. After the upload succeeds, the checkpoint file is deleted automatically. If the upload fails or is suspended, when you resume the upload the next time, the system would start from the point recorded in the checkpoint file.
r	Mandatory for uploading a folder (additional parameter) Optional for uploading multiple files	Indicates that files and subfolders within the folder are uploaded recursively.
f	Optional for uploading a folder or multiple files/folders (additional parameter)	Runs in force mode.

Parameter	Optional or Mandatory	Description
j	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the maximum number of concurrent tasks for uploading a folder. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The tool ensures that the value is at least 1.
msm	Mandatory for uploading multiple files/ folders (additional parameter)	Enables the mode for uploading multiple files/ folders. Possible values are **1** and **2**. **NOTE** ● If **msm** is set to **1**, the source URL indicates a list of file/folder names separated by commas. ● If **msm** is set to **2**, the source URL indicates a file containing a list of file/folder names. ● If the file or folder name already contains commas (,), do not set **msm** to **1**. ● If parameter **r** is not specified, the folders in the list will not be uploaded.
exclude	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the matching pattern of files that are excluded, for example, ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use ***** to represent * and **\?** to represent **?**. ● If the name of the file to be uploaded matches the value of this parameter, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and double quotation marks for Windows. ● The matching pattern applies to the absolute file path (including the file name and file directory). ● The matching pattern takes effect only for files in the folder. ● Multiple **exclude** parameters can be specified, for example, **-exclude=*.xxx -exclude=*.xxx**.

Parameter	Optional or Mandatory	Description
include	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the matching pattern of files that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use **\*** to represent ***** and **\\?** to represent **?**. ● The **exclude** rules are preferentially matched. If the name of a file to be uploaded does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is uploaded. If no, the file is skipped. **NOTE** Example of uploading files in a request with the **include** parameter contained: `./obsutil cp /localpath/ obs://test/ -include=/localpath/2022-12-09/* -f -r` This command uploads files that are under **localpath** and start with **/localpath/2022-12-09/** to bucket **test**. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and double quotation marks for Windows. ● The matching pattern applies to the absolute file path (including the file name and file directory). ● The matching pattern takes effect only for files in the folder. ● Multiple **include** parameters can be specified, for example, **-include=*.xxx -include=*.xxx**.
at	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates that only the files whose latest access time is within the value of **timeRange** are uploaded. **NOTE** ● This parameter must be used together with **timeRange**.
disableDirObject	Optional for uploading multiple folders (additional parameter)	Indicates the folders themselves are not uploaded as an object. Configuring this parameter can avoid uploading empty folders to a bucket. If a folder contains files, the files will be uploaded and the original path format is retained.

Parameter	Optional or Mandatory	Description
timeRange	Optional for uploading a folder or multiple files/ folders (additional parameter)	Only files whose latest modification time is within the configured time range are uploaded. This matching pattern has a lower priority than the file matching pattern **exclude** and **include**. **NOTE** - The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. - Automatic formatting is supported. For example, *yyyyMMdd* is equivalent to *yyyyMMdd***000000**, and *yyyyMM* is equivalent to *yyyyMM***01000000**. - If this parameter is set to ***-***time2*, all files whose latest modification time is earlier than *time2* are matched. If it is set to *time1***-***, all files whose latest modification time is later than *time1* are matched. **NOTICE** Time in the matching pattern is the UTC time.
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on folders.
o	Optional (additional parameter)	Indicates the folder where operation result lists are generated. After the command is executed, result lists (possibly success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** - The naming rule for result lists is as follows: **cp_{succeed | failed | warning}_report_***time*_*TaskId***.txt**. - By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring, respectively, **recordMaxLogSize** and **recordBackups** in the configuration file. - If there are multiple folders and files and you need to confirm the detailed error information about a failed task, refer to the failure list **cp_failed_report_***time*_*TaskId***.txt** in the result list folder and the **log files** in the log path.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **Parameter Description**.

Parameter	Optional or Mandatory	Description
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

2.3 Changing the Storage Classes of Buckets and Objects

Scenarios

This section describes how to change the storage classes of buckets and objects.

Constraints

- The storage class of a bucket can only be manually changed. The storage class of an object can be changed manually or automatically based on lifecycle rules.
- The data redundancy policy remains unchanged when the storage class is changed. If a bucket or object is configured with multi-AZ redundancy, it can only be moved to a storage class that supports multi-AZ redundancy, such as the Standard or Infrequent Access storage class. However, it cannot be moved to the Archive storage class, which does not support multi-AZ redundancy.

Manually Changing the Storage Class of a Bucket

OBS allows you to move a bucket to any storage class manually, but does not support using lifecycle rules to do that.

NOTE

- Changing the storage class of a bucket does not change the storage classes of existing objects in the bucket. The storage class of an object uploaded later will inherit the new storage class of the bucket by default. You can **configure lifecycle rules** to change storage classes of objects in batches.

  For example, suppose a bucket, **bucket1**, is in the Standard storage class and contains an object, **object1**, also in the Standard storage class. If **bucket1**'s storage class is transitioned to Infrequent Access, then **object1** will remain in the Standard storage class, but a new object, **object2**, uploaded after the change of the bucket storage class, will be in the Infrequent Access storage class.
- After a bucket is changed from Archive or Deep Archive to Standard or Infrequent Access, existing Archive or Deep Archive objects in the bucket will not be automatically restored.

You can change the storage class of a bucket using OBS Console, APIs, OBS SDKs, or obsutil.

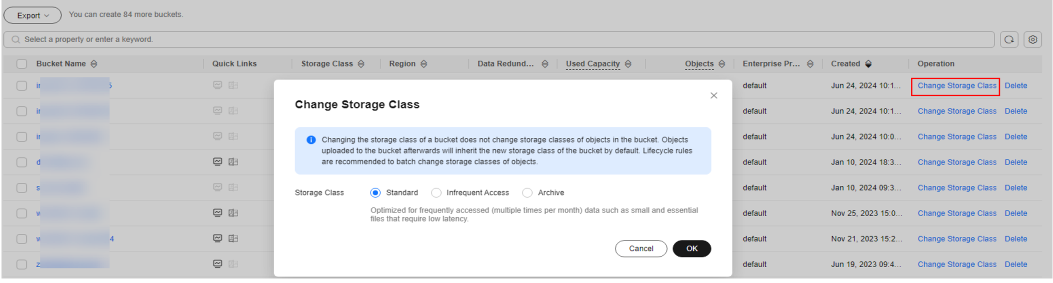
Using OBS Console

Step 1 Log in to **OBS Console**. In the navigation pane, choose **Object Storage**.

Step 2 In the bucket list, locate the bucket whose storage class you want to change and click **Change Storage Class** in the **Operation** column on the right.

Step 3 Choose a new storage class and click **OK**.

Figure 2-6 Manually changing the storage class of a bucket



Export You can create 84 more buckets.

Select a property or enter a keyword.

Bucket Name	Enterprise Pr...	Created	Operation
[redacted]	default	Jun 24, 2024 10:1...	Change Storage Class Delete
[redacted]	default	Jun 24, 2024 10:1...	Change Storage Class Delete
[redacted]	default	Jun 24, 2024 10:0...	Change Storage Class Delete
[redacted]	default	Jan 10, 2024 18:3...	Change Storage Class Delete
[redacted]	default	Jan 10, 2024 09:3...	Change Storage Class Delete
[redacted]	default	Nov 25, 2023 15:0...	Change Storage Class Delete
[redacted]	default	Nov 21, 2023 15:2...	Change Storage Class Delete
[redacted]	default	Jun 19, 2023 09:4...	Change Storage Class Delete

Change Storage Class

Changing the storage class of a bucket does not change storage classes of objects in the bucket. Objects uploaded to the bucket afterwards will inherit the new storage class of the bucket by default. Lifecycle rules are recommended to batch change storage classes of objects.

Storage Class ● Standard ○ Infrequent Access ○ Archive

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

Cancel OK

----End

Using the API

Configuring Storage Class for a Bucket

Using SDKs

Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
```
obsutil stat obs://bucket [-acl] [-bf=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```
- In Linux or macOS
```
./obsutil stat obs://bucket [-acl] [-bf=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

Examples

- Take the Windows OS as an example. Run the **obsutil chattri obs://bucket-test -acl=private** command to change the access control policy of the bucket to private read and write.
```
obsutil chattri obs://bucket-test -acl=private

Set the acl of bucket [bucket-test] to [private] successfully, request id
[04050000016836C5DA6FB21F14A2A0C0]
```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
acl	No	Queries the access control policies of the bucket.
bf	Optional (additional parameter)	How the used bucket capacity is displayed, in raw bytes or human-readable format. Possible values are: ● human-readable ● raw **NOTE** If this parameter is not configured, how the used bucket capacity is displayed is determined by the **humanReadableFormat** parameter in the configuration file.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **Parameter Description**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

Manually Changing the Storage Class of an Object

OBS allows you to move an object to any storage class manually. For objects that are in the Archive or Deep Archive storage class, you must restore them first before changing their storage classes. OBS also supports automatic change of object storage classes. For details, see **Changing the Storage Class of an Object Using Lifecycle Rules**.

NOTE

- When changing the storage class of objects that are in the Infrequent Access, Deep Archive, or Archive storage, there may be additional charges for the **minimum storage duration**.
- Even though objects that are in the Infrequent Access storage class do not need to be restored manually, you still need to pay for **restoration requests and traffic**.
- When changing the storage class of objects that are in the Archive or Deep Archive storage, you are charged for **restoration requests, traffic, and temporary copy storage**.

You can change the storage class of objects using OBS Console, APIs, OBS SDKs, OBS Browser+, or obsutil.

Using OBS Console

Step 1 Log in to **OBS Console**. In the navigation pane, choose **Object Storage**.

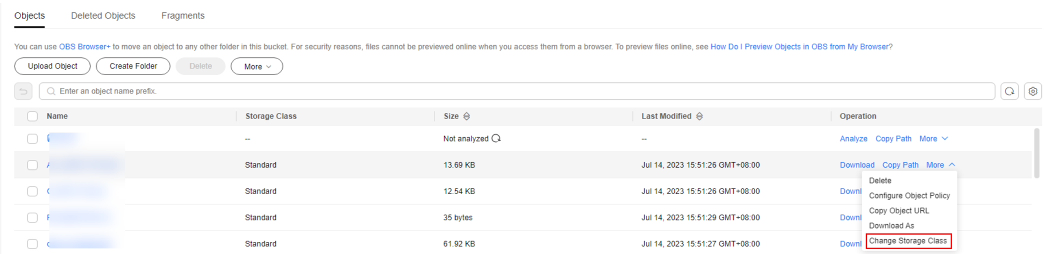
Step 2 In the bucket list, click the bucket name you want. The **Objects** page is displayed.

Step 3 Restore objects that are in the Archive or Deep Archive storage class if there are any. For details, see **Restoring an Object from Archive or Deep Archive Storage**.

Step 4 Change the storage class of objects individually or in a batch.

1. Individually: In the object list, locate the desired object and choose **More** > **Change Storage Class** in the **Operation** column on the right.

Figure 2-7 Changing object storage classes individually



You can use OBS Browser+ to move an object to any other folder in this bucket. For security reasons, files cannot be previewed online when you access them from a browser. To preview files online, see How Do I Preview Objects in OBS from My Browser?

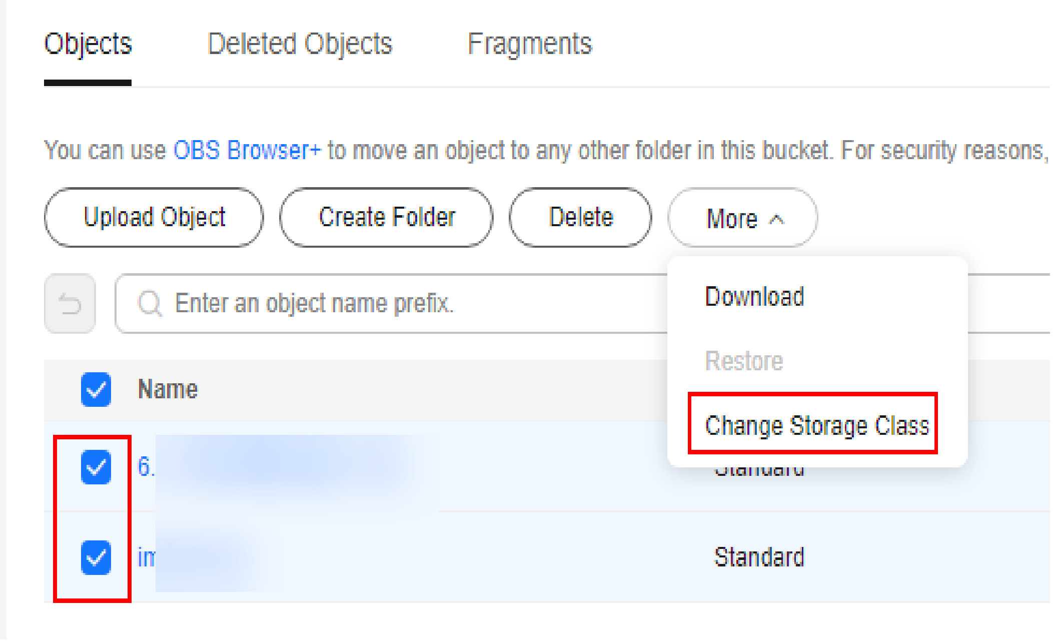
Name	Storage Class	Size	Last Modified	Operation
[redacted]	--	Not analyzed	--	Analyze Copy Path More
[redacted]	Standard	13.69 KB	Jul 14, 2023 15:51:26 GMT+08:00	Download Copy Path More
[redacted]	Standard	12.54 KB	Jul 14, 2023 15:51:26 GMT+08:00	Downl
[redacted]	Standard	35 bytes	Jul 14, 2023 15:51:29 GMT+08:00	Downl
[redacted]	Standard	61.92 KB	Jul 14, 2023 15:51:27 GMT+08:00	Downl

2. In a batch: In the object list, select the desired objects and choose **More** > **Change Storage Class** above the object list.

NOTE

Changing object storage classes in a batch is currently only available in some regions. If you cannot find the button for the batch operation, check whether this operation is supported in your region.

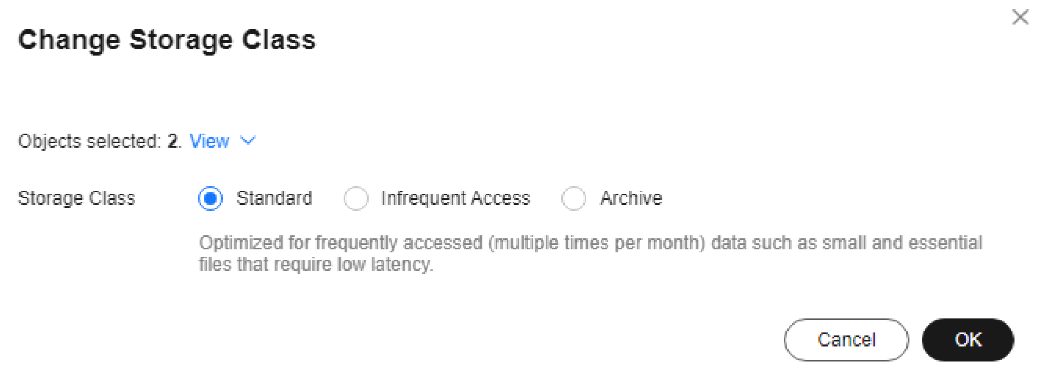
Figure 2-8 Changing object storage classes in a batch



You can use OBS Browser+ to move an object to any other folder in this bucket. For security reasons,

Step 5 Choose a new storage class and click **OK**.

Figure 2-9 Choosing a new storage class


Change Storage Class

Objects selected: 2. View

Storage Class ● Standard ○ Infrequent Access ○ Archive

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

----End

Using the API

OBS provides two APIs for you to manually change the storage class of objects. You can call an API to:

NOTE

If you want to change the storage class of Archive or Deep Archive objects, make sure you have **restored them** before you change their storage classes.

- **Modify object metadata** (adding the **x-obs-storage-class** header).
- **Copy an object** (adding the **x-obs-storage-class** header). If a bucket has versioning enabled, OBS automatically generates a unique version ID for a new object copy. If a bucket has versioning disabled or suspended, the existing object will be overwritten by a new object copy.

Using SDKs

NOTE

If you want to change the storage class of objects that are in the Archive or Deep Archive storage, make sure you have restored them by calling an API before you change their storage class.

Modifying object metadata	Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js
Copying objects	Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Click the name of the bucket that contains the object you want to modify the storage class for. The **Objects** page is displayed.

Step 3 If the storage class of the object to be modified is **Archive** or **Deep Archive**, right-click the object and choose **Restore** from the shortcut menu. After the restoration is complete, go to **Step 4**.

Step 4 Right-click the object to be modified and choose **Change Storage Class** from the shortcut menu.

Step 5 Select the desired storage class and click **OK**.

----End

Using the CLI Tool - obsutil

obsutil allows you to change the storage class of an object by modifying its properties.

NOTE

If you want to change the storage class of objects that are in the Archive or Deep Archive storage, make sure you have restored them before you change their storage classes. For commands for restoring objects, see **Restoring Objects**.

- In Windows

  - Setting properties of a single object

    ```
    obsutil chattri obs://bucket/key [-meta=aaa:bbb#ccc:ddd] [-sc=xxx] [-acl=xxx] [-aclXml=xxx] [-
    versionId=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```

  - Setting properties of objects in batches

    ```
    obsutil chattri obs://bucket[/key] -r [-f] [-v] [-meta=aaa:bbb#ccc:ddd] [-sc=xxx] [-acl=xxx] [-
    aclXml=xxx] [-o=xxx] [-j=1] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```

- In Linux or macOS
  - Setting properties of a single object

    ```
    ./obsutil chattri obs://bucket/key [-meta=aaa:bbb#ccc:ddd] [-sc=xxx] [-acl=xxx] [-aclXml=xxx] [-
    versionId=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```

  - Setting properties of objects in batches

    ```
    ./obsutil chattri obs://bucket[/key] -r [-f] [-v] [-meta=aaa:bbb#ccc:ddd] [-sc=xxx] [-acl=xxx] [-
    aclXml=xxx] [-o=xxx] [-j=1] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```

Examples

- Take the Windows OS as an example, run the **obsutil chattri obs://bucket-test/key -acl=public-read** command to set the access permission to an object to public read.

  ```
  obsutil chattri obs://bucket-test/key -acl=public-read

  Set the acl of object [key] in the bucket [bucket-test] to [public-read] successfully, request id
  [04050000016836DDFA73B2B5320E2651]
  ```

- Take the Windows OS as an example, run the **obsutil chattri obs://bucket-test -r -f -acl=public-read** command to set the access permission to all objects in the bucket to public read.

  ```
  obsutil chattri obs://bucket-test -r -f -acl=public-read

  [--------------------------------------------------] 100.00% tps:155.15 5/5 233ms
  Succeed count is:   5         Failed count is:    0
  Metrics [max cost:177 ms, min cost:53 ms, average cost:102.40 ms, average tps:20.41]
  Task id is: 9d7f73ff-f747-4fdd-9b2a-815ba2dc3b07
  ```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
key	Mandatory when setting properties for a single object. Optional when setting properties for a batch of objects.	The name of a single object whose properties are to be set, or the name prefix of a batch of objects whose properties are to be set. **NOTE** If this parameter is left blank for a batch operation, properties of all objects in a bucket are set.

Parameter	Optional or Mandatory	Description
meta	Optional (additional parameter)	Standard or custom metadata specified for object copies in object replication. This parameter should be configured in the following format: *key1*:*value1*#*key2*:*value2*#*key3*:*value3*. **NOTE** 1. This parameter takes effect only when it is used together with **direct**. 2. The format example above indicates that the object copy contains three pairs of custom metadata: *key1:value1*, *key2:value2*, and *key3:value3*. 3. Standard metadata headers include **Content-Type**, **Content-Encoding**, **Cache-Control**, **Content-Disposition**, **Content-Language** and **Expires**.
direct	Optional (additional parameter)	Metadata operation indicator. The value can be **REPLACE_NEW** or **REPLACE**. **REPLACE_NEW**: The existing metadata value is replaced with a new one, the metadata lacking a value is assigned one, and the metadata not specified remains unchanged. **REPLACE**: The metadata is replaced with the header included in the current request and the metadata not specified is deleted. **NOTE** This parameter takes effect only when it is used together with **meta**.

Parameter	Optional or Mandatory	Description
sc	Optional (additional parameter)	Storage class of an object. Possible values are: ● **standard**: Standard storage class. It features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month) or data that is smaller than 1 MB. ● **warm**: Infrequent Access storage class. It is ideal for storing infrequently accessed (less than 12 times a year) data, but when needed, the access has to be fast. ● **cold**: Archive storage class. It provides secure, durable, and inexpensive storage for rarely-accessed (once a year) data. ● **deep-archive**: Deep Archive storage class (under a beta test). It is suitable for storing data that is barely (once every few years) accessed. This storage class costs less than the Archive storage class, but takes longer (usually several hours) to restore data. **NOTE** For an object whose storage class is **cold**, restore the object first and then configure its storage class.
acl	Optional (additional parameter)	Access control policies that can be specified for objects. Possible values are: ● private ● public-read ● public-read-write ● bucket-owner-full-control **NOTE** These four values indicate, respectively, private read and write, public read, public read and write, and bucket owner full control.

Parameter	Optional or Mandatory	Description

aclXml	Optional (additional parameter)	Access control policy of the bucket, in XML format.
		<AccessControlPolicy>
		  <Owner>
		    <ID>*ownerid*</ID>
		  </Owner>
		  <AccessControlList>
		    <Grant>
		      <Grantee>
		        <ID>*userid*</ID>
		      </Grantee>
		      <Permission>*[WRITE|WRITE_ACP|READ|READ_ACP|FULL_CONTROL]*</Permission>
		    </Grant>
		    <Grant>
		      <Grantee>
		        <Canned>Everyone</Canned>
		      </Grantee>
		      <Permission>*[WRITE|WRITE_ACP|READ|READ_ACP|FULL_CONTROL]*</Permission>
		    </Grant>
		  </AccessControlList>
		</AccessControlPolicy>
		NOTE
		● **Owner**: Optional. Specify the object owner's ID.
		● Under **AccessControlList**, **Grant** contains the authorized users. **Grantee** specifies the IDs of authorized users. **Canned** specifies the authorized user group (currently, only **Everyone** is supported).
		● The following permissions can be granted: WRITE (write), WRITE_ACP (write ACL), READ (read), READ_ACP (read ACL), and FULL_CONTROL (full control).
		NOTICE
		Because angle brackets (<) and (>) are unavoidably included in the parameter value, you must use quotation marks to enclose them for escaping when running the command. Use single quotation marks for Linux and macOS and double quotation marks for Windows.

fs	Optional (additional parameter)	Specifies whether the method of listing parallel file systems is applied. If you are listing parallel file systems, you are recommended to add this parameter. **CAUTION** ● With this method, the listing time required varies greatly depending on the directory structure. ● When this parameter is enabled, **marker** and **limit** will be ignored. Then, the size of the bucket, parallel file system, or directory will be calculated. ● This parameter is only supported for obsutil 5.5.12 or later.

Parameter	Optional or Mandatory	Description
versionId	Optional when setting properties of a single object (additional parameter)	Version ID of the object whose properties are to be set
fr	Optional when setting properties of a single object (additional parameter)	Generates an operation result list when setting properties of a single object.
f	Optional when setting properties of objects in batches (additional parameter)	Runs in force mode.
r	Mandatory when setting properties of objects in batches (additional parameter)	Sets properties of objects in batches based on a specified object name prefix.
v	Optional when setting properties of objects in batches (additional parameter)	Sets properties of versions of objects in batches based on a specified object name prefix.

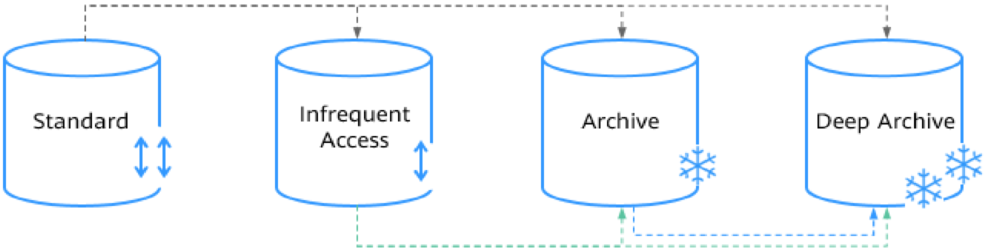
Parameter	Optional or Mandatory	Description
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (including success and failure files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** - The naming rule for result lists is as follows: **chattri_{succeed | failed}_report_*time*_TaskId.txt** By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring, respectively, **recordMaxLogSize** and **recordBackups** in the configuration file. - If there are multiple folders and files and you need to confirm the detailed error information about a failed task, refer to the failure list **chattri_failed_report_*time*_TaskId.txt** in the result list folder and the **log files** in the log path.
j	Optional when setting properties of objects in batches (additional parameter)	Indicates the maximum number of concurrent tasks for setting object properties in batches. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The tool ensures that the value is at least 1.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **Parameter Description**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

Changing the Storage Class of an Object Using Lifecycle Rules

For optimal storage costs, you can use lifecycle management to automatically change the storage classes of objects. Because multi-AZ redundancy is not supported for Archive and Deep Archive, you cannot use lifecycle rules to change the storage class of multi-AZ objects to Archive or Deep Archive. For more information, see **Transitioning Object Storage Classes Through Lifecycle Rules**.

If versioning is disabled, the lifecycle of an object starts when it is uploaded. If versioning is enabled, the lifecycle of a new object starts when it is uploaded. The lifecycle for historical objects starts when they become historical.

Figure 2-10 Changing the storage class of an object using lifecycle rules



As shown in the figure, OBS allows you to use lifecycle rules to change the storage class of objects:

- From Standard to Deep Archive storage, Archive, or Infrequent Access.
- From Infrequent Access to Archive or Deep Archive.
- From Archive to Deep Archive.

NOTE

- When changing the storage class of objects that are in the Infrequent Access, Deep Archive, or Archive storage, there may be additional charges for the **minimum storage duration**.

You can use OBS Console, APIs, SDKs, or OBS Browser+ to configure storage classes for buckets at bucket creation.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

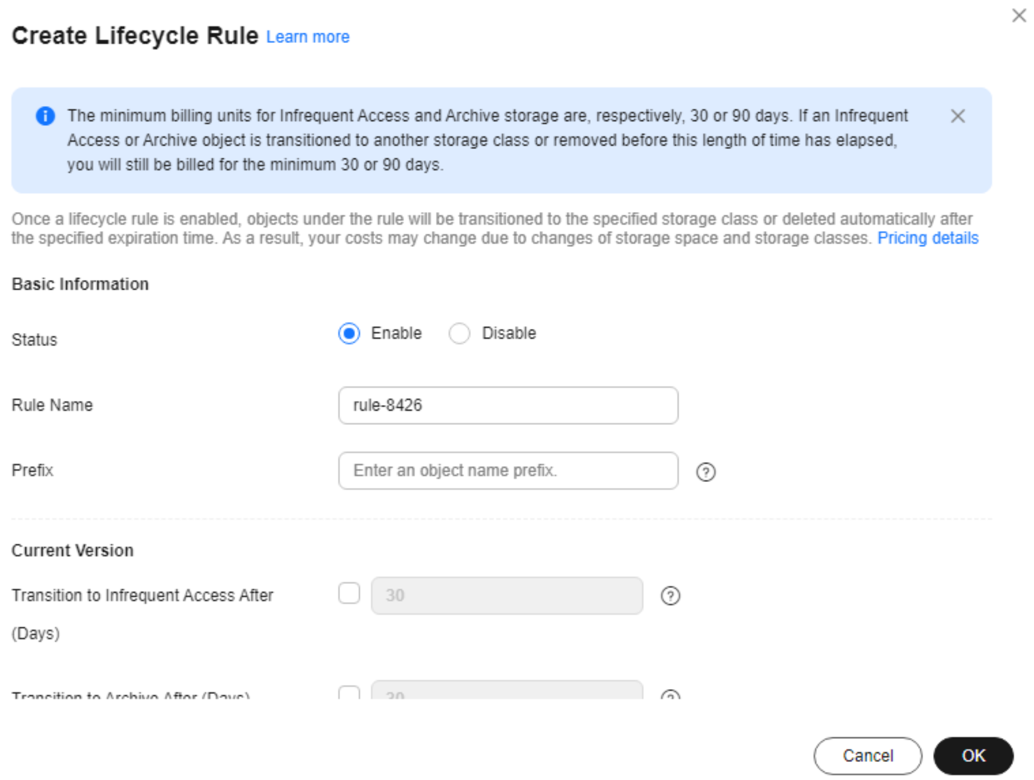
Step 3 In the navigation pane, choose **Overview**.

Step 4 In the **Basic Configurations** area, click **Lifecycle Rules**. The **Lifecycle Rules** page is displayed.

Alternatively, you can choose **Basic Configurations** > **Lifecycle Rules** in the navigation pane.

Step 5 Click **Create**. A dialog box as shown in **Figure 2-11** is displayed.

Figure 2-11 Creating a lifecycle rule



Create Lifecycle Rule Learn more

The minimum billing units for Infrequent Access and Archive storage are, respectively, 30 or 90 days. If an Infrequent Access or Archive object is transitioned to another storage class or removed before this length of time has elapsed, you will still be billed for the minimum 30 or 90 days.

Once a lifecycle rule is enabled, objects under the rule will be transitioned to the specified storage class or deleted automatically after the specified expiration time. As a result, your costs may change due to changes of storage space and storage classes. Pricing details

Basic Information

Status: ● Enable ○ Disable

Rule Name: rule-8426

Prefix: Enter an object name prefix.

Current Version

Transition to Infrequent Access After (Days): □ 30

Transition to Archive After (Days): □ 30

Step 6 Configure a lifecycle rule.

Basic Information:

- **Status**: Select **Enable** to enable this lifecycle rule after the configuration.
- **Rule Name**: It identifies a lifecycle rule. The rule name must be no longer than 255 characters.
- **Prefix**: It is optional.
  - If this field is configured, objects with the specified prefix will be managed by the lifecycle rule. The prefix cannot start with a slash (/) or contain two consecutive slashes (//), and cannot contain the following special characters: **\:*?"<>|**
  - If this field is not configured, all objects in the bucket will be managed by the lifecycle rule.

NOTE

- If **Object name prefix** is selected and the specified prefix and the prefix in an existing lifecycle rule overlap, OBS regards the two rules as one and forbids you to configure the current rule. For example, if there is already a rule with prefix **abc** in OBS, you cannot configure another rule whose prefix starts with **abc**.
- If there is already a lifecycle rule based on an object prefix, you are not allowed to configure another rule that is applied to the entire bucket.

Current Version or **Historical Version**:

NOTE

- **Current Version** and **Historical Version** are two concepts related to **Versioning**. If versioning is enabled for a bucket, uploading objects with the same name to the bucket creates different object versions. An object uploaded most recently is called **Current Version**, while an object uploaded previously is called **Historical Version**. For details, see **Versioning**.
- Either the **Current Version** or **Historical Version** must be configured, or you can configure both of them.

- **Transition to Infrequent Access After (Days)**: After a specified number of days have passed since the last update, objects meeting specified conditions will be transitioned to Infrequent Access. This number must be at least 30.
- **Transition to Archive After (Days)**: After a specified number of days have passed since the last update, objects meeting specified conditions will be transitioned to Archive. If you configure objects to transition first to Infrequent Access and then to Archive, make sure the objects stay in Infrequent Access at least 30 days before they are transitioned to Archive. There are, however, no such constraints on time if you configure objects to transition to only Archive.
- **Transition to Deep Archive After (Days)**: After a specified number of days have passed since the last update, objects meeting specified conditions will be transitioned to Deep Archive. If you configure objects to transition first to Infrequent Access and then to Deep Archive, make sure the objects stay in Infrequent Access at least 30 days before they are transitioned to Deep Archive. If you configure objects to transition first to Archive and then to Deep Archive, make sure the objects stay in Archive at least 90 days before they are transitioned to Deep Archive. There are, however, no such constraints on time if you configure objects to transition to only Deep Archive.
- **Expiration Time**: After a specified number of days have passed since the last update, objects meeting specified rules will be automatically expired and deleted. This number must be an integer larger than that specified for any of the transition operations.
- **Delete Fragments After (Days)**: After a specified number of days have passed since the fragment generation, OBS will automatically delete fragments in the bucket.

Assume you stored the following files in OBS on January 7, 2015:

- log/test1.log
- log/test2.log
- doc/example.doc
- doc/good.txt

Then, you stored the following files in OBS on January 10, 2015:

- log/clientlog.log
- log/serverlog.log
- doc/work.doc
- doc/travel.txt

If you created a rule to delete the objects prefixed with **log** one day after the last update on January 10, 2015, the following situations may occur:

- Objects **log/test1.log** and **log/test2.log** uploaded on January 7, 2015 might be deleted upon the next system scan. The deletion could happen on January 10, 2015 or January 11, 2015, depending on the time of the system scan.

- For objects **log/clientlog.log** and **log/serverlog.log** uploaded on January 10, 2015, each system scan would determine whether one full day had passed since their last update. If any scan determined one full day had passed, those objects would be deleted upon that scan. The deletion might happen on January 11, 2015 or January 12, 2015.

Suppose you configured the objects with the **log** prefix to be transitioned to Infrequent Access 30 days and to Archive 60 days and to be deleted 100 days after their last update. OBS would perform those actions on **log/clientlog.log**, **log/serverlog.log**, **log/test1.log**, and **log/test2.log** as you defined.

NOTE

After an object is uploaded, OBS calculates its lifecycle from 00:00 UTC the next day. It takes 24 hours at most to execute a lifecycle rule. As a result, there may be a delay in transitioning objects between storage classes and deleting expired objects. Generally, the delay does not exceed 48 hours. If you make changes to an existing lifecycle rule, when the changed rule takes effect will be recalculated.

Step 7 Click **OK** to complete the lifecycle rule configuration.

----End

Using the API

Configuring Bucket Lifecycle Rules

Using SDKs

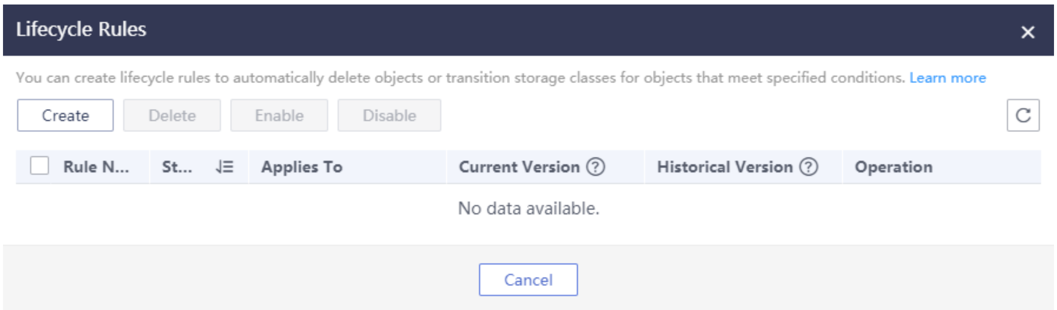
Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Select the bucket you want to configure a lifecycle rule and choose **More** > **Lifecycle Rules**.

Figure 2-12 Lifecycle Rules

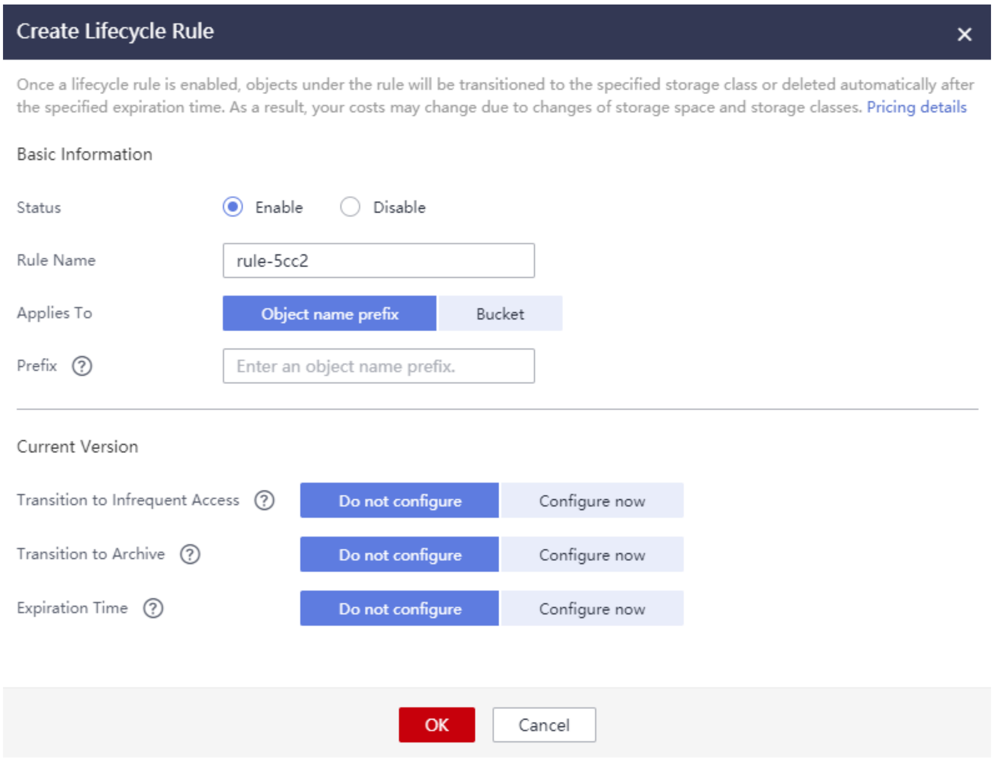


You can create lifecycle rules to automatically delete objects or transition storage classes for objects that meet specified conditions. Learn more

□ Rule N...	St...	Applies To	Current Version ?	Historical Version ?	Operation
No data available.					

Step 3 Click **Create**. The window shown in **Figure 2-13** is displayed.

Figure 2-13 Creating a lifecycle rule



Create Lifecycle Rule ×

Once a lifecycle rule is enabled, objects under the rule will be transitioned to the specified storage class or deleted automatically after the specified expiration time. As a result, your costs may change due to changes of storage space and storage classes. Pricing details

Basic Information

Status ● Enable ○ Disable

Rule Name rule-5cc2

Applies To Object name prefix | Bucket

Prefix ⓘ Enter an object name prefix.

Current Version

Transition to Infrequent Access ⓘ Do not configure | Configure now

Transition to Archive ⓘ Do not configure | Configure now

Expiration Time ⓘ Do not configure | Configure now

OK Cancel

Step 4 Configure related parameters.

- **Status**: Select **Enable** to enable this lifecycle rule after the configuration.
- **Rule Name**: Enter a rule name that is no longer than 255 characters.
- **Applies To**: By selecting **Object name prefix**, the lifecycle rule will apply to objects with the specified prefix contained in their name. You can also select **Bucket** for the lifecycle rule to apply to all objects in the bucket.

NOTE

- If **Object name prefix** is selected, and the specified prefix and the prefix in an existing lifecycle rule overlap, OBS regards the two rules as one and forbids you to configure the current rule. For example, if there is a rule with prefix **abc** in the system, another rule whose prefix contains **abc** cannot be configured.
- If there is already a lifecycle rule based on an object prefix, you are not allowed to configure another rule that is applied to the entire bucket.
- If there is already a lifecycle rule whose **Applies To** is set to **Bucket**, you are not allowed to configure a new rule whose **Applies To** is set to **Object name prefix**.

- You can use a lifecycle rule to specify the number of days after the last update objects that meet specified criteria are automatically transitioned to the Infrequent Access or Archive storage class, or are automatically deleted.
  - **Transition to Infrequent Access**: This rule transitions the objects meeting the conditions to the Infrequent Access storage class the specified number of days after the last update.
  - **Transition to Archive**: This rule transitions the objects meeting the conditions to the Archive storage class the specified number of days after the last update.

- **Expiration Time**: This determines when an object will expire and then be deleted, or the day after which objects matching the rule will be deleted.

Assume you stored the following files in OBS on January 7, 2022:

- log/test1.log
- log/test2.log
- doc/example.doc
- doc/good.txt

On January 10, 2022, you stored the following files in OBS:

- log/clientlog.log
- log/serverlog.log
- doc/work.doc
- doc/travel.txt

  Suppose you configured a rule on January 10, 2022 to expire and delete objects with the **log/** prefix one day after their upload, objects **log/test1.log**, **log/test2.log**, **log/clientlog.log**, and **log/serverlog.log** would deleted on January 12, 2022.

  If you configure lifecycle rules that transition objects with the **log** prefix to Infrequent Access 30 days and to Archive 60 days after the upload and delete them 100 days after the upload. OBS performs those actions on **log/clientlog.log**, **log/serverlog.log**, **log/test1.log**, and **log/test2.log** one day after the defined transition periods end.

Step 5 Click **OK** to save the lifecycle rule.

----End

Precautions

- Minimum billable object size

  Object smaller than 64 KB are regarded as 64 KB in size.

- Minimum storage duration

  The minimum storage duration refers to the least time that is eligible for billing. This means that objects will be billed for the minimum storage duration even if they are not stored for that long. For example, if an object is transitioned to Archive after being stored in Infrequent Access for 20 days, it will be billed for the storage of 30 days (the minimum storage duration for Infrequent Access).

Item	Standard	Infrequent Access	Archive	Deep Archive (in OBT)
Minimum storage duration	N/A	30 days	90 days	180 days

- Object restoration duration

Object restoration is required for accessing Archive and Deep Archive objects. The restoration will take some time. If your services require real-time data access, these two storage classes are not recommended.

Table 2-3 Object restoration duration

Restoration Speed	Duration of Restoration from Archive	Duration of Restoration from Deep Archive
Standard	3–5 h	5–12 h
Expedited	1–5 min	3–5 h

- Data restoration charges

Table 2-4 Data restoration charges

Operation	Billing Item	Description
Infrequent Access objects	Requests	You are billed for the number of successfully restored objects. If *N* objects were successfully restored, you are billed for *N* requests.
	Data transfer	You are billed for the bandwidth you consumed during data retrievals.
Archive or Deep Archive objects	Requests	You are billed for the number of successfully restored objects. If *N* objects were successfully restored, you are billed for *N* requests.
	Data transfer	You are billed for the bandwidth you consumed during data retrievals.
	Temporary file storage	When an Archive or a Deep Archive object is restored, a copy in the Standard storage class will be generated for the object. That means the object and its Standard copy will co-exist in the bucket. During the validity period of a restoration, you will be billed for the space taken up by both the object and its copy. The copy will be automatically deleted once the validity period ends.

3 Bucket Management

3.1 Bucket Overview

A bucket is a virtual container used to store **objects** in OBS. OBS offers a flat structure based on buckets and objects. This structure enables all objects to be stored at the same logical layer, rather than being stored hierarchically.

Buckets have their own attributes, such as **storage classes**, access permissions, and **regions**. You can specify access permissions, a storage class, and a region when creating a bucket. You can also configure advanced attributes to fit different storage requirements.

OBS provides the following storage classes: Standard, Infrequent Access, and Archive. With support for these storage classes, OBS caters to diverse storage performance and cost requirements. When creating a bucket, you can specify a storage class for the bucket, which can be changed later.

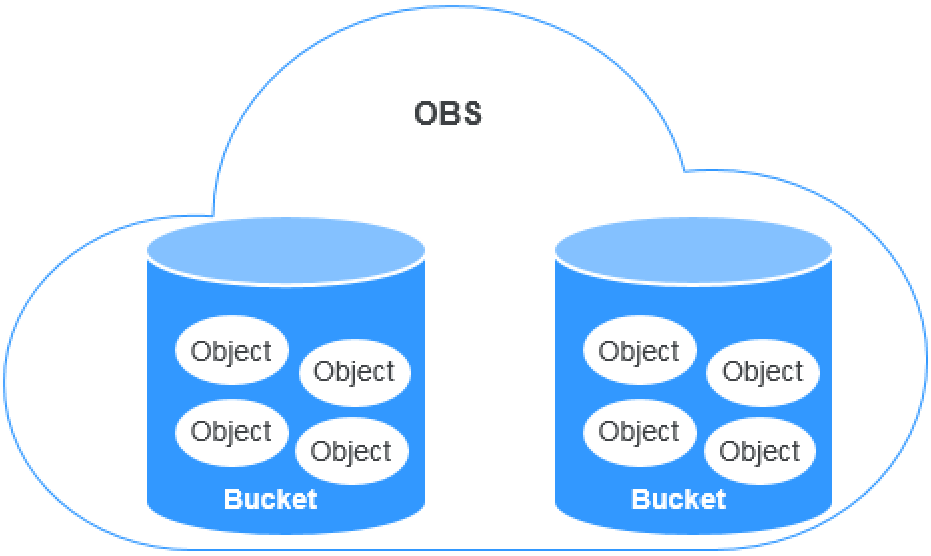
On OBS, each bucket name must be unique and cannot be changed. The region where a bucket resides cannot be changed once the bucket is created. When you create a bucket, OBS creates a default access control list (ACL) that grants the authorized user permissions on the bucket. Only authorized users can perform operations such as creating, deleting, viewing, and configuring buckets.

An account (including all IAM users under the account) can create a maximum of 100 buckets. You can leverage the fine-grained permission control capability of OBS to properly plan and use buckets. For example, you can create folders in a bucket based on object prefixes and use **fine-grained permission control** to isolate data between different departments. There is no limit on the number and total size of objects in a bucket.

As OBS is based on a RESTful architecture over HTTP and HTTPS, you can use uniform resource locators (URLs) to locate resources.

Figure 3-1 illustrates the relationship between buckets and objects in OBS.

Figure 3-1 Buckets and objects



You can use different methods to **create buckets** based on your use habits and storage needs. After a bucket is created, you can **upload files (data) to the bucket** in a way you like. OBS stores these files as objects. In OBS, buckets and objects are located in different regions, but you can access the same bucket and its resources in a region regardless of the method you choose to access OBS.

3.2 Creating a Bucket

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to create a bucket. A bucket is a container that stores objects in OBS. Before you can store data in OBS, you must create a bucket.

Prerequisites

To create a bucket, make sure you have a registered account, sufficient account balance, access keys (AK and SK), and the endpoint information. For details, see **Getting Started**.

Constraints

- After a bucket is created, its name and region cannot be changed. Make sure that the bucket name and region you set are appropriate.
- An account (including all IAM users under the account) can create a maximum of 100 buckets. You can use the fine-grained access control of OBS to properly plan and use buckets. For example, you can create folders in a bucket based on object prefixes and use **fine-grained permission control** to isolate permissions between departments. There is no limit on the number and total size of objects in a bucket.
- A bucket name is part of the access domain name and needs to be resolved. Therefore, the bucket name must conform to the **DNS domain naming rules**. When receiving a bucket creation request, OBS strictly checks the bucket name. A bucket name:

- Must be unique across all accounts and regions. The name of a deleted bucket can be reused for another bucket or a parallel file system at least 30 minutes after the deletion.
- Must be 3 to 63 characters long. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
- Cannot start or end with a period (.) or hyphen (-), and cannot contain two consecutive periods (..) or contain a period (.) and a hyphen (-) adjacent to each other.
- Cannot be formatted as an IP address.

NOTICE

If you use a bucket with periods (.) in its name to access OBS, the client will display a message indicating that the bucket is risky, for example, a red alarm may be displayed in the browser security prompt. This is because the SSL wildcard certificate matches only buckets without periods (.) in their names when HTTPS is used for OBS access. We recommend that you avoid using periods (.) in bucket names.

Ways to Create a Bucket

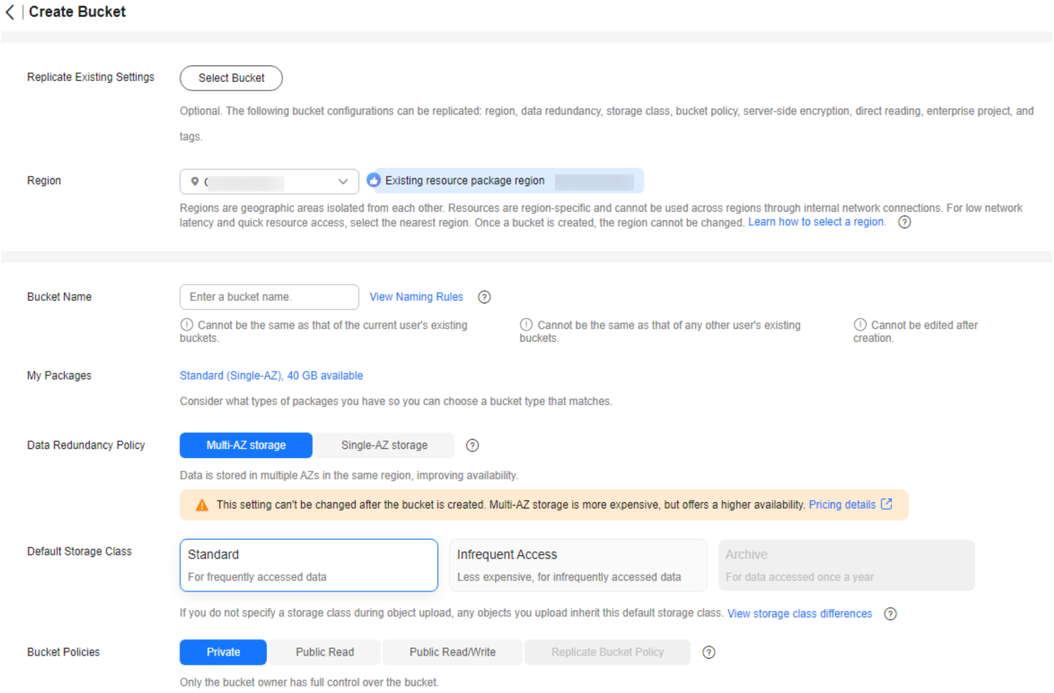
You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to create a bucket.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the upper right corner, click **Create Bucket**. The **Create Bucket** page is displayed.

Figure 3-2 Creating a bucket



Create Bucket

Replicate Existing Settings: Select Bucket

Optional. The following bucket configurations can be replicated: region, data redundancy, storage class, bucket policy, server-side encryption, direct reading, enterprise project, and tags.

Region: [redacted] Existing resource package region [redacted]

Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region. Once a bucket is created, the region cannot be changed. Learn how to select a region.

Bucket Name: Enter a bucket name. View Naming Rules

Cannot be the same as that of the current user's existing buckets.

Cannot be the same as that of any other user's existing buckets.

Cannot be edited after creation.

My Packages: Standard (Single-AZ), 40 GB available

Consider what types of packages you have so you can choose a bucket type that matches.

Data Redundancy Policy: ● Multi-AZ storage ○ Single-AZ storage

Data is stored in multiple AZs in the same region, improving availability.

This setting can't be changed after the bucket is created. Multi-AZ storage is more expensive, but offers a higher availability. Pricing details

Default Storage Class: ● Standard (For frequently accessed data) ○ Infrequent Access (Less expensive, for infrequently accessed data) ○ Archive (For data accessed once a year)

If you do not specify a storage class during object upload, any objects you upload inherit this default storage class. View storage class differences

Bucket Policies: ● Private ○ Public Read ○ Public Read/Write ○ Replicate Bucket Policy

Only the bucket owner has full control over the bucket.

Step 3 Configure bucket parameters.

Table 3-1 Bucket parameters

Parameter	Description
Replicate Existing Settings	Optional. To use this function, click **Select Bucket** and select a bucket from the list as the replication source. After the replication source is selected, the following settings are replicated to the bucket you are creating: region, data redundancy policy, storage class, bucket policy, server-side encryption, direct reading, enterprise project, and tags. You can still change some or all of the replicated settings as needed.
Region	Region where the bucket is located. For low latency and faster access, select the region nearest to you. Once the bucket is created, its region cannot be changed. Most OBS features are available in all regions, but some are only available for certain regions. Consider the feature availability in each region when you select a region for a bucket. For details, see **Function Overview**. If your ECS needs to access an OBS bucket over the intranet, ensure that the bucket and the ECS are in the same region. For details, see **Accessing OBS over an Intranet**.
Bucket Name	Name of the bucket. A bucket name must be unique across all accounts and regions. Once a bucket is created, its name cannot be changed. According to the globally applied DNS naming rules, an OBS bucket name: ● Must be unique across all accounts and regions. The name of a deleted bucket can be reused for another bucket or a parallel file system at least 30 minutes after the deletion. ● Must be 3 to 63 characters long. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. ● Cannot start or end with a period (.) or hyphen (-), and cannot contain two consecutive periods (..) or contain a period (.) and a hyphen (-) adjacent to each other. ● Cannot be formatted as an IP address. **NOTE** When you access OBS through HTTPS using virtual hosted-style URLs, if the bucket name contains a period (.), the certificate verification will fail. To work around this issue, you are advised not to use periods (.) in bucket names.

Parameter	Description
Data Redundancy Policy	● **Multi-AZ storage**: Data is stored in multiple AZs to achieve higher reliability. ● **Single-AZ storage**: Data is stored in a single AZ, with lower costs. For details about the performance comparison between multi-AZ and single-AZ storage, see **Comparison of Storage Classes**. Once a bucket is created, the data redundancy policy cannot be changed, so choose the policy that can meet your needs. ● Multi-AZ storage is not available for buckets in the Archive storage class.
Default Storage Class	Storage classes of a bucket. These storage classes can meet different needs for storage performance and costs. ● The Standard storage class is for storing a large number of hot files or small files that are frequently accessed (multiple times per month on average) and require quick retrieval. ● The Infrequent Access storage class is for storing data that is less frequently accessed (less than 12 times per year on average) and requires quick retrieval. ● The Archive storage class is for archiving data that is rarely accessed (once a year on average) and has no requirements for quick retrieval. For details, see **Storage Classes**.
Bucket Policy	Controls read and write permissions for buckets. ● **Private**: No access beyond the bucket ACL settings is granted. ● **Public Read**: Anyone can read objects in the bucket. ● **Public Read and Write**: Anyone can read, write, or delete objects in the bucket.

Parameter	Description
Server-Side Encryption	Choose **SSE-KMS**. ● You can choose **Default** to use the default key in the current region to encrypt the objects you upload to the bucket. If you do not have a default key, OBS automatically creates one the first time you upload an object. ● You can also choose **Custom** to use a custom key for encryption. If there is no custom key available, click **View KMS Keys** to create one. ● You can also select **Shared Key** to enter a shared key ID. The key shared by other users will be used to encrypt your objects. For details about how to obtain a shared key ID, see **Viewing a CMK**. **NOTE** A shared key from **a project or a subproject** can be configured here. However, if a shared key from a subproject is specified, the owner of the shared key cannot access objects encrypted with that key, but the bucket owner can. When **SSE-OBS** is chosen, the keys created and managed by OBS are used for encryption. When Server-Side Encryption is enabled for a bucket, you can configure the object you upload to inherit encryption from the bucket or choose SSE-KMS or SSE-OBS.
WORM	When you enable write-once-read-many (WORM), you can configure a retention policy for the current bucket. The object version which the retention policy is applied to cannot be deleted within a specified period. You can only enable WORM when you create a bucket. Once enabled for a bucket, WORM cannot be disabled. When you enable WORM, OBS automatically enables versioning for the bucket, and versioning cannot be suspended later for that bucket.
Direct Reading	Direct reading allows you to directly download objects from the Archive storage class without restoring them first. Direct reading is a billable function. For details, see **Product Pricing Details**. No matter which default storage class you select, you can enable direct reading for your bucket. For example, if you select the Standard storage class and enable direct reading for your bucket, you can directly download objects stored in the Archive storage class from your bucket.

Parameter	Description
Enterprise Project	You can add a bucket to an enterprise project for unified management. Create an enterprise project by referring to **Creating an Enterprise Project**. The default enterprise project is named **default**. On the **Enterprise Project Management** page, **create an enterprise project**, and **add a user group to the enterprise project**. By doing so, users in this user group obtain the operation permissions for the buckets and objects in the enterprise project. **NOTE** Only an enterprise account can configure enterprise projects. OBS ReadOnlyAccess and OBS OperateAccess are the fine-grained authorizations of the enterprise project user group in OBS.
Tags	Optional. Tags are used to identify and classify buckets in OBS. Each tag is represented by a key-value pair. For more information, see **Adding Tags to a Bucket**.

Step 4 Click **Create Now**.

----End

Using the API

Creating a Bucket

Using SDKs

Java	**Python**	**C**	**Go**	BrowserJS: not supported	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**

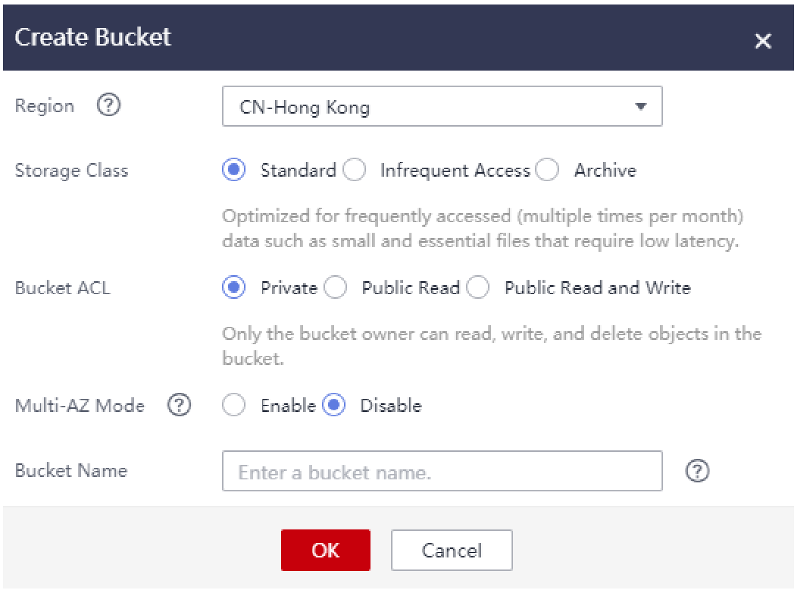
Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 In the upper part of the page, click **Create Bucket**.

Step 3 In the displayed dialog box, configure bucket parameters, as shown in **Figure 3-3**.

Figure 3-3 Creating a bucket



Create Bucket

Region: CN-Hong Kong

Storage Class: ● Standard ○ Infrequent Access ○ Archive

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

Bucket ACL: ● Private ○ Public Read ○ Public Read and Write

Only the bucket owner can read, write, and delete objects in the bucket.

Multi-AZ Mode: ○ Enable ● Disable

Bucket Name: Enter a bucket name.

Table 3-2 Bucket creation parameters

Parameter	Description
Region	Enter the region where you want to create a bucket. Once the bucket is created, its region cannot be changed.
Storage Class	Storage class of the bucket. Different storage classes meet customers' needs for storage performance and costs. ● **Standard**: applicable to scenarios where a large number of hot files or small files need to be accessed frequently (multiple times per month on average) and require fast access response. ● **Infrequent Access**: ideal for storing data that is not frequently accessed (less than 12 times per year on average) but requires fast access response. ● **Archive**: suitable for archiving data that is rarely accessed (averagely once a year) and has no requirements for quick response. For more information, see **Storage Classes**.
Bucket ACL	Controls read and write permissions on buckets. ● **Private**: Only users granted permissions by the ACL can access the bucket. ● **Public Read**: Anyone can read objects in the bucket. ● **Public Read and Write**: Anyone can read, write, or delete objects in the bucket.

Parameter	Description
Multi-AZ Mode	If multi-AZ is enabled, data will be stored in multiple AZs. ● Once a bucket is created, its multi-AZ status cannot be changed. Therefore, plan in advance and determine whether to enable the multi-AZ function during bucket creation.
Bucket Name	Name of the bucket you want to create, which must be globally unique. A bucket name: ● Must be 3 to 63 characters long and start with a digit or letter. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. ● Cannot be formatted as an IP address. ● Cannot start or end with a hyphen (-) or period (.). ● Cannot contain two consecutive periods (..), for example, **my..bucket**. ● Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, **my-.bucket** or **my.-bucket**.

You can click ⓘ next to the bucket name to learn about the bucket naming rules. A user can create a maximum of 100 buckets in OBS.

NOTE

- When a URL is used to access a bucket, the bucket name will become part of the URL. According to the DNS rule, URLs do not support uppercase letters and cannot recognize buckets whose name contains uppercase letters. Therefore, a bucket name can contain only lowercase letters, digits, hyphens (-), and periods (.) For example, if you attempt to access bucket **MyBucket** using a URL, the URL will parse **MyBucket** as **mybucket**. This results in an access error.
- DNS naming rules can standardize bucket names globally, facilitating the resolution during bucket access. With the DNS naming rules used, you can benefit from new functions and optimized features, and configure static website hosting for buckets.
- Once a bucket is created, its name cannot be changed. Make sure that the bucket name you set is appropriate.

Step 4 Click **OK**. If the bucket is successfully created, it is displayed in the bucket list. If the creation fails, an error message will be displayed.

----End

Using the CLI Tool - obsutil

Command Line Structure

- In Windows

```
obsutil mb obs://bucket [-fs] [-az=xxx] [-acl=xxx] [-sc=xxx] [-location=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

- In Linux or macOS

```
./obsutil mb obs://bucket [-fs] [-az=xxx] [-acl=xxx] [-sc=xxx] [-location=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

Examples

- Take the Windows OS as an example. Run the **obsutil mb obs://bucket-test** command to create a bucket. The creation is successful.

```
obsutil mb obs://bucket-test

Create bucket [bucket-test] successfully, request id [0000016979E1D2EA860BB5E80A6B8FCC]
```

- Take the Windows OS as an example. Run the **obsutil mb obs://bucket001** command to create a namesake bucket. The creation fails.

```
obsutil mb obs://bucket001

Create bucket [bucket001] failed, http status [409], error code [BucketAlreadyExists], error message
[The requested bucket name is not available. The bucket namespace is shared by all users of the
system. Please select a different name and try again.], request id
[04030000016757F31A0333281A6B1E92]
```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name **NOTE** A bucket name must comply with the following rules: ● Contains 3 to 63 characters, including lowercase letters, digits, hyphens (-), and periods (.), and starts with a digit or letter. ● Cannot be an IP address. ● Cannot start or end with a hyphen (-) or period (.). ● Cannot contain two consecutive periods (.), for example, **my..bucket**. ● Cannot contain periods (.) and hyphens (-) adjacent to each other, for example, **my-.bucket** or **my.-bucket**.
fs	Optional (additional parameter)	Creates a parallel file system.
az	Optional (additional parameter)	Specifies a bucket's data redundancy policy. Possible values are: ● multi-az **NOTE** If **multi-az** is used, a bucket with the multi-AZ storage policy will be created. If this parameter is not specified, a bucket with the single-AZ storage policy will be created.
acl	Optional (additional parameter)	Access control policies that can be specified when creating a bucket. Possible values are: ● private ● public-read ● public-read-write **NOTE** The preceding three values indicate private read and write, public read, and public read and write.

Parameter	Optional or Mandatory	Description
sc	Optional (additional parameter)	Default bucket storage class that can be specified when creating a bucket. Possible values are: ● **standard**: Standard storage class. It features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month) or data that is smaller than 1 MB. ● **warm**: Infrequent Access storage class. It is ideal for storing infrequently accessed (less than 12 times a year) data, but when needed, the access has to be fast. ● **cold**: Archive storage class. It provides secure, durable, and inexpensive storage for rarely-accessed (once a year) data. ● **deep-archive**: Deep Archive storage class (under limited beta testing). It is suitable for storing data that is barely (once every few years) accessed. This storage class costs less than the Archive storage class, but takes longer time (usually several hours) to restore data.
location	Mandatory unless the region where the OBS service resides is not the default region (additional parameter)	Region where the bucket resides. **NOTE** This parameter indicates the region where a bucket will be created. It is mandatory only when the endpoint belongs to any other regions than the default one CN North-Beijing1 (cn-north-1). To view the currently valid regions, see **Regions and Endpoints**.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.

Parameter	Optional or Mandatory	Description
t	Optional (additional parameter)	Specifies the user's security token.

Related Operations and FAQs

Changing the Bucket Storage Class

After the bucket is created, you can change its storage class by performing the following steps: For details, see **Configuring a Storage Class** and **Changing the Storage Classes of Buckets and Objects**.

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, locate the bucket you want and click **Change Storage Class** on the right.

Step 3 Select the desired storage class and click **OK**.

NOTE

- Changing the storage class of a bucket does not change the storage class of existing objects in the bucket.
- If you do not specify a storage class for an object when uploading it, it inherits the bucket's storage class by default. After the bucket's storage class is changed, newly uploaded objects will inherit the new storage class of the bucket by default.

----End

Accessing a Bucket

After a bucket is created, you can use the domain name to access the bucket. You can assemble the bucket domain name by putting the bucket name and endpoint together, or you can obtain it by **viewing the basic bucket information** on OBS Console or OBS Browser+.

An access domain name is structured as follows:

[Structure] *BucketName*.*Endpoint*

[Example] *bucketname*.**obs.ap-southeast-1.myhuaweicloud.com**

Causes of Bucket Creation Failures and Solutions

Why Am I Unable to Create a Bucket?

More FAQs

FAQ for Buckets and Objects

3.3 Replicating Settings from an Existing Bucket

Scenarios

You can replicate the settings of an existing bucket to the bucket you are creating.

The following configurations can be replicated:

- Bucket policies
- CORS rules
- Lifecycle rules
- Back-to-source rules
- Image processing styles
- Online decompression rules

Constraints

- The configurations replicated from a source bucket will not overwrite existing configurations in the destination bucket, and any that conflict with the existing ones will not be replicated.
- The version of both the source and destination buckets must be 3.0.
- For functions available for both buckets and parallel file systems, function configurations can be replicated from buckets to parallel file systems and vice versa.

Ways to Replicate Settings from an Existing Bucket

You can use OBS Console to replicate existing settings from an existing bucket.

Using OBS Console

- **Replicating Bucket Policies**
- **Replicating CORS Rules**
- **Replicating Lifecycle Rules**
- **Replicating Back-to-Source Rules**
- **Replicating Image Styles**
- **Replicating Online Decompression Policies**

3.4 Listing Buckets

You can list all created buckets to view their information.

Ways to List Buckets

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to list buckets.

Using OBS Console

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner and choose **Storage** > **Object Storage Service**.

Step 3 Go to the bucket list page, which displays all buckets under your account.

----End

Using the API

Listing Buckets

Using SDKs

Java	**Python**	**C**	**Go**	BrowserJS: not supported	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**

Using the GUI Tool - OBS Browser+

Log in to OBS Browser+. All buckets under your account are displayed in the bucket list.

Using the CLI Tool - obsutil

Command Line Structure

- In Windows

```
obsutil ls [-s] [-du] [-sc] [-j=1] [-limit=1] [-format=default] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

- In Linux or macOS

```
./obsutil ls [-s] [-du] [-sc] [-j=1] [-limit=1] [-format=default] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

Examples

- Take the Windows OS as an example. Run the **obsutil ls -limit=5** command to obtain the bucket list.

```
obsutil ls -limit=5

Bucket                  CreationDate              Location        BucketType
obs://bucket001            2018-09-03T01:53:02Z        example         OBJECT

obs://bucket002            2018-11-01T01:40:01Z        example         OBJECT

obs://bucket003            2018-10-25T11:45:45Z        example         OBJECT

obs://bucket004            2018-10-26T02:33:09Z        example         OBJECT

obs://bucket005            2018-10-26T02:34:50Z        example         OBJECT

Bucket number : 5
```

Parameter Description

Parameter	Optional or Mandatory	Description
s	Optional (additional parameter)	Displays simplified query result. **NOTE** In the simplified format, the returned result contains only the bucket name.
sc	Optional (additional parameter)	Queries the storage classes of the buckets when listing buckets.
j	Optional (additional parameter). It must be used together with **sc**.	Indicates the maximum number of concurrent tasks for querying the bucket storage class. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
du	Optional (additional parameter)	Quickly returns the total size of listed objects, without displaying detailed object information. This parameter can be used together with other parameters. **NOTE** This parameter takes effect only on listing objects, but not on listing buckets.
fs	Optional (additional parameter)	Specifies whether the method of listing parallel file systems is applied. If you are listing parallel file systems, you are recommended to add this parameter. **CAUTION** ● With this method, the listing time required varies largely depending on the directory structures. ● After this parameter is enabled, **marker** and **limit** will be ignored. Then, the size of the bucket or parallel file system will be calculated. ● This parameter is only supported by obsutil 5.5.12 and later.
limit	Optional (additional parameter)	Maximum number of buckets that can be queried. If the value is less than 0, all buckets are listed. If it is left blank, a maximum of 1000 buckets can be listed by default.

Parameter	Optional or Mandatory	Description
format	Optional (additional parameter)	Prints a listing result in the user-defined format. Currently, the value can only be **default**, indicating that the listing result is displayed in one row.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

NOTE

In a bucket listing result, **BucketType** indicates the bucket type. **OBJECT** indicates an object storage bucket, while **POSIX** indicates a parallel file system.

3.5 Viewing Bucket Information

After creating a bucket on OBS Console, you can view its details, including basic bucket information, usage statistics, alarms, domain name details, basic configurations, FAQs, and others. You can also export all buckets of the current account and view their basic information in the exported Excel file.

Ways to View Bucket Information

You can use OBS console, SDKs, OBS Browser+, obsutil, or APIs to view bucket information.

Using OBS Console

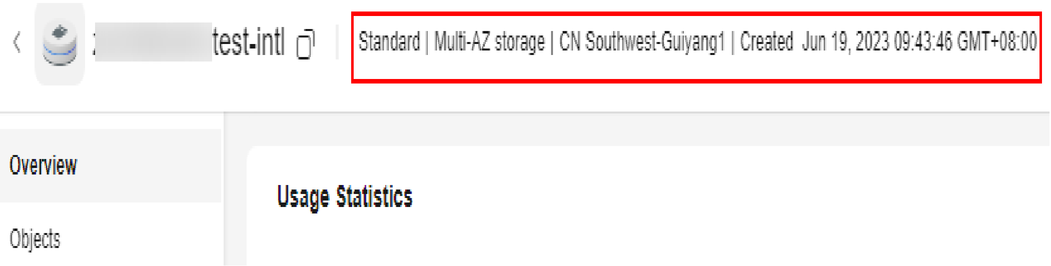
Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Overview**.

Step 4 On the top of the page, view the bucket information, including the bucket name, storage class, data redundancy policy, region, and creation time.

Figure 3-4 Bucket information



[redacted] test-intl

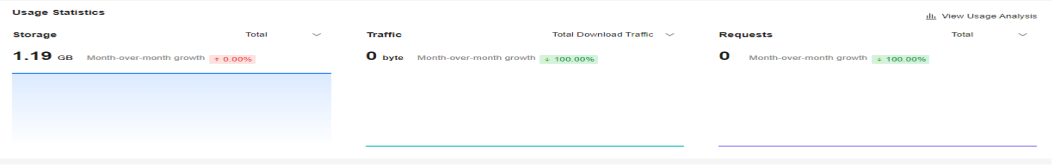
Standard | Multi-AZ storage | CN Southwest-Guiyang1 | Created Jun 19, 2023 09:43:46 GMT+08:00

Table 3-3 Bucket information

Parameter	Description
Bucket name	Name of the bucket
Storage class	Storage class of the bucket, which can be **Standard**, **Infrequent Access**, or **Archive**.
Data redundancy	Data redundancy storage policy of a bucket, which can be multi-AZ storage or single-AZ storage. This setting cannot be changed after the bucket is created.
Region	Region where the bucket is located
Created	Creation time of the bucket

Step 5 In the **Usage Statistics** area, view the storage, traffic, and request information of the bucket. Click **View Usage Analysis** in the upper right corner to go to the **Metrics** page and view details.

Figure 3-5 Bucket usage statistics



Usage Statistics

Storage: 1.19 GB, Month-over-month growth ↑ 0.00%

Traffic: 0 byte, Month-over-month growth ↓ 100.00%

Requests: 0, Month-over-month growth ↓ 100.00%

Table 3-4 Bucket usage metrics

Parameter	Description
Storage	Measures the storage occupied by all objects, Standard objects, Infrequent Access objects, and Archive objects in the bucket.
Traffic	**Total Download Traffic**: It measures the total download traffic for the bucket in the current month. Both intranet and Internet traffic are covered.

Parameter	Description
	Download Traffic (Intranet): It measures the total intranet download traffic for the bucket in the current month.
	Download Traffic (Internet): It measures the total Internet download traffic for the bucket in the current month.
	Total Upload Traffic: It measures the total upload traffic for the bucket in the current month. Both intranet and Internet traffic are covered.
	Upload Traffic (Intranet): It measures the total intranet upload traffic for the bucket in the current month.
	Upload Traffic (Internet): It measures the total Internet upload traffic for the bucket in the current month.
Requests	**Total**: It measures the total number of requests (including PUT, POST, COPY, LIST, GET, HEAD, and DELETE requests) made for the bucket and the objects in it in the current month.
	GET: It measures the total number of GET and HEAD requests made for the bucket and the objects in it in the current month.
	PUT: It measures the total number of PUT, POST, COPY, and LIST requests made for the bucket and the objects in it in the current month.
	DELETE: It measures the total number of DELETE requests made for the bucket and the objects in it in the current month.
Month-over-month growth	It compares the current month's data with the previous month's data, showing the data increase or decrease. Take the comparison between January 2023 and February 2023 as an example. Month-over-month (MoM) growth = (Current month's data – Previous month's data)/Previous month's data x 100% Suppose the Standard storage in January was 60 MB and that in February was 120 MB, the MoM growth was calculated as follows: (120 – 60) ÷ 60 x 100% = 100%. This tells that the standard storage in February was doubled compared to January.

NOTE

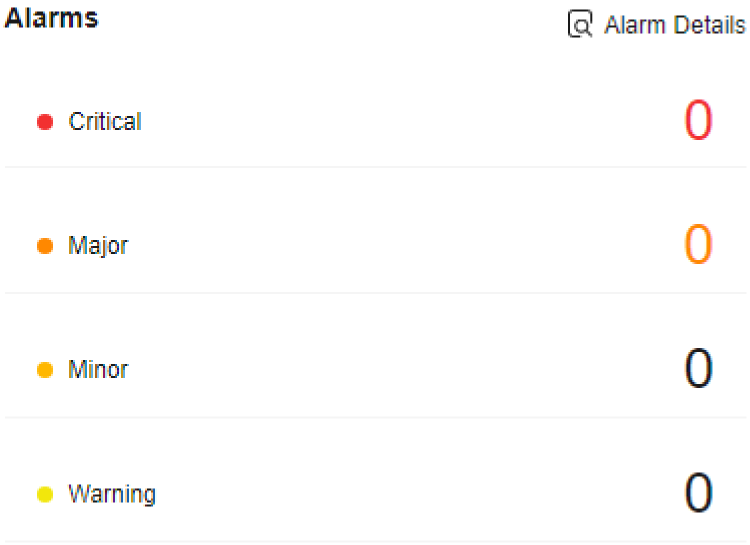
The data is not displayed in real time. There may be approximately one hour delay.

To view the number of requests and traffic statistics, you must have the **CES ReadOnlyAccess** permission or a higher Cloud Eye permission for the region where the bucket is located.

Usage Statistics is only available for buckets that support usage analysis.

Step 6 In the **Alarms** area, view the alarm severities and the alarm number of each severity. By clicking **Alarm Details**, you can explore more on the **Alarm Records** page of Cloud Eye.

Figure 3-6 Information about bucket alarms



Alarms

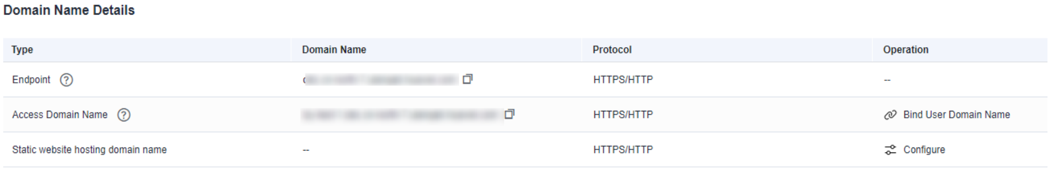
● Critical	0
● Major	0
● Minor	0
● Warning	0

NOTE

Alarms is only available for buckets that support usage analysis.

Step 7 In the **Domain Name Details** area, view information about the endpoint, access domain name, and static website hosting domain name. You can also perform related operations by clicking buttons in the **Operation** column.

Figure 3-7 Domain name details of the bucket

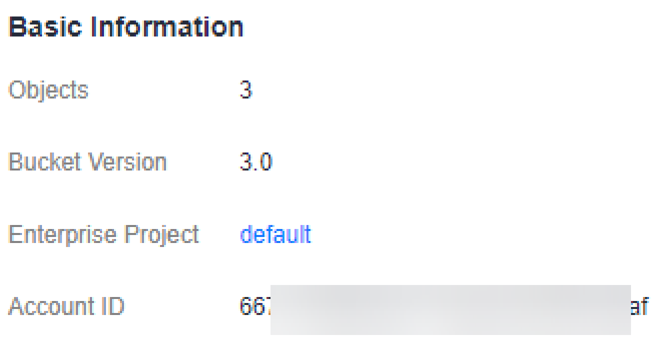


Domain Name Details

Type	Domain Name	Protocol	Operation
Endpoint ⓘ	[redacted]	HTTPS/HTTP	--
Access Domain Name ⓘ	[redacted]	HTTPS/HTTP	Bind User Domain Name
Static website hosting domain name	--	HTTPS/HTTP	Configure

Step 8 In the **Basic Information** area, view the bucket's basic information, including number of objects, bucket version, enterprise project, and account ID.

Figure 3-8 Bucket's basic information



Basic Information

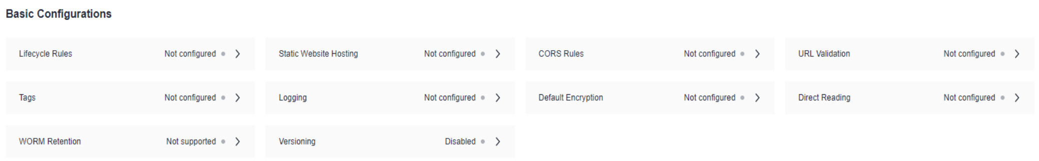
Objects	3
Bucket Version	3.0
Enterprise Project	default
Account ID	66[redacted]af

Table 3-5 Parameters in the Basic Information area

Parameter	Description
Objects	The total number of stored folders and objects of all versions in a bucket
Bucket Version	Version number of the bucket. **3.0** indicates the latest bucket version, and **--** indicates versions earlier than 3.0.
Enterprise Project	Enterprise project that a bucket belongs to
Account ID	Unique identity of the bucket owner. It is the same as **Account ID** on the **My Credentials** page.

Step 9 In the **Basic Configurations** area, view the bucket's basic configurations, including logging, server-side encryption, and direct reading. You can click a card to make required configurations.

Figure 3-9 Basic configurations of the bucket

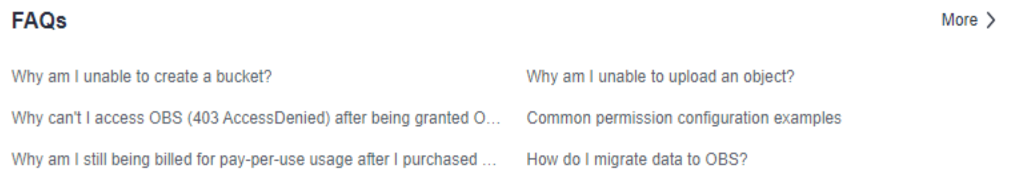


Basic Configurations

Lifecycle Rules	Not configured	Static Website Hosting	Not configured	CORS Rules	Not configured	URL Validation	Not configured
Tags	Not configured	Logging	Not configured	Default Encryption	Not configured	Direct Reading	Not configured
WORM Retention	Not supported	Versioning	Disabled				

Step 10 In the **FAQs** area, view bucket-related FAQs. You can click **More** in the upper right corner to view more FAQs.

Figure 3-10 Bucket FAQs



FAQs

- Why am I unable to create a bucket?
- Why can't I access OBS (403 AccessDenied) after being granted O...
- Why am I still being billed for pay-per-use usage after I purchased ...
- Why am I unable to upload an object?
- Common permission configuration examples
- How do I migrate data to OBS?

Step 11 In the **Learn More** area, choose to view best practices or usage guide.

Figure 3-11 Learn More

Learn More

Best Practices >
Quickly master skills and acquire knowledge.

Usage Guide >
Learn how to make better use of OBS.

----End

Using the API

Obtaining Bucket Metadata

Using SDKs

Java	Python	C: not supported	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

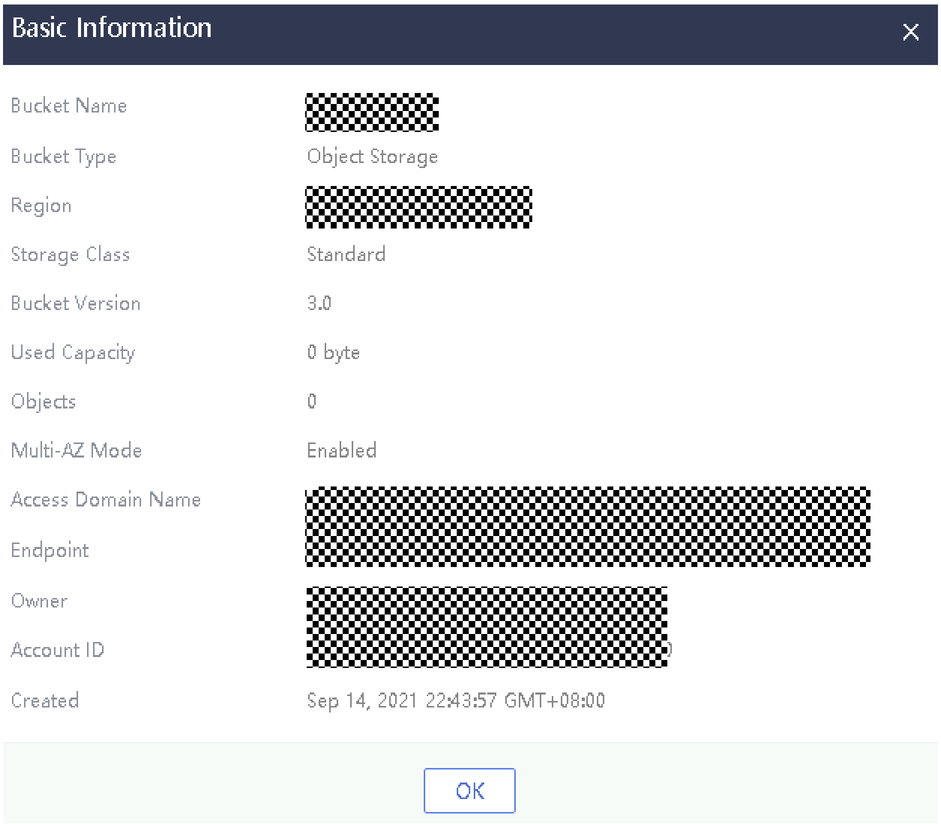
Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Right-click the bucket you want to view and choose **Basic Information** from the shortcut menu.

Step 3 In the displayed **Basic Information** window, view your required information, as shown in **Figure 3-12**.

Figure 3-12 Bucket's basic information



Basic Information

Bucket Name	[redacted]
Bucket Type	Object Storage
Region	[redacted]
Storage Class	Standard
Bucket Version	3.0
Used Capacity	0 byte
Objects	0
Multi-AZ Mode	Enabled
Access Domain Name	[redacted]
Endpoint	
Owner	[redacted]
Account ID	
Created	Sep 14, 2021 22:43:57 GMT+08:00

Table 3-6 Parameter description

Parameter	Description
Bucket Name	Name of the bucket.
Bucket Type	Object storage or a parallel file system.
Region	Region where the bucket is located.
Storage Class	Storage class of the bucket. It can be **Standard**, **Infrequent Access**, or **Archive**.
Bucket Version	Version number of the bucket.
Used Capacity	Total capacity used by objects in the bucket.
Objects	Number of objects in the bucket.
Multi-AZ Mode	Status of the multi-AZ function. If multi-AZ is enabled, data will be stored in multiple AZs.
Account ID	Unique identity of the bucket owner. It is the same as **Account ID** on the **My Credentials** page.
Created	Time when the bucket was created.

----End

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
```
obsutil stat obs://bucket [-acl] [-bf=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```
- In Linux or macOS
```
./obsutil stat obs://bucket [-acl] [-bf=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

Examples

- Take the Windows OS as an example. Run the **obsutil stat obs://bucket-test** command to query the basic properties of bucket **bucket-test**.
```
obsutil stat obs://bucket-test

Bucket:
  obs://bucket-test
StorageClass:
  standard

ObsVersion:
  3.0
BucketType:
  OBJECT
AvailableZone:
  multi-az
ObjectNumber:
  8005
Size:
  320076506
Quota:
  0
```

Parameter Description

Parameter	Optional or Mandatory	Description
s	Optional (additional parameter)	Displays simplified query result. **NOTE** In the simplified format, the returned result contains only the bucket name.
sc	Optional (additional parameter)	Queries the storage classes of the buckets when listing buckets.
j	Optional (additional parameter). It must be used together with **sc**.	Indicates the maximum number of concurrent tasks for querying the bucket storage class. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.

Parameter	Optional or Mandatory	Description
du	Optional (additional parameter)	Quickly returns the total size of listed objects, without displaying detailed object information. This parameter can be used together with other parameters. **NOTE** This parameter takes effect only on listing objects, but not on listing buckets.
fs	Optional (additional parameter)	Specifies whether the method of listing parallel file systems is applied. If you are listing parallel file systems, you are recommended to add this parameter. **CAUTION** ● With this method, the listing time required varies largely depending on the directory structures. ● After this parameter is enabled, **marker** and **limit** will be ignored. Then, the size of the bucket or parallel file system will be calculated. ● This parameter is only supported by obsutil 5.5.12 and later.
limit	Optional (additional parameter)	Maximum number of buckets that can be queried. If the value is less than 0, all buckets are listed. If it is left blank, a maximum of 1000 buckets can be listed by default.
format	Optional (additional parameter)	Prints a listing result in the user-defined format. Currently, the value can only be **default**, indicating that the listing result is displayed in one row.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.

Parameter	Optional or Mandatory	Description
t	Optional (additional parameter)	Specifies the user's security token.

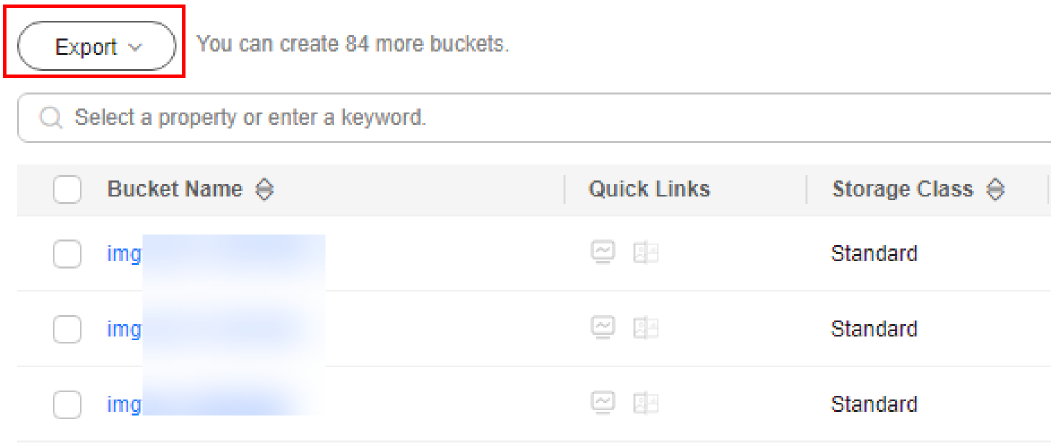
Related Operations

Exporting a Bucket List

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 Export all buckets. Specifically, click **Export** in the upper left corner of the bucket list.

Figure 3-13 Exporting all buckets



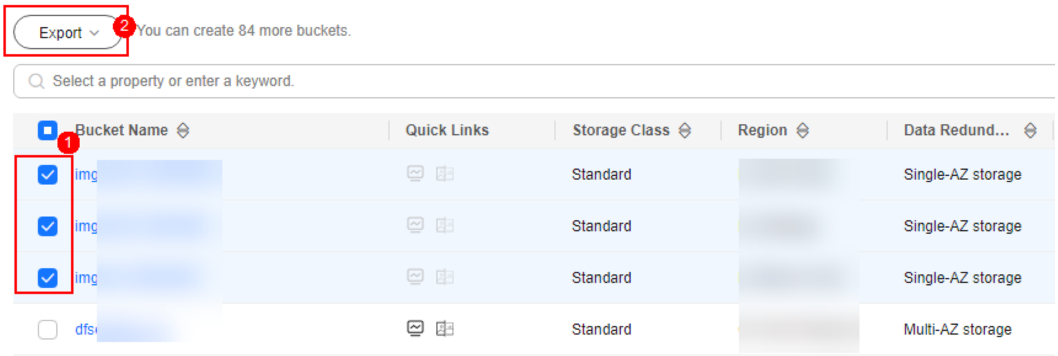
You can create 84 more buckets.

Select a property or enter a keyword.

☐	Bucket Name	Quick Links	Storage Class
☐	img[redacted]		Standard
☐	img[redacted]		Standard
☐	img[redacted]		Standard

Step 3 Export the selected buckets. Specifically, select the buckets to export and click **Export** in the upper left corner of the bucket list.

Figure 3-14 Exporting the selected buckets



You can create 84 more buckets.

Select a property or enter a keyword.

☐	Bucket Name	Quick Links	Storage Class	Region	Data Redund...
☑	img[redacted]		Standard	[redacted]	Single-AZ storage
☑	img[redacted]		Standard	[redacted]	Single-AZ storage
☑	img[redacted]		Standard	[redacted]	Single-AZ storage
☐	dfs[redacted]		Standard	[redacted]	Multi-AZ storage

Step 4 Obtain the bucket list in Excel, which is automatically downloaded to your local computer.

The file lists all the buckets of the current account and includes the following information: bucket name, storage class, region, data redundancy policy, used capacity, object quantity, bucket version, enterprise project, and bucket creation time.

----End

3.6 Managing Bucket Quotas

Scenarios

By default, neither the entire OBS system nor any single bucket has limits on storage capacity or the number of objects that can be stored. You can set a quota for a bucket to limit the total size of objects that can be uploaded to the bucket. If the total object size reaches the upper limit, object uploads will fail.

A bucket quota can control object uploading only after the quota is set. If the bucket quota is less than the capacity of the uploaded objects, the existing objects will not be deleted, but new objects cannot be uploaded. In this case, you can upload new objects only after deleting some existing objects until the used storage capacity is less than the quota limit.

NOTE

- A bucket quota must be a non-negative integer, in bytes. The maximum value is 2^{63} – 1.
- OBS does not provide an API for deleting bucket quotas. You can set the bucket quota to 0 to cancel the quota limit.

Ways to Manage Bucket Quotas

You can use APIs or SDKs to manage bucket quotas.

Using the API

Configuring Bucket Storage Quota

Using SDKs

Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

3.7 Viewing Bucket Usage Statistics

On OBS Console, you can view the storage, traffic, and requests of a single bucket.

NOTE

To view usage, you need to configure the **ces:metricData:list** policy in a regional project. For details, see **Cloud Eye Custom Policies**.

Scenarios

OBS provides the monitoring for the items described in **Table 3-7**.

Table 3-7 Metric description

Category	Item	Description
Storage	Usage (Total)	Measures the total storage capacity occupied by all objects and the object quantity in a bucket.
	Usage (By Class)	Measures the storage capacity occupied by objects in Standard, Infrequent Access, and Archive storage classes in a bucket. For more information about storage classes, see **Storage Classes**.
	Objects (By Class)	Measures the number of objects in Standard, Infrequent Access, and Archive storage classes in a bucket. For more information about storage classes, see **Storage Classes**.
Traffic	Traffic	Measures the inbound and outbound traffic.
Requests	TPS	Measures the total transactions per second (TPS) and average request latency.
	Requests	Measures the number of GET, PUT, and DELETE requests and total number of all requests.
	Percentage	Measures the percentage of successful, valid, and interrupted requests.
	Status Codes	Measures the distribution of status codes returned by the server. For more information about different status codes, see **Status Codes**.

Constraints

- You can query a period of up to 30 days.
- The monitoring data is not real time. There is approximately one hour delay.

Viewing Metrics

On the **Metrics** page, you can view the storage, traffic, and requests of a single bucket.

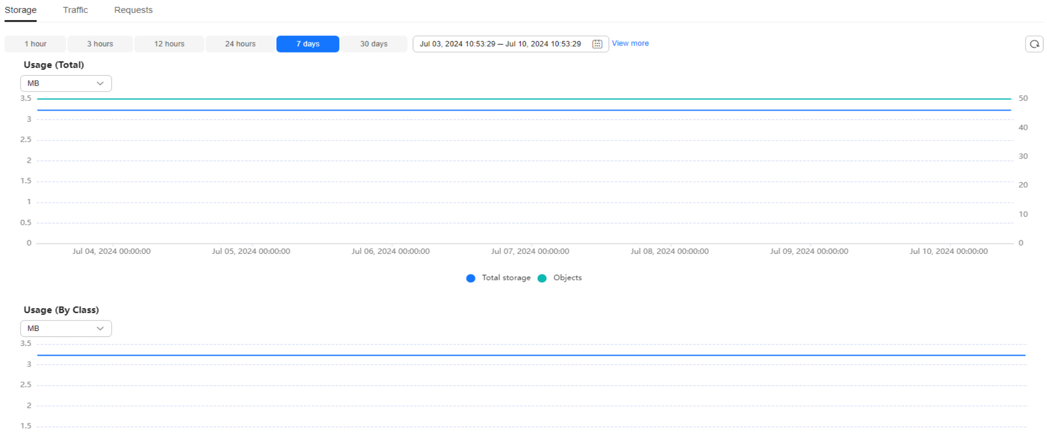
Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Metrics**.

Step 4 Select a metric type and a period to view related statistics,

Figure 3-15 Viewing metrics



In the chart, you can:

- Choose one or more legends to display what you want to view.
- Move your mouse pointer over the statistical line to view the statistics of each item at a specific point in time.

----End

3.8 Adding Tags to a Bucket

When creating a bucket, you can add tags to it. For details, see **Creating a Bucket**. You can also add tags to a bucket after it has been created. This section describes how to add tags to an existing bucket.

Scenarios

Tags help you to identify your cloud resources. When you have many cloud resources of the same type, you can use tags to classify them by dimension (for example, by purpose, owner, or environment) for usage or cost analysis.

In OBS, you can use tags to identify and classify OBS buckets.

If you add tags to a bucket, service detail records (SDRs) generated for it will be labeled with these tags. You can classify SDRs by tag for cost analysis. For example, if you have an application that uploads its running data to a bucket, you can tag the bucket with the application name. In this manner, the costs on the application can be analyzed using tags in SDRs.

Constraints

- A tag is structured as a key-value pair that is case sensitive.
- A bucket can have a maximum of 10 tags. Each tag has only one key and one value.

- Each key must be unique among all tags of a bucket, whereas the tag values can be repetitive or left blank.
- It takes approximately three minutes for an added tag to take effect.

Ways to Configure Bucket Tags

You can use OBS Console, APIs, or SDKs to configure tags for a bucket.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

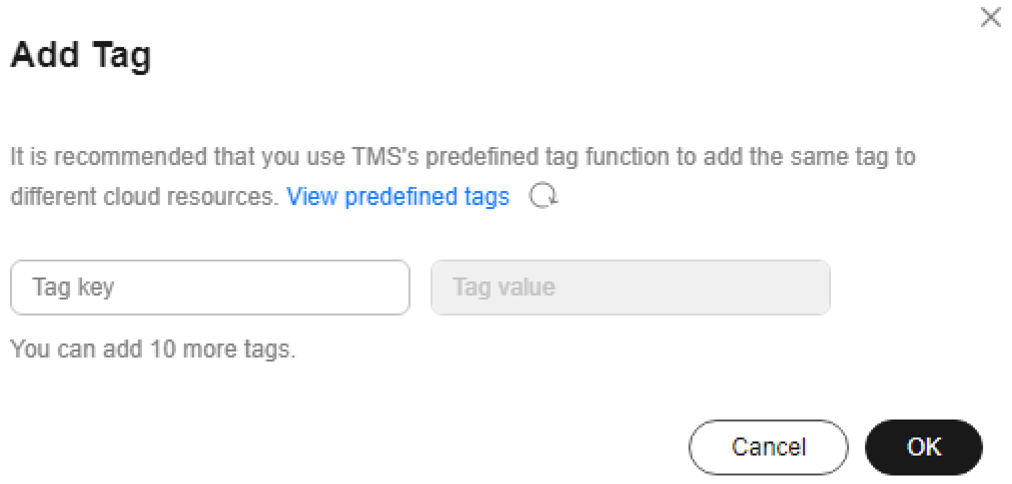
Step 3 In the navigation pane, choose **Overview**.

Step 4 In the **Basic Configurations** area, click **Tags**.

Alternatively, you can choose **Basic Configurations** > **Tagging** in the navigation pane.

Step 5 Click **Add Tag**. The **Add Tag** dialog box is displayed.

Figure 3-16 Add Tag



Add Tag

It is recommended that you use TMS's predefined tag function to add the same tag to different cloud resources. View predefined tags

You can add 10 more tags.

Step 6 Set the key and value based on **Table 3-8**.

Table 3-8 Parameter description

Parameter	Description
Tag key	Key of a tag. Tag keys for the same bucket must be unique. You can customize tags or select the ones predefined on TMS. A tag key: ● Must contain 1 to 36 characters and be case sensitive. ● Cannot start or end with a space or contain the following characters: =*<>\,|/

Parameter	Description
Tag value	Value of a tag. A tag value can be repetitive or left blank. A tag value: ● Can contain 0 to 43 characters and must be case sensitive. ● Cannot contain the following characters: =*<>\,|/

Step 7 Click **OK**.

It takes approximately 3 minutes for the tag to take effect.

----End

Using the API

Configuring Tags for a Bucket

Using SDKs

Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

Related Operations

In the tag list, click **Edit** to change the tag value or click **Delete** to remove the tag.

3.9 Deleting a Bucket

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to delete unwanted or empty buckets to free up the quota of buckets.

An empty bucket must meet the following requirements:

- There is no object or any historical version of an object in the bucket.
- There is not any incomplete multipart upload in the bucket. In other words, there are no fragments in the bucket.

NOTE

- If versioning is enabled for the bucket, ensure that all historical versions and versions with the **Delete Marker** (which are also considered as historical versions) have been deleted.
- The name of a deleted bucket can be reused at least 30 minutes after the deletion.

Prerequisites

- All objects in the bucket have been permanently deleted. A bucket must be emptied before it can be deleted.

NOTICE

Objects under the **Objects**, **Deleted Objects**, and **Fragments** tabs must be all deleted.

- A bucket can only be deleted by the bucket owner.

Ways to Delete a Bucket

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to delete a bucket.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

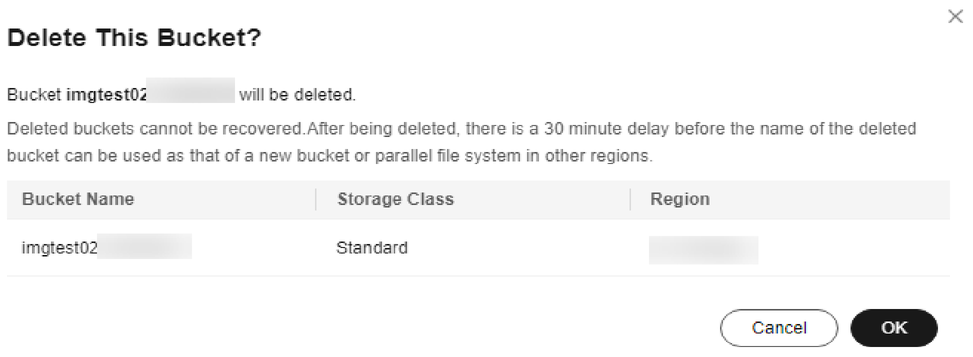
Step 2 In the bucket list, select the bucket you want to delete, and then click **Delete** on the right.

NOTE

The name of a deleted bucket can be reused for another bucket or parallel file system at least 30 minutes after the deletion.

Step 3 Click **OK** to confirm the deletion.

Figure 3-17 Deleting a bucket



Delete This Bucket?

Bucket **imgtest02**[redacted] will be deleted.

Deleted buckets cannot be recovered.After being deleted, there is a 30 minute delay before the name of the deleted bucket can be used as that of a new bucket or parallel file system in other regions.

Bucket Name	Storage Class	Region
imgtest02[redacted]	Standard	[redacted]

----End

Using the API

Deleting Buckets

Using SDKs

Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Select the bucket you want to delete and click **Delete**.

NOTE

The name of a deleted bucket can be reused for another bucket or a parallel file system at least 30 minutes after the deletion.

Step 3 Click **Yes** to delete the bucket.

----End

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
```
obsutil rm obs://bucket [-f] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```
- In Linux or macOS
```
./obsutil rm obs://bucket [-f] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

Examples

- Take the Windows OS as an example. Run the **obsutil rm obs://bucket-test** command to delete bucket **bucket-test**.
```
obsutil rm obs://bucket-test

Do you want to delete bucket [bucket-test] ? Please input (y/n) to confirm:
y
Delete bucket [bucket-test] successfully!
```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
f	Optional (additional parameter)	Runs in force mode.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.

Parameter	Optional or Mandatory	Description
t	Optional (additional parameter)	Specifies the user's security token.

4 Object Management

4.1 Object Overview

An object is the basic unit of data storage on OBS. It consists of object data and object metadata that describes object attributes. Data uploaded to OBS is stored as objects in **buckets**.

An object consists of data, metadata, and a key.

- A key specifies the name of an object. An object key is a UTF-8 string up to 1,024 characters long. Each object is uniquely identified by a key within a bucket.
- Metadata describes an object, and can be system-defined or user-defined. The metadata is a set of key-value pairs that are assigned to the object stored in OBS.
  - System-defined metadata is automatically assigned by OBS for processing objects. Such metadata includes Date, Content-Length, Last-Modified, Content-MD5, and more.
  - User-defined metadata is specified when you upload objects and is used to describe objects.
- Data refers to the content of the object.

Generally, objects are managed as files. However, OBS is an object-based storage service and there is no concept of files and folders. For easy data management, OBS provides a method to simulate folders. By adding a slash (/) in an object name, for example, **test/123.jpg**, you can simulate **test** as a folder and **123.jpg** as the name of a file under the **test** folder. However, the object key remains **test/123.jpg**.

When uploading an object, you can specify a **storage class** for it. If you do not specify a storage class, the object inherits the storage class of the bucket. You can also change the storage class of an existing object in a bucket.

You can **upload files (data) to a bucket** in the ways you like based on your habits and service scenarios. OBS then stores the files as objects in the bucket. In OBS, buckets are located in **different regions**. No matter what method you use, you can access the same bucket and its resources in the same region.

Guidelines on Naming Object Keys

Although any UTF-8 characters can be used in an object key name, naming object keys according to the following guidelines can help maximize the object keys' compatibility with other applications. Ways to analyze special characters vary with applications. The following guidelines help object key names substantially meet the requirements of DNS, web security characters, XML analyzers and other APIs.

The following character sets can be freely used in key names.

Alphanumeric characters (also known as unreserved characters)	[0-9a-zA-Z]
Special characters (also known as reserved characters)	Exclamation mark (!) Hyphen (-) Underscore (_) Period (.) Asterisk (*) Single quotation mark (') Left bracket "(" Right bracket ")"

The following are examples of valid object key names:

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

Percent-Encoding of Reserved Characters

If a reserved character has a special meaning (known as reserved purpose) in a URI and the character must be used for other purposes in the URI, this character must be percent-encoded. Use UTF-8 to encode non-ASCII characters. Otherwise, the names of the objects that are uploaded to OBS may be different from what is expected. For example, if reserved character "/" is used as the delimiter of path components in a URI, the character has a special meaning (separating a bucket name from an object name). If "/" is used in a component of the path in a URI, use three characters "%2F" or "%2f" to replace "/". The reserved character " " must be encoded as "%20". For example, the string "abc d" will be encoded as "abc%20d".

Characters That May Require Special Processing

Characters that require encoding in a key name

- Ampersand (&)
- Dollar sign ($)
- Semicolon (;)
- Colon (:)

- Plus sign (+): OBS decodes plus signs (+) in a request URI into spaces. If an original object key name contains plus signs (+), it must be encoded into %2B before being put into the request URI.
- Space: A large number of consecutive spaces may be lost in some cases.
- Equality sign (=)
- At sign (@)
- Comma (,)
- Question mark (?)
- ASCII characters: 00–1F in hexadecimal form (0–31 in decimal form) and 7F (127 in decimal form)

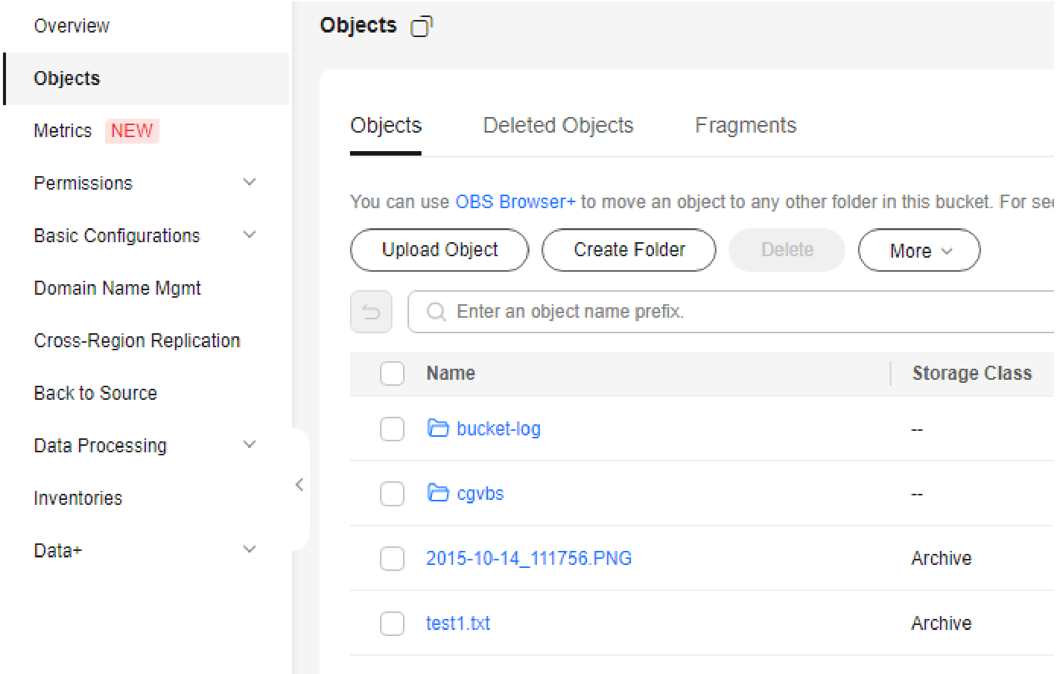
Avoid using the following characters in key names, because these characters require a lot of special processing to keep consistency across all applications.

- Backslash (\)
- Left brace ({)
- Non-printable ASCII characters (128–255 decimal characters)
- Insert symbol (^)
- Right brace (})
- Percentage character (%)
- Accent/Untick (`)
- Right square bracket (])
- Quotation mark
- Greater than sign (>)
- Left square bracket ([)
- Tilde (~)
- Less than sign (<)
- Number sign (#)
- Vertical bar (|)

Note that OBS adopts a flat structure, where you create buckets and store objects in buckets. There are no sub-buckets or sub-folders in the structure. However, you can use key name prefixes and delimiters to deduce the logical structure like OBS Console does. The folder concept is available on OBS Console. Assume that your bucket (**companybucket**) contains four objects with the following object keys:

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

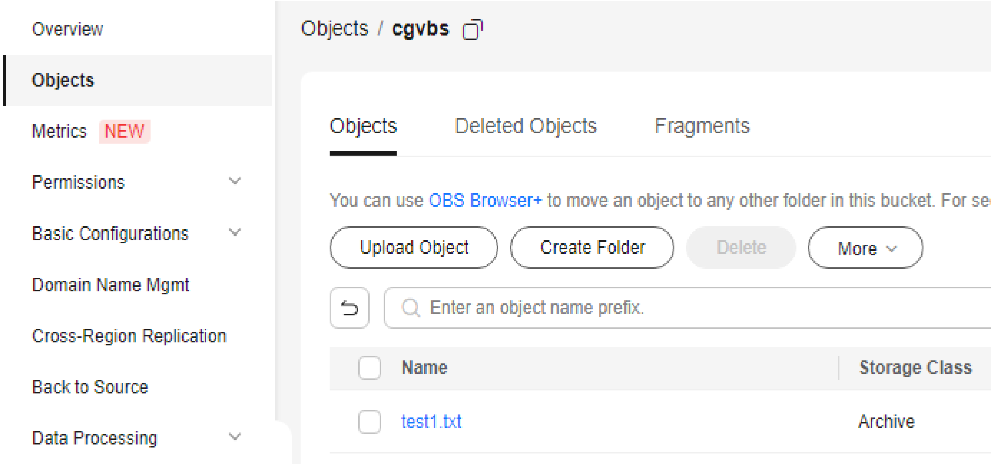
OBS Console uses a key name prefix (**bucket-log/** or **cgvbs/**) and separator (/) to display the folder structure, as shown in the following figure.



You can use OBS Browser+ to move an object to any other folder in this bucket. For se

Name	Storage Class
bucket-log	--
cgvbs	--
2015-10-14_111756.PNG	Archive
test1.txt	Archive

The **2015-10-14_111756.png** and **test1.txt** keys do not have a prefix, so they appear at the root level of the bucket. If you open the **cgvbs/** folder, you will see that it contains the **test1.txt** object.



You can use OBS Browser+ to move an object to any other folder in this bucket. For se

Name	Storage Class
test1.txt	Archive

Assume that your bucket (**companybucket**) contains two objects with the following object keys:

- obj
- 1/../obj or 1/./obj

If you use APIs, SDKs, or OBS Browser+ to obtain these two objects, you will successfully get them. If you use OBS Console to access them, you will actually get object **obj** based on the relative relationship. Since **../** and **./** has special semantics in URIs, avoid using them in object keys. In addition, if an object name contains **../** or **./**, this object cannot be edited on OBS Console, for example, transitioning storage classes or configuring lifecycle rules.

Object names cannot start with consecutive periods (..) or a single period (.).

4.2 Upload

4.2.1 Upload Overview

Scenarios

You can upload files or folders to an existing OBS bucket. If you want to classify files, you can create folders and upload files to different folders.

For details about how to create a folder, see **Creating a Folder**.

These files can be texts, images, videos, or any other type of files.

NOTICE

OBS allows you to upload objects to buckets in a specified region but is not aware of any content of your objects. If your object uploads involve cross-border transfer, ensure that your use complies with relevant laws and regulations.

Notes and Constraints

Size and number of objects to upload

- OBS Console puts limits on the size and number of files you can upload.
  - In regions that support batch uploads, a maximum of 100 files can be uploaded at a time, with a total size of no more than 5 GB. If you upload only one file in a batch, this file cannot exceed 5 GB.
  - In regions that do not support batch uploads, only one file can be uploaded at a time, with a size of no more than 50 MB.
- With OBS Browser+ or obsutil, you can upload files smaller than 48.8 TB. OBS Browser+ allows you to upload up to 500 files at a time. There is no limit on the number of files you can upload using obsutil at a time.
- With PUT, POST, or append methods of the OBS SDKs or API, you can upload files smaller than 5 GB.
- With multipart uploads of the OBS SDKs or API or resumable uploads of the OBS SDKs, you can upload files smaller than 48.8 TB.
- You can specify a quota for a bucket to limit its space usage. For details, see **Managing Bucket Quotas**.

Object naming

See **Guidelines on Naming Object Keys**.

Batch operations

Batch uploads are available only when:

1. The bucket is in any of the following regions: CN-Hong Kong, AP-Bangkok, or AP-Singapore.

2. The bucket version is 3.0. To view the bucket version, see **Checking OBS Version (OBS 2.0 or OBS 3.0)**.

Folder uploads

You can upload folders using OBS Console, OBS Browser+, or obsutil.

Others

- Versioning is not enabled for a bucket:
  - If you upload a new file with the same name as the one in the bucket, the new file will automatically overwrite the previous one and the ACL information of the previous file will not be retained.
  - If you upload a new folder with the same name as the one in the bucket, the two folders will be merged, and files in the new folder will overwrite those with the same name in the previous folder.
- If versioning is enabled for your bucket and you upload a new file with the same name as the one you previously uploaded to the bucket, a new file version will be added to the bucket. For details, see **Versioning**.

Object Types

Object Type	Description	Notes
Normal	Objects that are uploaded using **Streaming Upload (PUT)** or **Browser-based Upload (Post)**	If versioning is not enabled or is suspended, a newly uploaded object overwrites the previous one with the same name and only the object uploaded by the last API call is retained. If versioning is enabled, each operation on an object will create a new version of the object. The version generated by the most recent operation is the latest one.
Multipart	Objects that are uploaded using **Multipart Upload**	If versioning is not enabled or is suspended, the complete object that is assembled later overwrites the previous one with the same name and only the complete object assembled by the last API call is retained. If versioning is enabled, each operation on an assembled object will create a new version of the object. The version generated by the most recent operation is the latest one.
Appendable	Objects that are uploaded using **Appendable Upload**	You can directly append data to such objects.

Upload Methods

Upload Method	Description

Streaming upload (PUT)	Use PUT or POST if the size of the file to be uploaded is less than 5 GB.
	If you use PUT, you need to specify the object name in the URL, but object name is not required with POST, which uses the bucket domain name as the URL. Request lines of these two methods are given as follows:
	`PUT /ObjectName HTTP/1.1`
	`POST / HTTP/1.1`

Browser-based upload (POST)	
Multipart upload	Use this method when the size of the file to be uploaded is greater than 5 GB and less than 48.8 TB. For details, see **Multipart Upload**. **NOTE** If you have over 48.8 TB data to upload, refer to **Migrating Local Data to OBS**.
Resumable upload	Uploading large files often fails due to poor network conditions or program breakdowns. Re-uploads not only waste lost of resources, but may also fail again. To avoid such issues, you can use the resumable upload API, which works by dividing a to-be-uploaded file into multiple parts and then uploading them separately. This API helps save resources and improves efficiency by allowing for resumption from the break point and concurrent uploads. For details, see **Does OBS Support Resumable Data Transfer?**
Incremental synchronization	This function synchronizes incremental content in a local source directory to the specified target bucket in OBS to ensure consistency. It works by comparing the source files in a local directory with their counterparts in the corresponding bucket in OBS and uploading only those with content changes, so all source files can remain identical to those uploaded to OBS. For details, see **Synchronously Uploading Incremental Objects**.
Append upload	This method adds data to the end of an object in a specified bucket. If there is no object with the same key found in the bucket, a new object is created. For details, see **Appendable Upload**.
Temporary upload authorization	If a third-party user does not have the upload permission, they can be granted temporary access to upload objects within a validity period. For details, see **Authorizing a Third Party to Upload Objects**.

Ways to Upload

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to upload an object.

4.2.2 Streaming Upload (PUT)

Scenarios

If the size of an object, such as a text file, image, or video, to be uploaded is smaller than 5 GB, you can use streaming upload (PUT).

Parameters sent by PUT are passed in the request header. You need to specify the object name in the URL. An example request is as follows:

```
PUT /ObjectName HTTP/1.1
```

NOTICE

OBS allows you to upload objects to buckets in a specified region, but Huawei Cloud does not detect the object content you uploaded. If your object uploads involve cross-border transfer, ensure that your use complies with relevant laws and regulations.

Prerequisites

- You have the write permissions to a bucket.
- If you want to classify files, you can create folders and upload files to different folders. For details, see **Creating a Folder**.

Constraints

Size and number of objects to upload

- If a file is smaller than 5 GB, use streaming upload. To upload a file larger than 5 GB, refer to **Multipart Upload**.
- You can specify a quota for a bucket to limit its storage capacity. For details, see **Managing Bucket Quotas**.

Object naming

See **Guidelines on Naming Object Keys**.

Batch operations

Batch uploads are available only when:

1. The bucket is in CN-Hong Kong, AP-Bangkok, or AP-Singapore.
2. The bucket version is 3.0. To view a bucket version, see **Checking OBS Version (OBS 2.0 or OBS 3.0)**.

Folder uploads

You can upload folders using OBS Console, OBS Browser+, or obsutil.

Other

- Versioning is not enabled for a bucket:
  - If you upload a new file with the same name as the one in the bucket, the new file will automatically overwrite the previous one and the ACL information of the previous file will not be retained.
  - If you upload a new folder with the same name as the one in the bucket, the two folders will be merged, and files in the new folder will overwrite those with the same name in the previous folder.
- Versioning is enabled for a bucket: If you upload a new file with the same name as the one you previously uploaded to the bucket, a new file version will be added to the bucket. For details, see **Versioning**.

Ways to Upload

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to upload objects.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate.

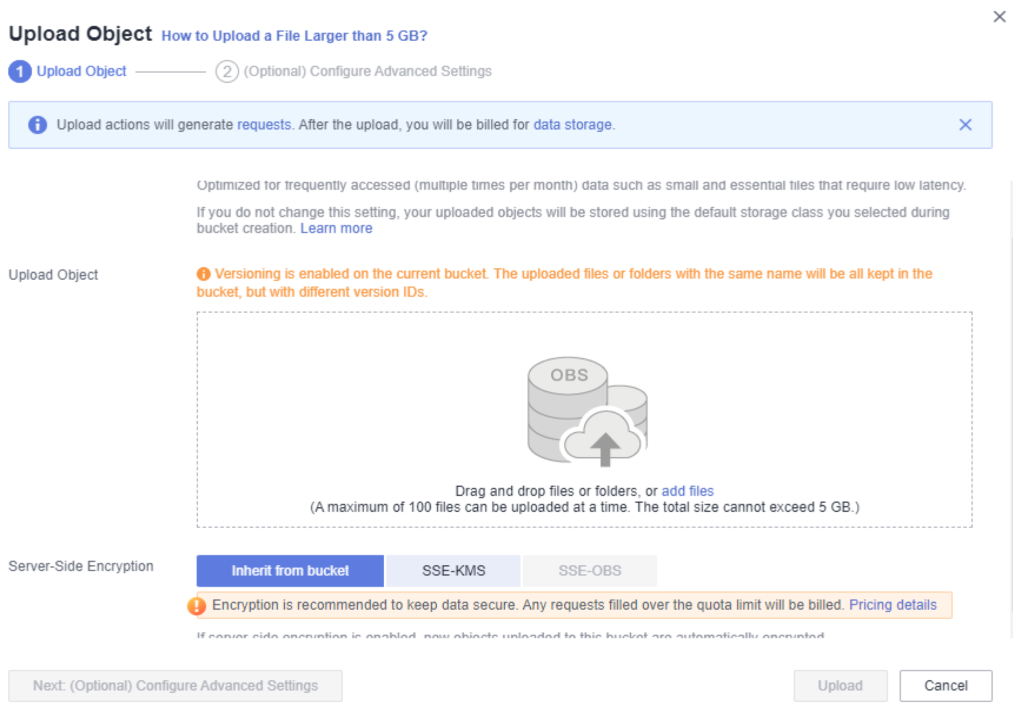
Step 3 Go to the folder that you want to upload files to and click **Upload Object**. The **Upload Object** dialog box is displayed.

Batch upload is used as an example here. If a region only supports single upload, follow the on-screen instructions.

NOTE

If the files that you want to upload to OBS are stored in Microsoft OneDrive, it is recommended that the names of these files contain a maximum of 32 characters to ensure compatibility.

Figure 4-1 Object Upload



Upload Object How to Upload a File Larger than 5 GB?

1 Upload Object —— 2 (Optional) Configure Advanced Settings

Upload actions will generate requests. After the upload, you will be billed for data storage.

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

If you do not change this setting, your uploaded objects will be stored using the default storage class you selected during bucket creation. Learn more

Upload Object

Versioning is enabled on the current bucket. The uploaded files or folders with the same name will be all kept in the bucket, but with different version IDs.

Drag and drop files or folders, or add files
(A maximum of 100 files can be uploaded at a time. The total size cannot exceed 5 GB.)

Server-Side Encryption: Inherit from bucket | SSE-KMS | SSE-OBS

Encryption is recommended to keep data secure. Any requests filled over the quota limit will be billed. Pricing details

If server-side encryption is enabled, new objects uploaded to this bucket are automatically encrypted

Next: (Optional) Configure Advanced Settings | Upload | Cancel

Step 4 Select a storage class. If you do not specify a storage class, the objects you upload inherit the default storage class of the bucket.

NOTE

An object can have a different storage class from its bucket. You can specify a storage class for an object when uploading it, or you can change the object storage class after the object is uploaded.

Step 5 Drag and drop the files or folders you want to upload to the **Upload Object** area.

You can also click **add files** in the **Upload Object** area to select files.

Step 6 **Server-Side Encryption**: Choose **Disable**, **SSE-KMS**, or **SSE-OBS**. For details, see **Server-Side Encryption**.

NOTE

If the bucket has Server-Side Encryption configured, an object you upload will inherit encryption from the bucket by default, but you can change the encryption option as required.

Step 7 (Optional) To configure WORM retention policies, or metadata, click **Next: (Optional) Configure Advanced Settings**.

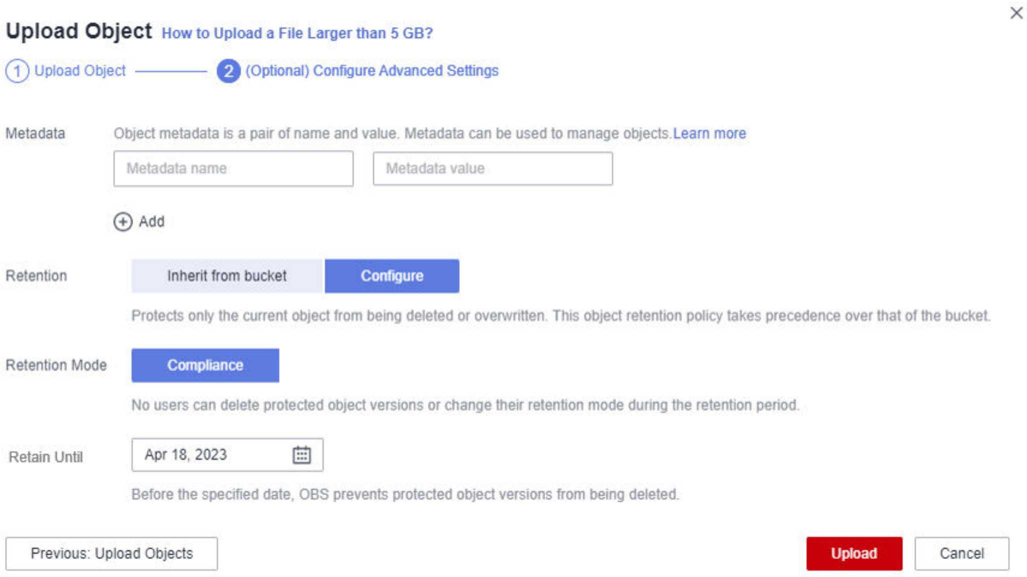
NOTE

WORM retention policies can be configured in the advanced settings only when WORM is enabled for the bucket.

Configuring metadata: Add metadata ContentDisposition, ContentLanguage, WebsiteRedirectLocation, ContentEncoding, or ContentType as needed. For more information, see **OBS Metadata**. Metadata is a set of name-value pairs. The metadata value cannot be left blank. You can add two or more metadata entries by clicking **Add**.

Configuring WORM retention: Choose **Inherit from bucket**, or choose **Configure** and then specify a retention period, to automatically protect new objects uploaded to the bucket from being deleted.

Figure 4-2 Configuring metadata or WORM retention



Upload Object How to Upload a File Larger than 5 GB?

1 Upload Object — 2 (Optional) Configure Advanced Settings

Metadata: Object metadata is a pair of name and value. Metadata can be used to manage objects. Learn more

Metadata name | Metadata value

Add

Retention: Inherit from bucket | Configure

Protects only the current object from being deleted or overwritten. This object retention policy takes precedence over that of the bucket.

Retention Mode: Compliance

No users can delete protected object versions or change their retention mode during the retention period.

Retain Until: Apr 18, 2023

Before the specified date, OBS prevents protected object versions from being deleted.

Previous: Upload Objects | Upload | Cancel

Step 8 Click **Upload**.

----End

Using APIs

Uploading an Object

Using SDKs

Java	**Python**	**C**	**Go**	BrowserJS: not supported	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**

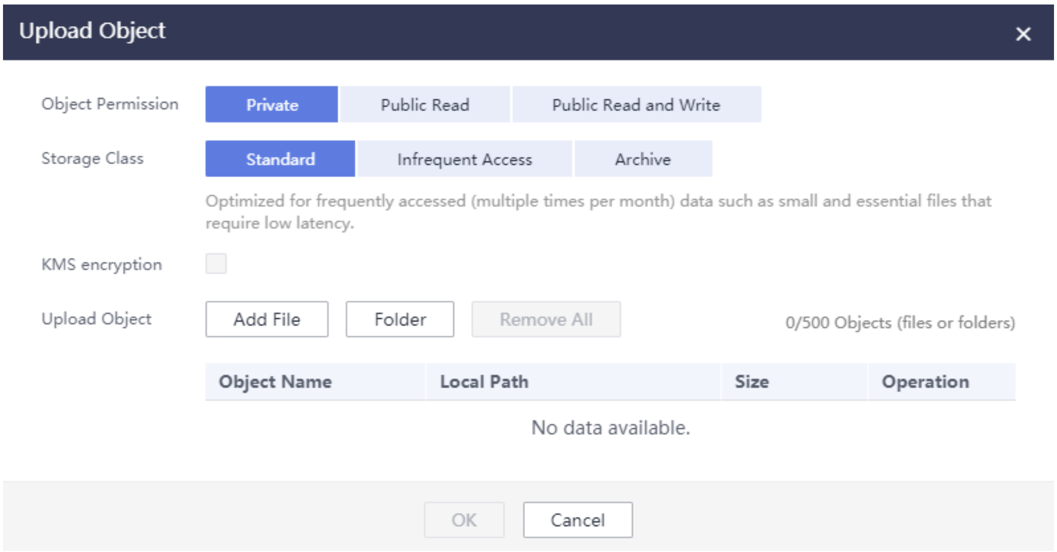
Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Click the bucket where you want to upload files or folders.

Step 3 Click **Upload** and then **Add File** or **Folder**, as shown in **Figure 4-3**.

Figure 4-3 Uploading a file or folder



Upload Object

Object Permission: Private | Public Read | Public Read and Write

Storage Class: Standard | Infrequent Access | Archive

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

KMS encryption □

Upload Object: Add File | Folder | Remove All — 0/500 Objects (files or folders)

Object Name	Local Path	Size	Operation

No data available.

For better experience when using the **Add File** function, you are advised to upload a maximum of 100 files at a time. If you need to upload more, place all the files in a folder and upload them by adding a folder.

NOTE

1. If message "Service Unavailable" is displayed when files are being uploaded, try again later.
2. If an access deny message is displayed when you are uploading a file or folder, possible causes are as follows:

- Access to the bucket is restricted by an ACL. For example, you do not have the write permission for the bucket.
- Access to the bucket is restricted by a bucket policy. For example, you do not have the write permission for the bucket, or write operations cannot be performed on the bucket during the current period.

  If such message is displayed, check ACL and policy settings of the bucket and resolve the problem accordingly.

⚠ CAUTION

You must have access to the file you want to upload, or the file upload will fail.

Step 4 In the displayed dialog box, select the file or folder you want to upload and click **Open**.

You can upload one folder or multiple files at a time. To upload multiple files, hold down **Ctrl** or **Shift** to select multiple files and batch upload them. You can also press **Ctrl+A** to select all files. The operations are consistent with those in Windows operating systems.

Step 5 Click **OK**.

----End

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
  - Uploading a file
    ```
    obsutil cp file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-u] [-vlength] [-vmd5] [-p=1] [-threshold=5248800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading a folder
    ```
    obsutil cp folder_url obs://bucket[/key] -r [-arcDir=xxx] [-dryRun] [-link] [-f] [-flat] [-u] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading multiple files/folders
    ```
    obsutil cp file1_url,folder1_url|filelist_url obs://bucket[/prefix] -msm=1 [-r] [-arcDir=xxx] [-dryRun] [-link] [-f] [-u] [-vlength] [-vmd5] [-flat] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx][-timeRange=time1-time2] [-at] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```

    NOTE

    In this command, **/prefix** is the name prefix for uploading folders. For the execution examples, see **Upload**.
- In Linux or macOS
  - Uploading a file
    ```
    ./obsutil cp file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-u] [-vlength] [-vmd5] [-p=1] [-threshold=5248800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading a folder
    ```
    ./obsutil cp folder_url obs://bucket[/key] -r [-arcDir=xxx] [-dryRun] [-link] [-f] [-flat] [-u] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-at] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading multiple files/folders
    ```
    ./obsutil cp file1_url,folder1_url|filelist_url obs://bucket[/prefix] -msm=1 [-r] [-arcDir=xxx] [-dryRun] [-link] [-f] [-u] [-vlength] [-vmd5] [-flat] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx][-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```

    NOTE

    In this command, **/prefix** is the name prefix for uploading folders. For the execution examples, see **Upload**.

Examples

- Take the Windows OS as an example. Run the **obsutil cp d:\temp\test.txt obs://bucket-test/key** command to upload the **test.txt** file in the **temp** directory in the **D:** drive to bucket **bucket-test** and rename the file as **key**.
  ```
  obsutil cp d:\temp\test.txt obs://bucket-test/key

  Parallel:      5                   Jobs:          5
  Threshold:     50.00MB             PartSize:      auto
  VerifyLength:  false               VerifyMd5:     false
  ```

```
CheckpointDir: C:\Users\Administrator\.obsutil_checkpoint

[==================================================] 100.00% 1.68 MB/s 8.46MB/
8.46MB 5s
Upload successfully, 8.46MB, n/a, d:\temp\test.txt --> obs://bucket-test/key, cost [5], status [200],
request id [0000016979E1D2B2860BB5181229C72C]
```

- Take the Windows OS as an example. Run the **obsutil cp d:\temp obs://bucket-test -f -r** command to recursively upload all files and subfolders in the **temp** directory in the **D:** drive to the **temp** folder in bucket **bucket-test**.

```
obsutil cp d:\temp obs://bucket-test -f -r

Parallel:      5                   Jobs:          5
Threshold:     50.00MB             PartSize:      auto
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: C:\Users\Administrator\.obsutil_checkpoint

Task id: 104786c8-27c2-48fc-bc6a-5886596fb0ed
OutputDir: C:\Users\Administrator\.obsutil_output

[======================================================] 100.00% tps:35.71 2.02 KB/s
7.20MB/7.20MB 0s
Succeed count is:   5         Failed count is:    0
Metrics [max cost:90 ms, min cost:45 ms, average cost:63.80 ms, average tps:35.71]

Task id: 104786c8-27c2-48fc-bc6a-5886596fb0ed
```

- For more examples, see **Upload**.

Parameter Description

Parameter	Optional or Mandatory	Description
file_url	Optional for uploading multiple files/ folders Mandatory for uploading a file	Local file path **NOTE** ● Do not nest paths when uploading multiple files/folders. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time. ● If this parameter is configured when uploading multiple files/folders, **msm** must be set to **1**. In this case, use commas (,) to separate multiple file paths, for example, **file_url1,file_url2**. ● Files and folders can both be included when uploading multiple files/folders. For example, **file_url1,folder_url1,file_url2,folder_url2**.

Parameter	Optional or Mandatory	Description
folder_url	Optional for uploading multiple files/ folders Mandatory for uploading a folder	Local folder path **NOTE** ● If **flat** is not configured when uploading a folder, the entire folder is uploaded. If **flat** is configured, all files in the folder are uploaded. ● Do not nest paths when uploading multiple files/folders. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time. ● If this parameter is configured when uploading multiple files/folders, **msm** must be set to **1**. In this case, use commas (,) to separate multiple folder paths, for example, **folder_url1,folder_url2**. ● Files and folders can both be included when uploading multiple files/folders. For example, **file_url1,folder_url1,file_url2,folder_url2**.
filelist_url	Optional for uploading multiple files/ folders	Indicates the path of the file that contains the list of files/folders to be uploaded. If this parameter is configured, **msm** must be set to **2**. **NOTE** ● The list file is in common text file formats, such as TXT and CSV. Each line in the file indicates a file or folder to be uploaded. For example: file_url1 file_url2 folder_url1 folder_url2 ● Do not nest paths in the list file. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time.
bucket	Mandatory	Bucket name

Parameter	Optional or Mandatory	Description
key	Optional	Indicates the object name or object name prefix specified when uploading a file, or the object name prefix specified when uploading a folder. The rules are as follows: ● If this parameter is left blank when uploading a file, the file is uploaded to the root directory of the bucket and the object name is the file name. If the value ends with a slash (/), the value is used as the object name prefix when the file is uploaded, and the object name is the value plus the file name. If the value does not end with a slash (/), the file is uploaded with the value as the object name. ● If this parameter is left blank when uploading a folder, the folder is uploaded to the root directory of the bucket. If the value ends with a slash (/), the value is used as the object name prefix of the folder to be uploaded. If the value does not end with a slash (/), the folder to be uploaded is prefixed with the value plus a slash (/). **NOTE** For details about how to use this parameter, see **Upload**.
fr	Optional for uploading a file (additional parameter)	Generates an operation result list when uploading a file.
flat	Optional for uploading a folder or multiple files/ folders (additional parameter)	Uploads all files in a folder but not the folder itself.
arcDir	Optional (additional parameter)	Path to which the uploaded files are archived
dryRun	Optional (additional parameter)	Conducts a dry run.

Parameter	Optional or Mandatory	Description
link	Optional (additional parameter)	Uploads the actual path of the symbolic-link file/folder **NOTICE** - If this parameter is not specified and the file to be uploaded is a symbolic-link file whose target file does not exist, the exception message "The system cannot find the file specified" will be displayed in Windows OS, while the exception message "No such file or directory" will be displayed in macOS or Linux OS. - Avoid the symbolic link loop of a folder, otherwise, the upload will exit due to panic. If you do not want the system to panic, set **panicForSymbolicLinkCircle** to **false** in the configuration file.
u	Optional (additional parameter)	Indicates incremental upload. If this parameter is set, each file can be uploaded only when it does not exist in the bucket, its size is different from the namesake one in the bucket, or it has the latest modification time. **CAUTION** When you compare each local file with data in the bucket, a billable HEAD request is generated. For details, see **Requests**.
vlength	Optional (additional parameter)	After the upload completes, check whether the sizes of the objects in the bucket are the same as those of the local files.
vmd5	Optional (additional parameter)	After the upload completes, check whether the MD5 values of the objects in the bucket are the same as those of the local files. **NOTE** - If the size of the file or folder to be uploaded is too large, using this parameter will degrade the overall performance due to MD5 calculation. - After the MD5 verification is successful, this parameter value is used for metadata **x-obs-meta-md5chksum**, for later MD5 verification during download or copy.
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart upload tasks when uploading a file. The default value is the value of **defaultParallels** in the configuration file.

Parameter	Optional or Mandatory	Description
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart upload, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the file or folder to be uploaded is smaller than the threshold, upload it directly. Otherwise, a multipart upload is required. ● If you upload a file or folder directly, no part record is generated, and resumable transmission is not supported. ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes).
acl	Optional (additional parameter)	Access control policies that can be specified when uploading files. Possible values are: ● private ● public-read ● public-read-write ● bucket-owner-full-control **NOTE** The preceding four values indicate private read and write, public read, public read and write, and bucket owner full control.
sc	Optional (additional parameter)	Indicates the storage classes of objects that can be specified when uploading files. Possible values are: ● **standard**: Standard storage class. It features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month) or data that is smaller than 1 MB. ● **warm**: Infrequent Access storage class. It is ideal for storing infrequently accessed (less than 12 times a year) data, but when needed, the access has to be fast. ● **cold**: Archive storage class. It provides secure, durable, and inexpensive storage for rarely-accessed (once a year) data. ● **deep-archive**: Deep Archive storage class (under limited beta testing). It is suitable for storing data that is barely (once every few years) accessed. This storage class costs less than the Archive storage class, but takes longer time (usually several hours) to restore data.

Parameter	Optional or Mandatory	Description
meta	Optional (additional parameter)	Indicates the standard and custom metadata that can be specified during file upload. This parameter should be configured in the following format: *key1*:*value1*#*key2*:*value2*#*key3*:*value3*. **NOTE** 1. The format example above indicates that the destination objects contain three groups of custom metadata: *key1:value1*, *key2:value2*, and *key3:value3*. 2. Standard metadata headers include **Content-Type**, **Content-Encoding**, **Cache-Control**, **Content-Disposition**, **Content-Language** and **Expires**.
ps	Optional (additional parameter)	Indicates the size of each part in a multipart upload task, in bytes. The value ranges from **100KB** to **5GB**. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes). ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source file size.
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart upload and saved to the **upload** subfolder. After the upload succeeds, its part record is deleted automatically. If the upload fails or is suspended, the system attempts to resume the task according to its part record when you perform the upload the next time.
r	Mandatory for uploading a folder (additional parameter) Optional for uploading multiple files/ folders	Indicates files and subfolders within the folder when uploading a folder recursively.

Parameter	Optional or Mandatory	Description
f	Optional for uploading a folder or multiple files/ folders (additional parameter)	Runs in force mode.
j	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the maximum number of concurrent tasks for uploading a folder. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
msm	Mandatory for uploading multiple files/ folders (additional parameter)	Enables the mode for uploading multiple files/folders. Possible values are **1** and **2**. **NOTE** ● If **msm** is set to **1**, the source URL indicates a list of file/folder names separated by commas. ● If **msm** is set to **2**, the source URL indicates a file containing a list of file/folder names. ● If the file or folder name already contains commas (,), do not set **msm** to **1**. ● If parameter **r** is not set, the folders in the list will not be uploaded.

Parameter	Optional or Mandatory	Description
exclude	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the file matching patterns that are excluded, for example: ***.txt**. **NOTE** - The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. - You can use ***** to represent ***** and **\?** to represent **?**. - If the name of the file to be uploaded matches the value of this parameter, the file is skipped. **NOTICE** - You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. - The matching pattern applies to the absolute file path (including the file name and file directory). - The matching pattern takes effect only for files in the folder. - Multiple **exclude** parameters can be specified, for example, **-exclude=*.xxx -exclude=*.xxx**.

Parameter	Optional or Mandatory	Description
include	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the file matching patterns that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use **\*** to represent ***** and **\?** to represent **?**. ● Only after identifying that the name of the file to be uploaded does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is uploaded. If not, the file is skipped. **NOTE** Example of uploading files in a request with the **include** parameter contained: `./obsutil cp /localpath/ obs://test/ -include=/localpath/2022-12-09/* -f -r` This command uploads files that are under **localpath** and start with **/localpath/2022-12-09/** to bucket **test**. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute file path (including the file name and file directory). ● The matching pattern takes effect only for files in the folder. ● Multiple **include** parameters can be specified, for example, **-include=*.xxx -include=*.xxx**.
at	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates that only the files whose latest access time is within the value of **timeRange** are uploaded. **NOTE** ● This parameter must be used together with **timeRange**.
disableDir Object	Optional for uploading multiple folders (additional parameter)	Indicates the folders themselves are not uploaded as an object. Configuring this parameter can avoid uploading empty folders to a bucket. If a folder contains files, the files will be uploaded and the original path format is retained.

Parameter	Optional or Mandatory	Description
timeRange	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the time range matching pattern when uploading files. Only files whose latest modification time is within the configured time range are uploaded. This pattern has a lower priority than the file matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured file matching patterns. **NOTE** ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-***time2*, all files whose latest modification time is earlier than *time2* are matched. If it is set to *time1***-***, all files whose latest modification time is later than *time1* are matched. **NOTICE** Time in the matching pattern is the UTC time.
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on folders.

Parameter	Optional or Mandatory	Description
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **cp_{succeed | failed | warning}_report_***time***_TaskId.txt** ● By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file. ● If there are multiple folders and files and you need to confirm the detailed error information about a failed task, refer to the failure list **cp_failed_report_***time***_TaskId.txt** in the result list folder and the **log files** in the log path.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

Response

Field	Description
Parallel	Parameter **-p** in the request

Field	Description
Jobs	Parameter **-j** in the request
Threshold	Parameter **-threshold** in the request
PartSize	Parameter **-ps** in the request
Exclude	Parameter **-exclude** in the request
Include	Parameter **-include** in the request
TimeRange	Parameter **-timeRange** in the request
VerifyLength	Parameter **-vlength** in the request
VerifyMd5	Parameter **-vmd5** in the request
CheckpointDir	Parameter **-cpd** in the request
OutputDir	Parameter **-o** in the request
ArcDir	Parameter **-arcDir** in the request
Succeed count	Number of successful tasks
Failed count	Number of failed tasks
Skip count	Number of tasks that are skipped during incremental upload, download, or copy, and synchronous upload, download, or copy. **NOTE** Skipped tasks are recorded into successful tasks.
Warning count	Number of tasks that are executed successfully but contain warnings. **NOTE** ● The task for which a warning is generated may be a failure or a success, which needs to be further determined according to the corresponding result list. ● The number of tasks that generate warnings is independent of the number of successful or failed tasks. The total number of tasks is the number of successful tasks plus the number of failed tasks.
Succeed bytes	Number of bytes that are successfully uploaded or downloaded.
max cost	Maximum duration of all tasks, in ms
min cost	Minimum duration of all tasks, in ms
average cost	Average duration of all tasks, in ms
average tps	The average number of tasks completed per second
Task id	Unique ID of an operation, which is used to search for the result list generated in a batch task

4.2.3 Browser-based Upload (Post)

Scenarios

If the size of an object, such as a text file, image, or video, to be uploaded is smaller than 5 GB, you can use browser-based upload (POST).

Data sent by POST is passed through the form field in the message body. You need to specify bucket domain name in the URL. An example request is as follows:

```
POST / HTTP/1.1
```

NOTICE

OBS allows you to upload objects to buckets in a specified region, but Huawei Cloud does not detect the object content you uploaded. If your object uploads involve cross-border transfer, ensure that your use complies with relevant laws and regulations.

Prerequisites

- You have the write permissions to a bucket.
- If you want to classify files, you can create folders and upload files to different folders. For details, see **Creating a Folder**.

Constraints

Size and number of objects to upload

- If a file is smaller than 5 GB, use streaming upload. To upload a file larger than 5 GB, refer to **Multipart Upload**.
- You can specify a quota for a bucket to limit its storage capacity. For details, see **Managing Bucket Quotas**.

Object naming

See **Guidelines on Naming Object Keys**.

Batch operations

Batch uploads are available only when:

1. The bucket is in CN-Hong Kong, AP-Bangkok, or AP-Singapore.
2. The bucket version is 3.0. To view a bucket version, see **Checking OBS Version (OBS 2.0 or OBS 3.0)**.

Folder uploads

You can upload folders using OBS Console, OBS Browser+, or obsutil.

Other

- Versioning is not enabled for a bucket:

  - If you upload a new file with the same name as the one in the bucket, the new file will automatically overwrite the previous one and the ACL information of the previous file will not be retained.
  - If you upload a new folder with the same name as the one in the bucket, the two folders will be merged, and files in the new folder will overwrite those with the same name in the previous folder.
- Versioning is enabled for a bucket: If you upload a new file with the same name as the one you previously uploaded to the bucket, a new file version will be added to the bucket. For details, see **Versioning**.

Ways to Upload

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to upload objects.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate.

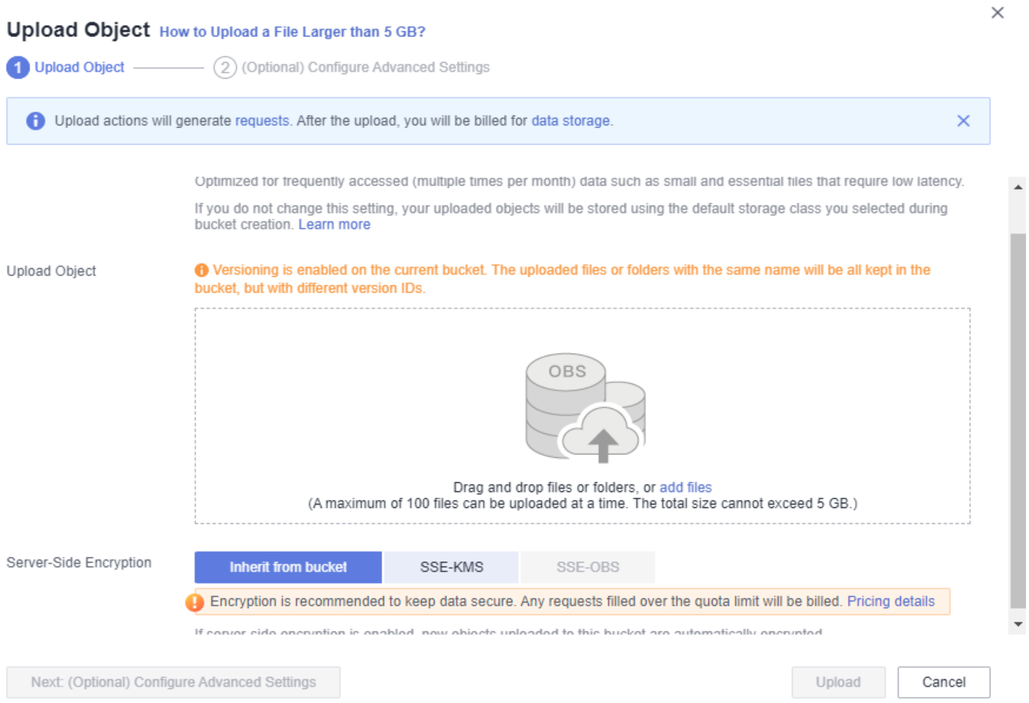
Step 3 Go to the folder that you want to upload files to and click **Upload Object**. The **Upload Object** dialog box is displayed.

Batch upload is used as an example here. If a region only supports single upload, follow the on-screen instructions.

NOTE

If the files that you want to upload to OBS are stored in Microsoft OneDrive, it is recommended that the names of these files contain a maximum of 32 characters to ensure compatibility.

Figure 4-4 Object Upload



Upload Object How to Upload a File Larger than 5 GB?

1 Upload Object —— 2 (Optional) Configure Advanced Settings

Upload actions will generate requests. After the upload, you will be billed for data storage.

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

If you do not change this setting, your uploaded objects will be stored using the default storage class you selected during bucket creation. Learn more

Upload Object

Versioning is enabled on the current bucket. The uploaded files or folders with the same name will be all kept in the bucket, but with different version IDs.

Drag and drop files or folders, or add files
(A maximum of 100 files can be uploaded at a time. The total size cannot exceed 5 GB.)

Server-Side Encryption: **Inherit from bucket** | SSE-KMS | SSE-OBS

Encryption is recommended to keep data secure. Any requests filled over the quota limit will be billed. Pricing details

If server-side encryption is enabled, new objects uploaded to this bucket are automatically encrypted

Step 4 Select a storage class. If you do not specify a storage class, the objects you upload inherit the default storage class of the bucket.

NOTE

An object can have a different storage class from its bucket. You can specify a storage class for an object when uploading it, or you can change the object storage class after the object is uploaded.

Step 5 Drag and drop the files or folders you want to upload to the **Upload Object** area.

You can also click **add files** in the **Upload Object** area to select files.

Step 6 **Server-Side Encryption**: Choose **Disable**, **SSE-KMS**, or **SSE-OBS**. For details, see **Server-Side Encryption**.

NOTE

If the bucket has Server-Side Encryption configured, an object you upload will inherit encryption from the bucket by default, but you can change the encryption option as required.

Step 7 (Optional) To configure WORM retention policies, or metadata, click **Next: (Optional) Configure Advanced Settings**.

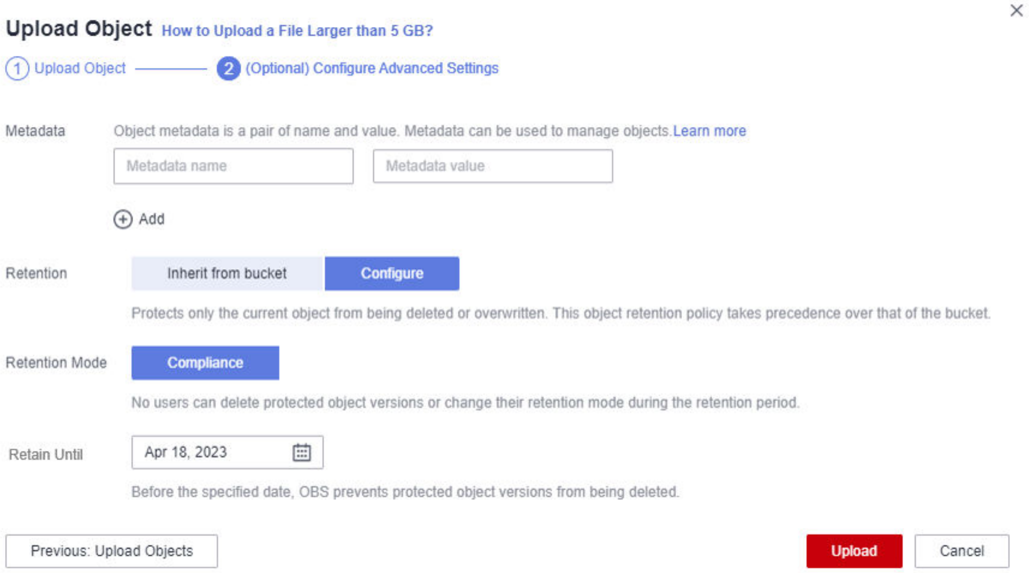
NOTE

WORM retention policies can be configured in the advanced settings only when WORM is enabled for the bucket.

Configuring metadata: Add metadata ContentDisposition, ContentLanguage, WebsiteRedirectLocation, ContentEncoding, or ContentType as needed. For more information, see **OBS Metadata**. Metadata is a set of name-value pairs. The metadata value cannot be left blank. You can add two or more metadata entries by clicking **Add**.

Configuring WORM retention: Choose **Inherit from bucket**, or choose **Configure** and then specify a retention period, to automatically protect new objects uploaded to the bucket from being deleted.

Figure 4-5 Configuring metadata or WORM retention



Upload Object How to Upload a File Larger than 5 GB?

1 Upload Object — 2 (Optional) Configure Advanced Settings

Metadata: Object metadata is a pair of name and value. Metadata can be used to manage objects.Learn more

Metadata name | Metadata value

Retention: Inherit from bucket | Configure

Protects only the current object from being deleted or overwritten. This object retention policy takes precedence over that of the bucket.

Retention Mode: Compliance

No users can delete protected object versions or change their retention mode during the retention period.

Retain Until: Apr 18, 2023

Before the specified date, OBS prevents protected object versions from being deleted.

Step 8 Click **Upload**.

----End

Using APIs

Uploading an Object

Using SDKs

Java	**Python**	C: not supported	Go: not supported	**BrowserJS**	.NET: not supported	**Android**	iOS: not supported	**PHP**	**Node.js**

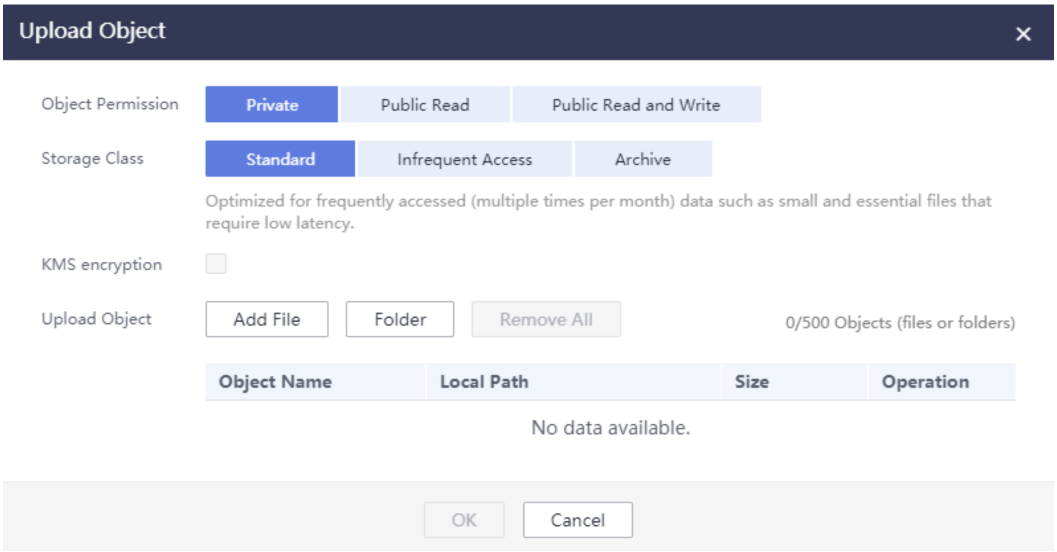
Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Click the bucket where you want to upload files or folders.

Step 3 Click **Upload** and then **Add File** or **Folder**, as shown in **Figure 4-6**.

Figure 4-6 Uploading a file or folder



Upload Object

Object Permission: Private | Public Read | Public Read and Write

Storage Class: Standard | Infrequent Access | Archive

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

KMS encryption □

Upload Object: Add File | Folder | Remove All   0/500 Objects (files or folders)

Object Name	Local Path	Size	Operation

No data available.

For better experience when using the **Add File** function, you are advised to upload a maximum of 100 files at a time. If you need to upload more, place all the files in a folder and upload them by adding a folder.

NOTE

1. If message "Service Unavailable" is displayed when files are being uploaded, try again later.
2. If an access deny message is displayed when you are uploading a file or folder, possible causes are as follows:

- Access to the bucket is restricted by an ACL. For example, you do not have the write permission for the bucket.
- Access to the bucket is restricted by a bucket policy. For example, you do not have the write permission for the bucket, or write operations cannot be performed on the bucket during the current period.

  If such message is displayed, check ACL and policy settings of the bucket and resolve the problem accordingly.

CAUTION

You must have access to the file you want to upload, or the file upload will fail.

Step 4 In the displayed dialog box, select the file or folder you want to upload and click **Open**.

You can upload one folder or multiple files at a time. To upload multiple files, hold down **Ctrl** or **Shift** to select multiple files and batch upload them. You can also press **Ctrl+A** to select all files. The operations are consistent with those in Windows operating systems.

Step 5 Click **OK**.

----End

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
  - Uploading a file
    ```
    obsutil cp file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-u] [-vlength] [-vmd5] [-p=1] [-threshold=5248800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading a folder
    ```
    obsutil cp folder_url obs://bucket[/key] -r [-arcDir=xxx] [-dryRun] [-link] [-f] [-flat] [-u] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading multiple files/folders
    ```
    obsutil cp file1_url,folder1_url|filelist_url obs://bucket[/prefix] -msm=1 [-r] [-arcDir=xxx] [-dryRun] [-link] [-f] [-u] [-vlength] [-vmd5] [-flat] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx][-timeRange=time1-time2] [-at] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```

    NOTE

    In this command, **/prefix** is the name prefix for uploading folders. For the execution examples, see **Upload**.
- In Linux or macOS

- Uploading a file

  ```
  ./obsutil cp file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-u] [-vlength] [-vmd5] [-p=1] [-threshold=5248800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
  ```

- Uploading a folder

  ```
  ./obsutil cp folder_url obs://bucket[/key] -r [-arcDir=xxx] [-dryRun] [-link] [-f] [-flat] [-u] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-at] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
  ```

- Uploading multiple files/folders

  ```
  ./obsutil cp file1_url,folder1_url|filelist_url obs://bucket[/prefix] -msm=1 [-r] [-arcDir=xxx] [-dryRun] [-link] [-f] [-u] [-vlength] [-vmd5] [-flat] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx][-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
  ```

  NOTE

  In this command, **/prefix** is the name prefix for uploading folders. For the execution examples, see **Upload**.

Examples

- Take the Windows OS as an example. Run the **obsutil cp d:\temp\test.txt obs://bucket-test/key** command to upload the **test.txt** file in the **temp** directory in the **D:** drive to bucket **bucket-test** and rename the file as **key**.

  ```
  obsutil cp d:\temp\test.txt obs://bucket-test/key

  Parallel:      5                   Jobs:          5
  Threshold:     50.00MB             PartSize:      auto
  VerifyLength:  false               VerifyMd5:     false
  CheckpointDir: C:\Users\Administrator\.obsutil_checkpoint

  [===================================================] 100.00% 1.68 MB/s 8.46MB/
  8.46MB 5s
  Upload successfully, 8.46MB, n/a, d:\temp\test.txt --> obs://bucket-test/key, cost [5], status [200],
  request id [0000016979E1D2B2860BB5181229C72C]
  ```

- Take the Windows OS as an example. Run the **obsutil cp d:\temp obs://bucket-test -f -r** command to recursively upload all files and subfolders in the **temp** directory in the **D:** drive to the **temp** folder in bucket **bucket-test**.

  ```
  obsutil cp d:\temp obs://bucket-test -f -r

  Parallel:      5                   Jobs:          5
  Threshold:     50.00MB             PartSize:      auto
  VerifyLength:  false               VerifyMd5:     false
  CheckpointDir: C:\Users\Administrator\.obsutil_checkpoint

  Task id: 104786c8-27c2-48fc-bc6a-5886596fb0ed
  OutputDir: C:\Users\Administrator\.obsutil_output

  [=======================================================] 100.00% tps:35.71 2.02 KB/s
  7.20MB/7.20MB 0s
  Succeed count is:   5         Failed count is:    0
  Metrics [max cost:90 ms, min cost:45 ms, average cost:63.80 ms, average tps:35.71]

  Task id: 104786c8-27c2-48fc-bc6a-5886596fb0ed
  ```

- For more examples, see **Upload**.

Parameter Description

Parameter	Optional or Mandatory	Description
file_url	Optional for uploading multiple files/ folders Mandatory for uploading a file	Local file path **NOTE** ● Do not nest paths when uploading multiple files/folders. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time. ● If this parameter is configured when uploading multiple files/folders, **msm** must be set to **1**. In this case, use commas (,) to separate multiple file paths, for example, **file_url1,file_url2**. ● Files and folders can both be included when uploading multiple files/folders. For example, **file_url1,folder_url1,file_url2,folder_url2**.
folder_url	Optional for uploading multiple files/ folders Mandatory for uploading a folder	Local folder path **NOTE** ● If **flat** is not configured when uploading a folder, the entire folder is uploaded. If **flat** is configured, all files in the folder are uploaded. ● Do not nest paths when uploading multiple files/folders. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time. ● If this parameter is configured when uploading multiple files/folders, **msm** must be set to **1**. In this case, use commas (,) to separate multiple folder paths, for example, **folder_url1,folder_url2**. ● Files and folders can both be included when uploading multiple files/folders. For example, **file_url1,folder_url1,file_url2,folder_url2**.
filelist_url	Optional for uploading multiple files/ folders	Indicates the path of the file that contains the list of files/folders to be uploaded. If this parameter is configured, **msm** must be set to **2**. **NOTE** ● The list file is in common text file formats, such as TXT and CSV. Each line in the file indicates a file or folder to be uploaded. For example: file_url1 file_url2 folder_url1 folder_url2 ● Do not nest paths in the list file. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time.
bucket	Mandatory	Bucket name

Parameter	Optional or Mandatory	Description
key	Optional	Indicates the object name or object name prefix specified when uploading a file, or the object name prefix specified when uploading a folder. The rules are as follows: ● If this parameter is left blank when uploading a file, the file is uploaded to the root directory of the bucket and the object name is the file name. If the value ends with a slash (/), the value is used as the object name prefix when the file is uploaded, and the object name is the value plus the file name. If the value does not end with a slash (/), the file is uploaded with the value as the object name. ● If this parameter is left blank when uploading a folder, the folder is uploaded to the root directory of the bucket. If the value ends with a slash (/), the value is used as the object name prefix of the folder to be uploaded. If the value does not end with a slash (/), the folder to be uploaded is prefixed with the value plus a slash (/). **NOTE** For details about how to use this parameter, see **Upload**.
fr	Optional for uploading a file (additional parameter)	Generates an operation result list when uploading a file.
flat	Optional for uploading a folder or multiple files/ folders (additional parameter)	Uploads all files in a folder but not the folder itself.
arcDir	Optional (additional parameter)	Path to which the uploaded files are archived
dryRun	Optional (additional parameter)	Conducts a dry run.

Parameter	Optional or Mandatory	Description
link	Optional (additional parameter)	Uploads the actual path of the symbolic-link file/folder **NOTICE** ● If this parameter is not specified and the file to be uploaded is a symbolic-link file whose target file does not exist, the exception message "The system cannot find the file specified" will be displayed in Windows OS, while the exception message "No such file or directory" will be displayed in macOS or Linux OS. ● Avoid the symbolic link loop of a folder, otherwise, the upload will exit due to panic. If you do not want the system to panic, set **panicForSymbolicLinkCircle** to **false** in the configuration file.
u	Optional (additional parameter)	Indicates incremental upload. If this parameter is set, each file can be uploaded only when it does not exist in the bucket, its size is different from the namesake one in the bucket, or it has the latest modification time. **CAUTION** When you compare each local file with data in the bucket, a billable HEAD request is generated. For details, see **Requests**.
vlength	Optional (additional parameter)	After the upload completes, check whether the sizes of the objects in the bucket are the same as those of the local files.
vmd5	Optional (additional parameter)	After the upload completes, check whether the MD5 values of the objects in the bucket are the same as those of the local files. **NOTE** ● If the size of the file or folder to be uploaded is too large, using this parameter will degrade the overall performance due to MD5 calculation. ● After the MD5 verification is successful, this parameter value is used for metadata **x-obs-meta-md5chksum**, for later MD5 verification during download or copy.
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart upload tasks when uploading a file. The default value is the value of **defaultParallels** in the configuration file.

Parameter	Optional or Mandatory	Description
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart upload, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the file or folder to be uploaded is smaller than the threshold, upload it directly. Otherwise, a multipart upload is required. ● If you upload a file or folder directly, no part record is generated, and resumable transmission is not supported. ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes).
acl	Optional (additional parameter)	Access control policies that can be specified when uploading files. Possible values are: ● private ● public-read ● public-read-write ● bucket-owner-full-control **NOTE** The preceding four values indicate private read and write, public read, public read and write, and bucket owner full control.
sc	Optional (additional parameter)	Indicates the storage classes of objects that can be specified when uploading files. Possible values are: ● **standard**: Standard storage class. It features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month) or data that is smaller than 1 MB. ● **warm**: Infrequent Access storage class. It is ideal for storing infrequently accessed (less than 12 times a year) data, but when needed, the access has to be fast. ● **cold**: Archive storage class. It provides secure, durable, and inexpensive storage for rarely-accessed (once a year) data. ● **deep-archive**: Deep Archive storage class (under limited beta testing). It is suitable for storing data that is barely (once every few years) accessed. This storage class costs less than the Archive storage class, but takes longer time (usually several hours) to restore data.

Parameter	Optional or Mandatory	Description
meta	Optional (additional parameter)	Indicates the standard and custom metadata that can be specified during file upload. This parameter should be configured in the following format: *key1*:*value1*#*key2*:*value2*#*key3*:*value3*. **NOTE** 1. The format example above indicates that the destination objects contain three groups of custom metadata: *key1:value1*, *key2:value2*, and *key3:value3*. 2. Standard metadata headers include **Content-Type**, **Content-Encoding**, **Cache-Control**, **Content-Disposition**, **Content-Language** and **Expires**.
ps	Optional (additional parameter)	Indicates the size of each part in a multipart upload task, in bytes. The value ranges from **100KB** to **5GB**. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes). ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source file size.
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart upload and saved to the **upload** subfolder. After the upload succeeds, its part record is deleted automatically. If the upload fails or is suspended, the system attempts to resume the task according to its part record when you perform the upload the next time.
r	Mandatory for uploading a folder (additional parameter) Optional for uploading multiple files/ folders	Indicates files and subfolders within the folder when uploading a folder recursively.

Parameter	Optional or Mandatory	Description
f	Optional for uploading a folder or multiple files/ folders (additional parameter)	Runs in force mode.
j	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the maximum number of concurrent tasks for uploading a folder. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
msm	Mandatory for uploading multiple files/ folders (additional parameter)	Enables the mode for uploading multiple files/folders. Possible values are **1** and **2**. **NOTE** ● If **msm** is set to **1**, the source URL indicates a list of file/folder names separated by commas. ● If **msm** is set to **2**, the source URL indicates a file containing a list of file/folder names. ● If the file or folder name already contains commas (,), do not set **msm** to **1**. ● If parameter **r** is not set, the folders in the list will not be uploaded.

Parameter	Optional or Mandatory	Description
exclude	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the file matching patterns that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use **\*** to represent ***** and **\\?** to represent **?**. ● If the name of the file to be uploaded matches the value of this parameter, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute file path (including the file name and file directory). ● The matching pattern takes effect only for files in the folder. ● Multiple **exclude** parameters can be specified, for example, **-exclude=*.xxx -exclude=*.xxx**.

Parameter	Optional or Mandatory	Description
include	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the file matching patterns that are included, for example: ***.jpg**. **NOTE** - The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. - You can use **\*** to represent ***** and **\\?** to represent **?**. - Only after identifying that the name of the file to be uploaded does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is uploaded. If not, the file is skipped. **NOTE** Example of uploading files in a request with the **include** parameter contained: `./obsutil cp /localpath/ obs://test/ -include=/localpath/2022-12-09/* -f -r` This command uploads files that are under **localpath** and start with **/localpath/2022-12-09/** to bucket **test**. **NOTICE** - You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. - The matching pattern applies to the absolute file path (including the file name and file directory). - The matching pattern takes effect only for files in the folder. - Multiple **include** parameters can be specified, for example, **-include=*.xxx -include=*.xxx**.
at	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates that only the files whose latest access time is within the value of **timeRange** are uploaded. **NOTE** - This parameter must be used together with **timeRange**.
disableDirObject	Optional for uploading multiple folders (additional parameter)	Indicates the folders themselves are not uploaded as an object. Configuring this parameter can avoid uploading empty folders to a bucket. If a folder contains files, the files will be uploaded and the original path format is retained.

Parameter	Optional or Mandatory	Description
timeRange	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the time range matching pattern when uploading files. Only files whose latest modification time is within the configured time range are uploaded. This pattern has a lower priority than the file matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured file matching patterns. **NOTE** ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-***time2*, all files whose latest modification time is earlier than *time2* are matched. If it is set to *time1***-***, all files whose latest modification time is later than *time1* are matched. **NOTICE** Time in the matching pattern is the UTC time.
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on folders.

Parameter	Optional or Mandatory	Description
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **cp_{succeed | failed | warning}_report_***time***_TaskId.txt** ● By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file. ● If there are multiple folders and files and you need to confirm the detailed error information about a failed task, refer to the failure list **cp_failed_report_***time***_TaskId.txt** in the result list folder and the **log files** in the log path.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

Response

Field	Description
Parallel	Parameter **-p** in the request

Field	Description
Jobs	Parameter **-j** in the request
Threshold	Parameter **-threshold** in the request
PartSize	Parameter **-ps** in the request
Exclude	Parameter **-exclude** in the request
Include	Parameter **-include** in the request
TimeRange	Parameter **-timeRange** in the request
VerifyLength	Parameter **-vlength** in the request
VerifyMd5	Parameter **-vmd5** in the request
CheckpointDir	Parameter **-cpd** in the request
OutputDir	Parameter **-o** in the request
ArcDir	Parameter **-arcDir** in the request
Succeed count	Number of successful tasks
Failed count	Number of failed tasks
Skip count	Number of tasks that are skipped during incremental upload, download, or copy, and synchronous upload, download, or copy. **NOTE** Skipped tasks are recorded into successful tasks.
Warning count	Number of tasks that are executed successfully but contain warnings. **NOTE** ● The task for which a warning is generated may be a failure or a success, which needs to be further determined according to the corresponding result list. ● The number of tasks that generate warnings is independent of the number of successful or failed tasks. The total number of tasks is the number of successful tasks plus the number of failed tasks.
Succeed bytes	Number of bytes that are successfully uploaded or downloaded.
max cost	Maximum duration of all tasks, in ms
min cost	Minimum duration of all tasks, in ms
average cost	Average duration of all tasks, in ms
average tps	The average number of tasks completed per second
Task id	Unique ID of an operation, which is used to search for the result list generated in a batch task

4.2.4 Multipart Upload

Scenarios

Multipart upload allows you to upload a single object as a group of parts. Each part is a contiguous one of the object data. You can upload these parts separately and in any sequence. If a part fails to be uploaded, you can upload it again without affecting other parts. After all parts are uploaded, OBS assembles these parts to create the object.

Generally, if the size of an object reaches 100 MB, multipart upload is recommended. For example, if you want to upload a 500 MB object to an OBS bucket, you can use OBS Browser+ for multipart upload. OBS Browser+ divides the object into small parts and then uploads the parts. Alternatively, you can call the multipart upload API, improving upload efficiency and reducing failures.

Advantages of Multipart Upload

- Improved throughput: You can upload parts in parallel to improve throughput.
- Quick recovery from network errors: Small parts minimize the impact of restarting a failed uploading caused by network errors.
- Convenient pause and resuming of object uploads: You can upload parts at any time. A multipart upload will not expire after being initiated. You must explicitly complete or abort a multipart upload.
- Starting uploading before knowing the object size: You can upload an object while creating it.

Constraints

Table 4-1 Constraints on multipart upload

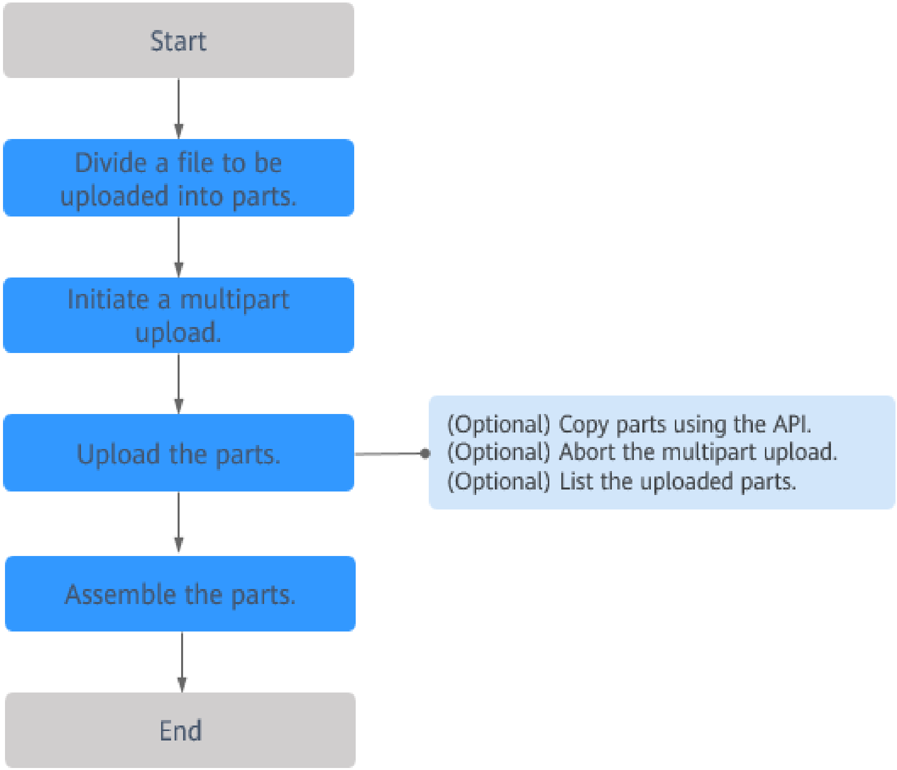
Item	Constraints
Maximum object size	48.8 TB
Maximum number of parts for each upload task	10,000
Part number	1 to 10,000 (inclusive)
Part size	5 MB to 5 GB. The size of the last part is between 0 bytes to 5 GB.
Maximum number of parts returned in response to the request for listing uploaded parts	1,000
Maximum number of multipart uploads returned in response to the request for listing initiated multipart uploads	1,000

NOTE

If you have over 48.8 TB data to upload, refer to **Migrating Local Data to OBS**.

Multipart Upload Process

Figure 4-7 Multipart upload


Start
Divide a file to be uploaded into parts.
Initiate a multipart upload.
Upload the parts.
(Optional) Copy parts using the API.
(Optional) Abort the multipart upload.
(Optional) List the uploaded parts.
Assemble the parts.
End

1. **Divide a file to be uploaded into parts.**
2. **Initiate a multipart upload.**

   When you send a request to start multipart upload, OBS returns a response with the upload ID, which is the unique identifier of the multipart upload. This ID must be included in the request for uploading parts, listing uploaded parts, completing a multipart upload, or aborting a multipart upload.
3. **Upload parts.**

   When uploading a part, you must specify the upload ID and a part number. You can select any part number between 1 and 10,000. A part number uniquely identifies a part and its location in the object you are uploading. If the number of an uploaded part is used to upload a new part, the uploaded part will be overwritten.

   Whenever you upload a part, OBS returns the ETag header in the response. For each part upload, you must record the part number and the ETag value. These part numbers and ETag values are required in subsequent requests to complete the multipart upload task.

   When concurrent upload operations are performed for the same part of an object, the server complies with the Last Write Wins policy, but the time referred in Last Write is the time when the part metadata is created. To ensure data accuracy, the client must be locked during the concurrent upload for the same part of an object. Concurrent upload for different parts of an object does not require the client to be locked.

4. **(Optional) Copy parts.**

   After initiating a multipart upload, you can specify upload ID for the multipart upload and call the multipart copy API to copy part or all of the uploaded object as parts.

   If you copy the source object as a part called part1 and another part1 already exists before the copy operation, the original part1 will be overwritten by the new one after the copy operation. After the copy succeeds, only the new part1 can be listed and the original part1 will be deleted. Therefore, ensure that the original part does not exist or has no value when copying a part. Otherwise, data may be deleted by mistake. The source object does not change during the copy.

   NOTICE

   You cannot determine whether a request is successful only based on the **status_code** in the returned HTTP header. If **200** is returned for **status_code**, the server has received the request and started to process the request. The copy is successful only when the body in the response contains ETag.

5. **(Optional) Abort the multipart upload.**

   You can abort a multipart upload. After a multipart upload is aborted, the upload ID cannot be used to upload any part. Then, OBS releases the storage of all uploaded parts. If you stop an ongoing multipart upload, the uploading will still complete and the result can be successful or failed. To release the storage capacity occupied by all parts, you need to abort the multipart upload after the entire task is complete.

6. **(Optional) List parts.**

   – Listing uploaded parts

     You can list the parts of a specific multipart upload task or the parts of all the multipart upload tasks in progress. Information about uploaded parts in a specific multipart upload will be returned for a request to list uploaded parts. For each request to list uploaded parts, OBS returns information about the uploaded parts in the specific multipart upload. Information about a maximum of 1,000 parts can be returned. If there are more than 1,000 parts in a multipart upload, you need to send multiple requests to list all uploaded parts. The list of uploaded parts does not include assembled parts.

     NOTICE

     A returned list can only be used for verification. After a multipart upload is complete, the result in the list is no longer valid. However, when part numbers and the ETag values returned by OBS are uploaded, the list of your specified part numbers will be reserved.

– Listing multipart uploads

You can list initiated multipart uploads by listing the multipart uploads in the bucket. Initiated multipart uploads refer to the multipart uploads that are not assembled or aborted after initiation. A maximum of 1,000 multipart uploads can be returned for each request. If there are more than 1,000 multipart uploads in progress, you need to send more requests to query the remaining tasks.

7. **Assemble parts.**

   OBS creates an object by assembling the parts in ascending order based on the part number. If any object metadata is provided for initiating a multipart upload task, OBS associates the metadata with the object. After the request for the multipart upload is complete, the parts will no longer exist. A part assembling request must contain the upload ID, part numbers, and a list of corresponding ETag values. An OBS response includes an ETag that uniquely identifies the combined object data. The ETag is not an MD5 hash of the object data.

   NOTICE

   - After the multipart upload task is initiated and one or more parts are uploaded, you must assemble the parts or abort the multipart upload. Otherwise, you have to pay for the storage fee of the uploaded parts. OBS releases the storage and stops charging the storage fee only after the uploaded parts are assembled or the multipart upload is aborted.
   - If 10 parts are uploaded but only nine parts are assembled, the parts that are not assembled will be automatically deleted by the system and cannot be restored after being deleted. Before assembling the parts, adopt the API used to list the parts that have been uploaded to check all parts to ensure that no part is missed.

Permissions

You can perform multipart upload only after being granted with the permission. You can use ACLs, bucket policies, or user policies to grant users the permission. The following table lists multipart upload operations and the required permissions that can be granted by ACLs, bucket policies, or user policies.

Operation	Required Permissions
Initiate a multipart upload.	To perform this operation, you need to have the PutObject permission. A bucket owner can grant the PutObject permission to others.

Operation	Required Permissions
Upload parts.	To perform this operation, you need to have the PutObject permission. Only the initiator of a multipart upload can upload parts. The bucket owner must grant the multipart upload initiator the PutObject permission so that the initiator can upload parts of the object.
Copy parts.	To perform this operation, you need to have the PutObject permission as well as the GetObject permission on the object to be copied. Only the initiator of a multipart upload can copy parts. The bucket owner must grant the multipart upload initiator the PutObject permission so that the initiator can upload parts of the object.
Assemble parts.	To perform this operation, you need to have the PutObject permission. Only the initiator of a multipart upload can assemble parts. The bucket owner must grant the multipart upload initiator the PutObject permission so that the initiator can complete the multipart upload.
Abort the multipart upload.	To perform this operation, you need to have the AbortMultipartUpload permission. By default, only the bucket owner and the multipart upload initiator have this permission. In addition to the default configuration, the bucket owner can allow trustees to perform this operation. The bucket owner can also deny any trustees performing this operation.

Operation	Required Permissions
List uploaded parts.	To perform this operation, you need to have the ListMultipartUploadParts permission. By default, the bucket owner can list the uploaded parts of any multipart upload to the bucket. The multipart upload initiator can list the uploaded parts of a specific multipart upload. In addition to the default configuration, the bucket owner can allow trustees to perform this operation. The bucket owner can also deny any trustees performing this operation.
List multipart uploads.	To list multipart uploads to the bucket, you need to have the ListBucketMultipartUploads permission. In addition to the default configuration, the bucket owner can allow trustees to perform this operation.

Important Notes

- A directory cannot be uploaded. Only one object can be uploaded at a time.
- Each request can upload only one part, but multiple requests can be initiated at the same time.
- If you want to upload a large number of Deep Archive objects, you can upload them in the Standard storage class and then transition them to the Deep Archive storage class through **lifecycle rules** to lower the costs on PUT requests.
- If a large number of objects need to be uploaded, do not name the objects using sequential prefixes, such as timestamps or alphabetical order. Objects named with sequential prefixes may be stored in a specific storage partition. In such case, if there are a large number of access requests for the objects, the requests cannot be handled efficiently.
- It is recommended to enable versioning to prevent objects with the same name from being overwritten. Versioning keeps every version of objects in the same bucket. You can restore any historical version at any time.
- The checkpoint file records the status of uploads using SDKs. Ensure that you have write permissions for the checkpoint file.
- Do not modify the verification information in the checkpoint file. If the checkpoint file is damaged, all fragments will be uploaded again.
- If the local file changes during the upload, all fragments will be uploaded again.

Ways to Upload

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to upload objects.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate.

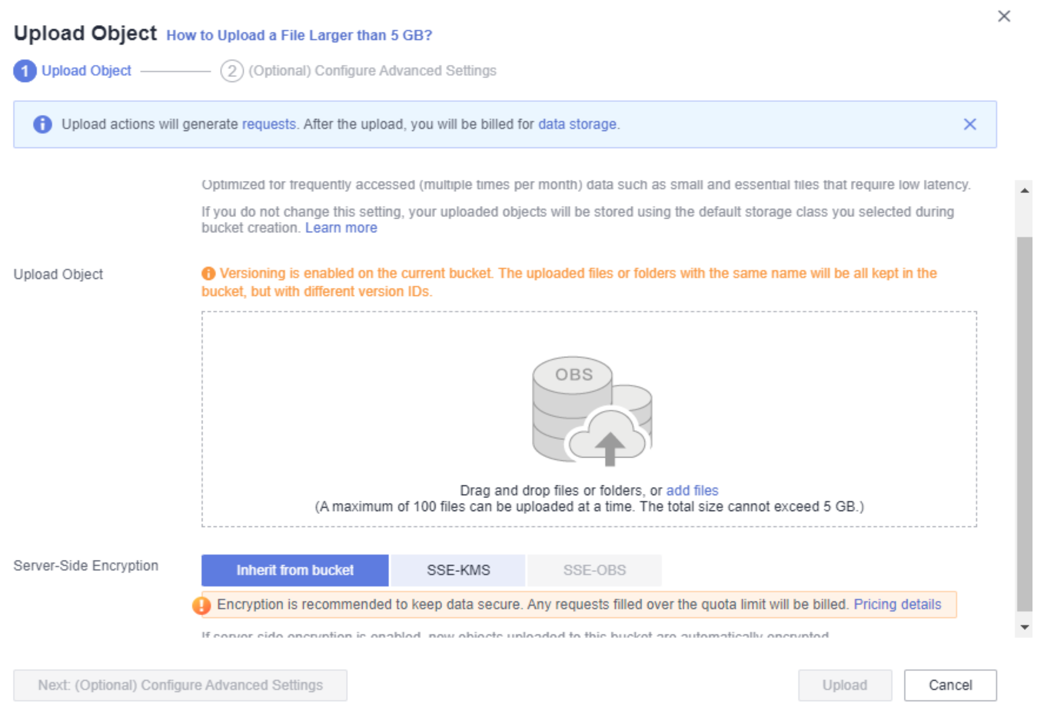
Step 3 Go to the folder that you want to upload files to and click **Upload Object**. The **Upload Object** dialog box is displayed.

Batch upload is used as an example here. If a region only supports single upload, follow the on-screen instructions.

NOTE

If the files that you want to upload to OBS are stored in Microsoft OneDrive, it is recommended that the names of these files contain a maximum of 32 characters to ensure compatibility.

Figure 4-8 Object Upload



Upload Object How to Upload a File Larger than 5 GB?

1 Upload Object —— 2 (Optional) Configure Advanced Settings

Upload actions will generate requests. After the upload, you will be billed for data storage.

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

If you do not change this setting, your uploaded objects will be stored using the default storage class you selected during bucket creation. Learn more

Upload Object

Versioning is enabled on the current bucket. The uploaded files or folders with the same name will be all kept in the bucket, but with different version IDs.

Drag and drop files or folders, or add files
(A maximum of 100 files can be uploaded at a time. The total size cannot exceed 5 GB.)

Server-Side Encryption: Inherit from bucket | SSE-KMS | SSE-OBS

Encryption is recommended to keep data secure. Any requests filled over the quota limit will be billed. Pricing details

If server-side encryption is enabled, new objects uploaded to this bucket are automatically encrypted

Step 4 Select a storage class. If you do not specify a storage class, the objects you upload inherit the default storage class of the bucket.

NOTE

An object can have a different storage class from its bucket. You can specify a storage class for an object when uploading it, or you can change the object storage class after the object is uploaded.

Step 5 Drag and drop the files or folders you want to upload to the **Upload Object** area.

You can also click **add files** in the **Upload Object** area to select files.

Step 6 **Server-Side Encryption**: Choose **Disable**, **SSE-KMS**, or **SSE-OBS**. For details, see **Server-Side Encryption**.

NOTE

If the bucket has Server-Side Encryption configured, an object you upload will inherit encryption from the bucket by default, but you can change the encryption option as required.

Step 7 (Optional) To configure WORM retention policies, or metadata, click **Next: (Optional) Configure Advanced Settings**.

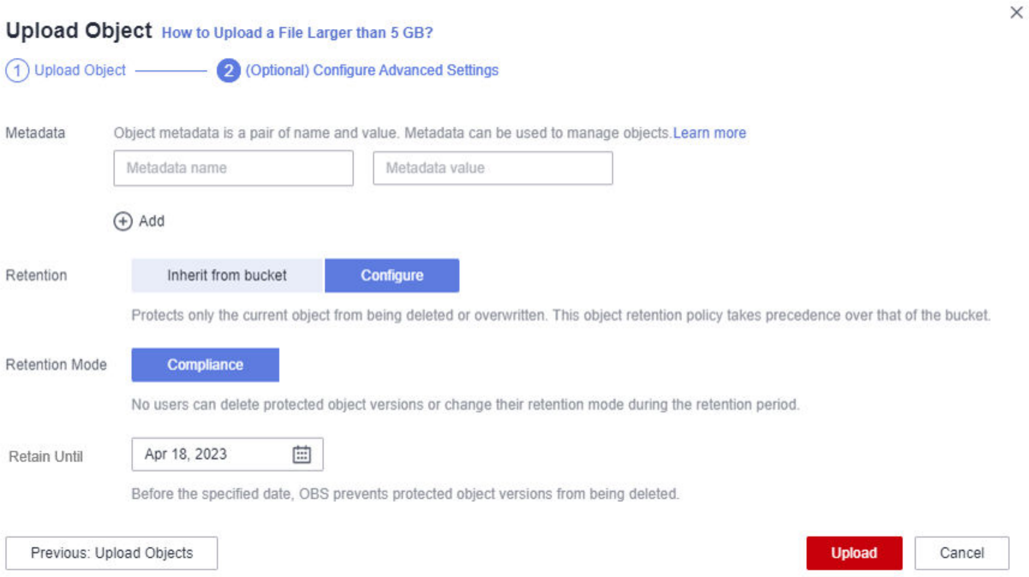
NOTE

WORM retention policies can be configured in the advanced settings only when WORM is enabled for the bucket.

Configuring metadata: Add metadata ContentDisposition, ContentLanguage, WebsiteRedirectLocation, ContentEncoding, or ContentType as needed. For more information, see **OBS Metadata**. Metadata is a set of name-value pairs. The metadata value cannot be left blank. You can add two or more metadata entries by clicking **Add**.

Configuring WORM retention: Choose **Inherit from bucket**, or choose **Configure** and then specify a retention period, to automatically protect new objects uploaded to the bucket from being deleted.

Figure 4-9 Configuring metadata or WORM retention



Upload Object How to Upload a File Larger than 5 GB?

1 Upload Object — 2 (Optional) Configure Advanced Settings

Metadata: Object metadata is a pair of name and value. Metadata can be used to manage objects.Learn more

Metadata name | Metadata value

Add

Retention: Inherit from bucket | Configure

Protects only the current object from being deleted or overwritten. This object retention policy takes precedence over that of the bucket.

Retention Mode: Compliance

No users can delete protected object versions or change their retention mode during the retention period.

Retain Until: Apr 18, 2023

Before the specified date, OBS prevents protected object versions from being deleted.

Step 8 Click **Upload**.

----End

Using APIs

Operations on Multipart Upload

Using SDKs

Java	Python: not supported	C	Go: not supported	BrowserJS	.NET	Android	iOS	PHP	Node.js

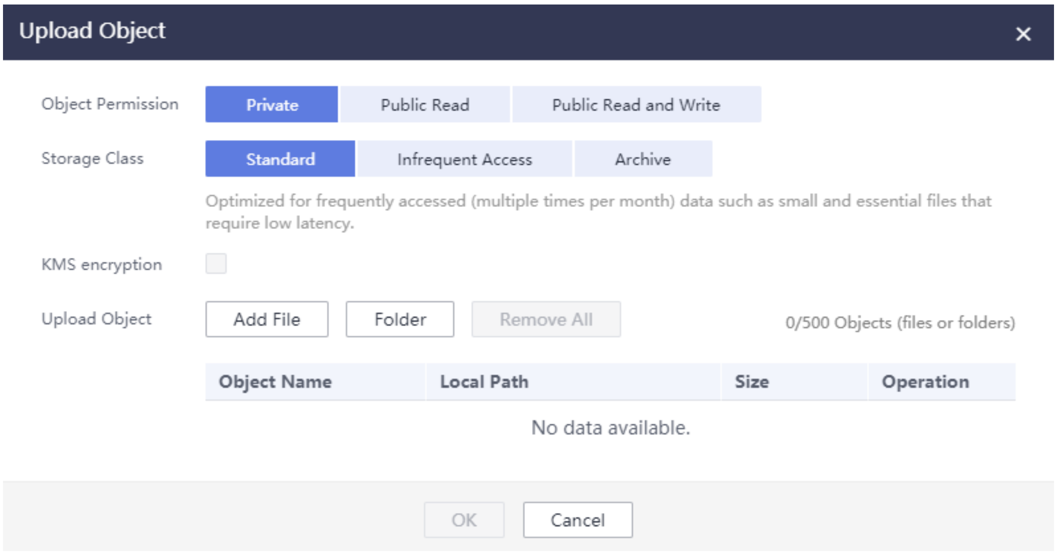
Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Click the bucket where you want to upload files or folders.

Step 3 Click **Upload** and then **Add File** or **Folder**, as shown in **Figure 4-10**.

Figure 4-10 Uploading a file or folder



Upload Object

Object Permission: Private | Public Read | Public Read and Write

Storage Class: Standard | Infrequent Access | Archive

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

KMS encryption □

Upload Object: Add File | Folder | Remove All 0/500 Objects (files or folders)

Object Name	Local Path	Size	Operation

No data available.

For better experience when using the **Add File** function, you are advised to upload a maximum of 100 files at a time. If you need to upload more, place all the files in a folder and upload them by adding a folder.

NOTE

1. If message "Service Unavailable" is displayed when files are being uploaded, try again later.
2. If an access deny message is displayed when you are uploading a file or folder, possible causes are as follows:

- Access to the bucket is restricted by an ACL. For example, you do not have the write permission for the bucket.
- Access to the bucket is restricted by a bucket policy. For example, you do not have the write permission for the bucket, or write operations cannot be performed on the bucket during the current period.

  If such message is displayed, check ACL and policy settings of the bucket and resolve the problem accordingly.

CAUTION

You must have access to the file you want to upload, or the file upload will fail.

Step 4 In the displayed dialog box, select the file or folder you want to upload and click **Open**.

You can upload one folder or multiple files at a time. To upload multiple files, hold down **Ctrl** or **Shift** to select multiple files and batch upload them. You can also press **Ctrl+A** to select all files. The operations are consistent with those in Windows operating systems.

Step 5 Click **OK**.

----End

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
  - Uploading a file
    ```
    obsutil cp file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-u] [-vlength] [-vmd5] [-
    p=1] [-threshold=5248800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-
    cpd=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading a folder
    ```
    obsutil cp folder_url obs://bucket[/key] -r [-arcDir=xxx] [-dryRun] [-link] [-f] [-flat] [-u] [-
    vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-
    meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2]
    [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading multiple files/folders
    ```
    obsutil cp file1_url,folder1_url|filelist_url obs://bucket[/prefix] -msm=1 [-r] [-arcDir=xxx] [-
    dryRun] [-link] [-f] [-u] [-vlength] [-vmd5] [-flat] [-j=1] [-p=1] [-threshold=52428800] [-
    acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx][-
    timeRange=time1-time2] [-at] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-
    k=xxx] [-t=xxx]
    ```

    NOTE

    In this command, **/prefix** is the name prefix for uploading folders. For the execution examples, see **Upload**.
- In Linux or macOS

- Uploading a file

  ```
  ./obsutil cp file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-u] [-vlength] [-vmd5] [-p=1] [-threshold=5248800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
  ```

- Uploading a folder

  ```
  ./obsutil cp folder_url obs://bucket[/key] -r [-arcDir=xxx] [-dryRun] [-link] [-f] [-flat] [-u] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-at] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
  ```

- Uploading multiple files/folders

  ```
  ./obsutil cp file1_url,folder1_url|filelist_url obs://bucket[/prefix] -msm=1 [-r] [-arcDir=xxx] [-dryRun] [-link] [-f] [-u] [-vlength] [-vmd5] [-flat] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx][-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
  ```

  NOTE

  In this command, **/prefix** is the name prefix for uploading folders. For the execution examples, see **Upload**.

Examples

- Take the Windows OS as an example. Run the **obsutil cp d:\temp\test.txt obs://bucket-test/key** command to upload the **test.txt** file in the **temp** directory in the **D:** drive to bucket **bucket-test** and rename the file as **key**.

  ```
  obsutil cp d:\temp\test.txt obs://bucket-test/key

  Parallel:      5                   Jobs:          5
  Threshold:     50.00MB              PartSize:      auto
  VerifyLength:  false                VerifyMd5:     false
  CheckpointDir: C:\Users\Administrator\.obsutil_checkpoint

  [===================================================] 100.00% 1.68 MB/s 8.46MB/
  8.46MB 5s
  Upload successfully, 8.46MB, n/a, d:\temp\test.txt --> obs://bucket-test/key, cost [5], status [200],
  request id [0000016979E1D2B2860BB5181229C72C]
  ```

- Take the Windows OS as an example. Run the **obsutil cp d:\temp obs://bucket-test -f -r** command to recursively upload all files and subfolders in the **temp** directory in the **D:** drive to the **temp** folder in bucket **bucket-test**.

  ```
  obsutil cp d:\temp obs://bucket-test -f -r

  Parallel:      5                   Jobs:          5
  Threshold:     50.00MB              PartSize:      auto
  VerifyLength:  false                VerifyMd5:     false
  CheckpointDir: C:\Users\Administrator\.obsutil_checkpoint

  Task id: 104786c8-27c2-48fc-bc6a-5886596fb0ed
  OutputDir: C:\Users\Administrator\.obsutil_output

  [========================================================] 100.00% tps:35.71 2.02 KB/s
  7.20MB/7.20MB 0s
  Succeed count is:   5          Failed count is:    0
  Metrics [max cost:90 ms, min cost:45 ms, average cost:63.80 ms, average tps:35.71]

  Task id: 104786c8-27c2-48fc-bc6a-5886596fb0ed
  ```

- For more examples, see **Upload**.

Parameter Description

Parameter	Optional or Mandatory	Description
file_url	Optional for uploading multiple files/ folders Mandatory for uploading a file	Local file path **NOTE** ● Do not nest paths when uploading multiple files/folders. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time. ● If this parameter is configured when uploading multiple files/folders, **msm** must be set to **1**. In this case, use commas (,) to separate multiple file paths, for example, **file_url1,file_url2**. ● Files and folders can both be included when uploading multiple files/folders. For example, **file_url1,folder_url1,file_url2,folder_url2**.
folder_url	Optional for uploading multiple files/ folders Mandatory for uploading a folder	Local folder path **NOTE** ● If **flat** is not configured when uploading a folder, the entire folder is uploaded. If **flat** is configured, all files in the folder are uploaded. ● Do not nest paths when uploading multiple files/folders. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time. ● If this parameter is configured when uploading multiple files/folders, **msm** must be set to **1**. In this case, use commas (,) to separate multiple folder paths, for example, **folder_url1,folder_url2**. ● Files and folders can both be included when uploading multiple files/folders. For example, **file_url1,folder_url1,file_url2,folder_url2**.
filelist_url	Optional for uploading multiple files/ folders	Indicates the path of the file that contains the list of files/folders to be uploaded. If this parameter is configured, **msm** must be set to **2**. **NOTE** ● The list file is in common text file formats, such as TXT and CSV. Each line in the file indicates a file or folder to be uploaded. For example: file_url1 file_url2 folder_url1 folder_url2 ● Do not nest paths in the list file. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time.
bucket	Mandatory	Bucket name

Parameter	Optional or Mandatory	Description
key	Optional	Indicates the object name or object name prefix specified when uploading a file, or the object name prefix specified when uploading a folder. The rules are as follows: ● If this parameter is left blank when uploading a file, the file is uploaded to the root directory of the bucket and the object name is the file name. If the value ends with a slash (/), the value is used as the object name prefix when the file is uploaded, and the object name is the value plus the file name. If the value does not end with a slash (/), the file is uploaded with the value as the object name. ● If this parameter is left blank when uploading a folder, the folder is uploaded to the root directory of the bucket. If the value ends with a slash (/), the value is used as the object name prefix of the folder to be uploaded. If the value does not end with a slash (/), the folder to be uploaded is prefixed with the value plus a slash (/). **NOTE** For details about how to use this parameter, see **Upload**.
fr	Optional for uploading a file (additional parameter)	Generates an operation result list when uploading a file.
flat	Optional for uploading a folder or multiple files/ folders (additional parameter)	Uploads all files in a folder but not the folder itself.
arcDir	Optional (additional parameter)	Path to which the uploaded files are archived
dryRun	Optional (additional parameter)	Conducts a dry run.

Parameter	Optional or Mandatory	Description
link	Optional (additional parameter)	Uploads the actual path of the symbolic-link file/folder **NOTICE** - If this parameter is not specified and the file to be uploaded is a symbolic-link file whose target file does not exist, the exception message "The system cannot find the file specified" will be displayed in Windows OS, while the exception message "No such file or directory" will be displayed in macOS or Linux OS. - Avoid the symbolic link loop of a folder, otherwise, the upload will exit due to panic. If you do not want the system to panic, set **panicForSymbolicLinkCircle** to **false** in the configuration file.
u	Optional (additional parameter)	Indicates incremental upload. If this parameter is set, each file can be uploaded only when it does not exist in the bucket, its size is different from the namesake one in the bucket, or it has the latest modification time. **CAUTION** When you compare each local file with data in the bucket, a billable HEAD request is generated. For details, see **Requests**.
vlength	Optional (additional parameter)	After the upload completes, check whether the sizes of the objects in the bucket are the same as those of the local files.
vmd5	Optional (additional parameter)	After the upload completes, check whether the MD5 values of the objects in the bucket are the same as those of the local files. **NOTE** - If the size of the file or folder to be uploaded is too large, using this parameter will degrade the overall performance due to MD5 calculation. - After the MD5 verification is successful, this parameter value is used for metadata **x-obs-meta-md5chksum**, for later MD5 verification during download or copy.
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart upload tasks when uploading a file. The default value is the value of **defaultParallels** in the configuration file.

Parameter	Optional or Mandatory	Description
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart upload, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the file or folder to be uploaded is smaller than the threshold, upload it directly. Otherwise, a multipart upload is required. ● If you upload a file or folder directly, no part record is generated, and resumable transmission is not supported. ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes).
acl	Optional (additional parameter)	Access control policies that can be specified when uploading files. Possible values are: ● private ● public-read ● public-read-write ● bucket-owner-full-control **NOTE** The preceding four values indicate private read and write, public read, public read and write, and bucket owner full control.
sc	Optional (additional parameter)	Indicates the storage classes of objects that can be specified when uploading files. Possible values are: ● **standard**: Standard storage class. It features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month) or data that is smaller than 1 MB. ● **warm**: Infrequent Access storage class. It is ideal for storing infrequently accessed (less than 12 times a year) data, but when needed, the access has to be fast. ● **cold**: Archive storage class. It provides secure, durable, and inexpensive storage for rarely-accessed (once a year) data. ● **deep-archive**: Deep Archive storage class (under limited beta testing). It is suitable for storing data that is barely (once every few years) accessed. This storage class costs less than the Archive storage class, but takes longer time (usually several hours) to restore data.

Parameter	Optional or Mandatory	Description
meta	Optional (additional parameter)	Indicates the standard and custom metadata that can be specified during file upload. This parameter should be configured in the following format: *key1*:*value1*#*key2*:*value2*#*key3*:*value3*. **NOTE** 1. The format example above indicates that the destination objects contain three groups of custom metadata: *key1:value1*, *key2:value2*, and *key3:value3*. 2. Standard metadata headers include **Content-Type**, **Content-Encoding**, **Cache-Control**, **Content-Disposition**, **Content-Language** and **Expires**.
ps	Optional (additional parameter)	Indicates the size of each part in a multipart upload task, in bytes. The value ranges from **100KB** to **5GB**. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes). ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source file size.
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart upload and saved to the **upload** subfolder. After the upload succeeds, its part record is deleted automatically. If the upload fails or is suspended, the system attempts to resume the task according to its part record when you perform the upload the next time.
r	Mandatory for uploading a folder (additional parameter) Optional for uploading multiple files/ folders	Indicates files and subfolders within the folder when uploading a folder recursively.

Parameter	Optional or Mandatory	Description
f	Optional for uploading a folder or multiple files/ folders (additional parameter)	Runs in force mode.
j	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the maximum number of concurrent tasks for uploading a folder. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
msm	Mandatory for uploading multiple files/ folders (additional parameter)	Enables the mode for uploading multiple files/folders. Possible values are **1** and **2**. **NOTE** ● If **msm** is set to **1**, the source URL indicates a list of file/folder names separated by commas. ● If **msm** is set to **2**, the source URL indicates a file containing a list of file/folder names. ● If the file or folder name already contains commas (,), do not set **msm** to **1**. ● If parameter **r** is not set, the folders in the list will not be uploaded.

Parameter	Optional or Mandatory	Description
exclude	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the file matching patterns that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use **\*** to represent ***** and **\\?** to represent **?**. ● If the name of the file to be uploaded matches the value of this parameter, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute file path (including the file name and file directory). ● The matching pattern takes effect only for files in the folder. ● Multiple **exclude** parameters can be specified, for example, **-exclude=*.xxx -exclude=*.xxx**.

Parameter	Optional or Mandatory	Description
include	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the file matching patterns that are included, for example: ***.jpg**. **NOTE** - The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. - You can use **\*** to represent ***** and **\\?** to represent **?**. - Only after identifying that the name of the file to be uploaded does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is uploaded. If not, the file is skipped. **NOTE** Example of uploading files in a request with the **include** parameter contained: `./obsutil cp /localpath/ obs://test/ -include=/localpath/2022-12-09/* -f -r` This command uploads files that are under **localpath** and start with **/localpath/2022-12-09/** to bucket **test**. **NOTICE** - You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. - The matching pattern applies to the absolute file path (including the file name and file directory). - The matching pattern takes effect only for files in the folder. - Multiple **include** parameters can be specified, for example, **-include=*.xxx -include=*.xxx**.
at	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates that only the files whose latest access time is within the value of **timeRange** are uploaded. **NOTE** - This parameter must be used together with **timeRange**.
disableDir Object	Optional for uploading multiple folders (additional parameter)	Indicates the folders themselves are not uploaded as an object. Configuring this parameter can avoid uploading empty folders to a bucket. If a folder contains files, the files will be uploaded and the original path format is retained.

Parameter	Optional or Mandatory	Description
timeRange	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the time range matching pattern when uploading files. Only files whose latest modification time is within the configured time range are uploaded. This pattern has a lower priority than the file matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured file matching patterns. **NOTE** ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-***time2*, all files whose latest modification time is earlier than *time2* are matched. If it is set to *time1***-***, all files whose latest modification time is later than *time1* are matched. **NOTICE** Time in the matching pattern is the UTC time.
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on folders.

Parameter	Optional or Mandatory	Description
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** - The naming rule for result lists is as follows: **cp_{succeed | failed | warning}_report_***time***_TaskId.txt** - By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file. - If there are multiple folders and files and you need to confirm the detailed error information about a failed task, refer to the failure list **cp_failed_report_***time***_TaskId.txt** in the result list folder and the **log files** in the log path.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

Response

Field	Description
Parallel	Parameter **-p** in the request

Field	Description
Jobs	Parameter **-j** in the request
Threshold	Parameter **-threshold** in the request
PartSize	Parameter **-ps** in the request
Exclude	Parameter **-exclude** in the request
Include	Parameter **-include** in the request
TimeRange	Parameter **-timeRange** in the request
VerifyLength	Parameter **-vlength** in the request
VerifyMd5	Parameter **-vmd5** in the request
CheckpointDir	Parameter **-cpd** in the request
OutputDir	Parameter **-o** in the request
ArcDir	Parameter **-arcDir** in the request
Succeed count	Number of successful tasks
Failed count	Number of failed tasks
Skip count	Number of tasks that are skipped during incremental upload, download, or copy, and synchronous upload, download, or copy. **NOTE** Skipped tasks are recorded into successful tasks.
Warning count	Number of tasks that are executed successfully but contain warnings. **NOTE** ● The task for which a warning is generated may be a failure or a success, which needs to be further determined according to the corresponding result list. ● The number of tasks that generate warnings is independent of the number of successful or failed tasks. The total number of tasks is the number of successful tasks plus the number of failed tasks.
Succeed bytes	Number of bytes that are successfully uploaded or downloaded.
max cost	Maximum duration of all tasks, in ms
min cost	Minimum duration of all tasks, in ms
average cost	Average duration of all tasks, in ms
average tps	The average number of tasks completed per second
Task id	Unique ID of an operation, which is used to search for the result list generated in a batch task

4.2.5 Resumable Upload

Scenarios

Uploading large files often fails due to poor network conditions or program breakdowns. Re-uploads not only waste resources, but may also fail again. To avoid such issues, you can use the resumable upload API, which works by dividing a file into multiple parts and then uploading them separately. This API helps save resources for re-uploads and improves efficiency by concurrent uploads.

Important Notes

- To upload objects, you must be the bucket owner or have the required permission (**obs:object:PutObject** in IAM or bPutObjectb in a bucket policy). For details, see **Introduction to OBS Access Control**, **IAM Custom Policies**, and **Configuring an Object Policy**.
- If you want to know the regions where OBS is available and the mapping between regions and endpoints, refer to **Regions and Endpoints**.
- The file uploaded by the resumable upload API must be larger than 100 KB in size.
- Before using SDK for resumable upload, you must enable the resumable unload option. Only in this way, the progress of the last upload can be read when you continue the upload process again.

Ways to Upload

You can use SDKs, OBS Browser+, or obsutil to upload objects.

Using SDKs

Java	**Python**	**C**	**Go**	**BrowserJS**	**.NET**	**Android**	**iOS**	**PHP**	**PHP**

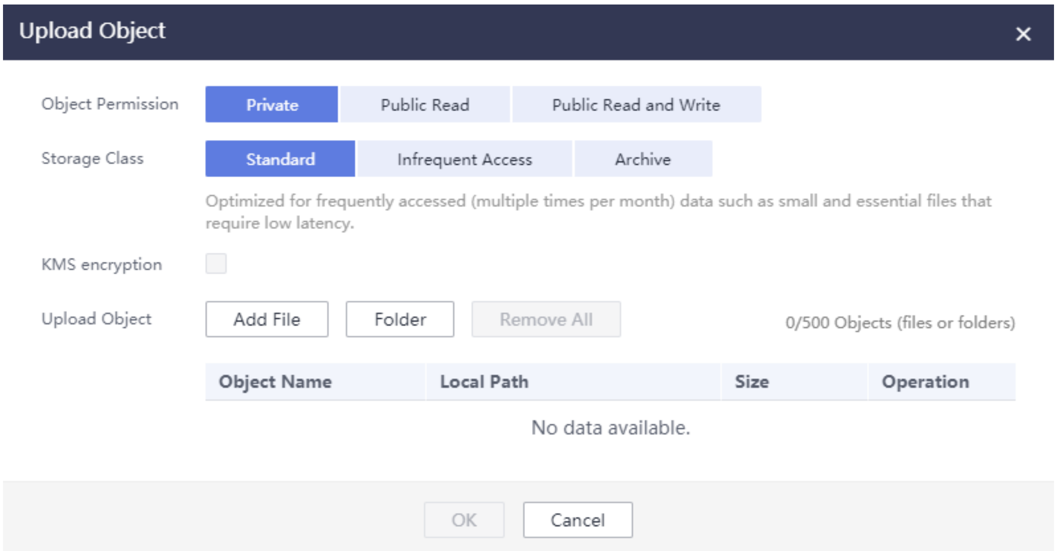
Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Click the bucket where you want to upload files or folders.

Step 3 Click **Upload** and then **Add File** or **Folder**, as shown in **Figure 4-11**.

Figure 4-11 Uploading a file or folder



Upload Object

Object Permission: Private | Public Read | Public Read and Write

Storage Class: Standard | Infrequent Access | Archive

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

KMS encryption □

Upload Object: Add File | Folder | Remove All — 0/500 Objects (files or folders)

Object Name	Local Path	Size	Operation

No data available.

For better experience when using the **Add File** function, you are advised to upload a maximum of 100 files at a time. If you need to upload more, place all the files in a folder and upload them by adding a folder.

NOTE

1. If message "Service Unavailable" is displayed when files are being uploaded, try again later.
2. If an access deny message is displayed when you are uploading a file or folder, possible causes are as follows:

- Access to the bucket is restricted by an ACL. For example, you do not have the write permission for the bucket.
- Access to the bucket is restricted by a bucket policy. For example, you do not have the write permission for the bucket, or write operations cannot be performed on the bucket during the current period.

  If such message is displayed, check ACL and policy settings of the bucket and resolve the problem accordingly.

⚠ CAUTION

You must have access to the file you want to upload, or the file upload will fail.

Step 4 In the displayed dialog box, select the file or folder you want to upload and click **Open**.

You can upload one folder or multiple files at a time. To upload multiple files, hold down **Ctrl** or **Shift** to select multiple files and batch upload them. You can also press **Ctrl+A** to select all files. The operations are consistent with those in Windows operating systems.

Step 5 Click **OK**.

----End

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
  - Uploading a file
    ```
    obsutil cp file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-u] [-vlength] [-vmd5] [-
    p=1] [-threshold=5248800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-
    cpd=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading a folder
    ```
    obsutil cp folder_url obs://bucket[/key] -r [-arcDir=xxx] [-dryRun] [-link] [-f] [-flat] [-u] [-
    vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-
    meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2]
    [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading multiple files/folders
    ```
    obsutil cp file1_url,folder1_url|filelist_url obs://bucket[/prefix] -msm=1 [-r] [-arcDir=xxx] [-
    dryRun] [-link] [-f] [-u] [-vlength] [-vmd5] [-flat] [-j=1] [-p=1] [-threshold=52428800] [-
    acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx][-
    timeRange=time1-time2] [-at] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-
    k=xxx] [-t=xxx]
    ```

    NOTE

    In this command, **/prefix** is the name prefix for uploading folders. For the execution examples, see **Upload**.
- In Linux or macOS
  - Uploading a file
    ```
    ./obsutil cp file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-u] [-vlength] [-vmd5] [-
    p=1] [-threshold=5248800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-
    cpd=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading a folder
    ```
    ./obsutil cp folder_url obs://bucket[/key] -r [-arcDir=xxx] [-dryRun] [-link] [-f] [-flat] [-u] [-
    vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-
    meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2]
    [-at] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Uploading multiple files/folders
    ```
    ./obsutil cp file1_url,folder1_url|filelist_url obs://bucket[/prefix] -msm=1 [-r] [-arcDir=xxx] [-
    dryRun] [-link] [-f] [-u] [-vlength] [-vmd5] [-flat] [-j=1] [-p=1] [-threshold=52428800] [-
    acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx][-
    timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-
    t=xxx]
    ```

    NOTE

    In this command, **/prefix** is the name prefix for uploading folders. For the execution examples, see **Upload**.

Examples

- Take the Windows OS as an example. Run the **obsutil cp d:\temp\test.txt obs://bucket-test/key** command to upload the **test.txt** file in the **temp** directory in the **D:** drive to bucket **bucket-test** and rename the file as **key**.
  ```
  obsutil cp d:\temp\test.txt obs://bucket-test/key

  Parallel:      5                   Jobs:          5
  Threshold:     50.00MB             PartSize:      auto
  VerifyLength:  false               VerifyMd5:     false
  CheckpointDir: C:\Users\Administrator\.obsutil_checkpoint

  [==================================================] 100.00% 1.68 MB/s 8.46MB/
  8.46MB 5s
  Upload successfully, 8.46MB, n/a, d:\temp\test.txt --> obs://bucket-test/key, cost [5], status [200],
  request id [0000016979E1D2B2860BB5181229C72C]
  ```

- Take the Windows OS as an example. Run the **obsutil cp d:\temp obs://bucket-test -f -r** command to recursively upload all files and subfolders in the **temp** directory in the **D:** drive to the **temp** folder in bucket **bucket-test**.

```
obsutil cp d:\temp obs://bucket-test -f -r

Parallel:      5                   Jobs:          5
Threshold:     50.00MB             PartSize:      auto
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: C:\Users\Administrator\.obsutil_checkpoint

Task id: 104786c8-27c2-48fc-bc6a-5886596fb0ed
OutputDir: C:\Users\Administrator\.obsutil_output

[======================================================] 100.00% tps:35.71 2.02 KB/s 7.20MB/7.20MB 0s
Succeed count is:   5         Failed count is:    0
Metrics [max cost:90 ms, min cost:45 ms, average cost:63.80 ms, average tps:35.71]

Task id: 104786c8-27c2-48fc-bc6a-5886596fb0ed
```

- For more examples, see **Upload**.

Parameter Description

Parameter	Optional or Mandatory	Description
file_url	Optional for uploading multiple files/folders Mandatory for uploading a file	Local file path **NOTE** ● Do not nest paths when uploading multiple files/folders. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time. ● If this parameter is configured when uploading multiple files/folders, **msm** must be set to **1**. In this case, use commas (,) to separate multiple file paths, for example, **file_url1,file_url2**. ● Files and folders can both be included when uploading multiple files/folders. For example, **file_url1,folder_url1,file_url2,folder_url2**.
folder_url	Optional for uploading multiple files/folders Mandatory for uploading a folder	Local folder path **NOTE** ● If **flat** is not configured when uploading a folder, the entire folder is uploaded. If **flat** is configured, all files in the folder are uploaded. ● Do not nest paths when uploading multiple files/folders. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time. ● If this parameter is configured when uploading multiple files/folders, **msm** must be set to **1**. In this case, use commas (,) to separate multiple folder paths, for example, **folder_url1,folder_url2**. ● Files and folders can both be included when uploading multiple files/folders. For example, **file_url1,folder_url1,file_url2,folder_url2**.

Parameter	Optional or Mandatory	Description
filelist_url	Optional for uploading multiple files/ folders	Indicates the path of the file that contains the list of files/folders to be uploaded. If this parameter is configured, **msm** must be set to **2**. **NOTE** - The list file is in common text file formats, such as TXT and CSV. Each line in the file indicates a file or folder to be uploaded. For example: file_url1 file_url2 folder_url1 folder_url2 - Do not nest paths in the list file. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time.
bucket	Mandatory	Bucket name
key	Optional	Indicates the object name or object name prefix specified when uploading a file, or the object name prefix specified when uploading a folder. The rules are as follows: - If this parameter is left blank when uploading a file, the file is uploaded to the root directory of the bucket and the object name is the file name. If the value ends with a slash (/), the value is used as the object name prefix when the file is uploaded, and the object name is the value plus the file name. If the value does not end with a slash (/), the file is uploaded with the value as the object name. - If this parameter is left blank when uploading a folder, the folder is uploaded to the root directory of the bucket. If the value ends with a slash (/), the value is used as the object name prefix of the folder to be uploaded. If the value does not end with a slash (/), the folder to be uploaded is prefixed with the value plus a slash (/). **NOTE** For details about how to use this parameter, see **Upload**.

Parameter	Optional or Mandatory	Description
fr	Optional for uploading a file (additional parameter)	Generates an operation result list when uploading a file.
flat	Optional for uploading a folder or multiple files/ folders (additional parameter)	Uploads all files in a folder but not the folder itself.
arcDir	Optional (additional parameter)	Path to which the uploaded files are archived
dryRun	Optional (additional parameter)	Conducts a dry run.
link	Optional (additional parameter)	Uploads the actual path of the symbolic-link file/folder **NOTICE** ● If this parameter is not specified and the file to be uploaded is a symbolic-link file whose target file does not exist, the exception message "The system cannot find the file specified" will be displayed in Windows OS, while the exception message "No such file or directory" will be displayed in macOS or Linux OS. ● Avoid the symbolic link loop of a folder, otherwise, the upload will exit due to panic. If you do not want the system to panic, set **panicForSymbolicLinkCircle** to **false** in the configuration file.
u	Optional (additional parameter)	Indicates incremental upload. If this parameter is set, each file can be uploaded only when it does not exist in the bucket, its size is different from the namesake one in the bucket, or it has the latest modification time. **CAUTION** When you compare each local file with data in the bucket, a billable HEAD request is generated. For details, see **Requests**.
vlength	Optional (additional parameter)	After the upload completes, check whether the sizes of the objects in the bucket are the same as those of the local files.

Parameter	Optional or Mandatory	Description
vmd5	Optional (additional parameter)	After the upload completes, check whether the MD5 values of the objects in the bucket are the same as those of the local files. **NOTE** ● If the size of the file or folder to be uploaded is too large, using this parameter will degrade the overall performance due to MD5 calculation. ● After the MD5 verification is successful, this parameter value is used for metadata **x-obs-meta-md5chksum**, for later MD5 verification during download or copy.
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart upload tasks when uploading a file. The default value is the value of **defaultParallels** in the configuration file.
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart upload, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the file or folder to be uploaded is smaller than the threshold, upload it directly. Otherwise, a multipart upload is required. ● If you upload a file or folder directly, no part record is generated, and resumable transmission is not supported. ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes).
acl	Optional (additional parameter)	Access control policies that can be specified when uploading files. Possible values are: ● private ● public-read ● public-read-write ● bucket-owner-full-control **NOTE** The preceding four values indicate private read and write, public read, public read and write, and bucket owner full control.

Parameter	Optional or Mandatory	Description
sc	Optional (additional parameter)	Indicates the storage classes of objects that can be specified when uploading files. Possible values are: ● **standard**: Standard storage class. It features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month) or data that is smaller than 1 MB. ● **warm**: Infrequent Access storage class. It is ideal for storing infrequently accessed (less than 12 times a year) data, but when needed, the access has to be fast. ● **cold**: Archive storage class. It provides secure, durable, and inexpensive storage for rarely-accessed (once a year) data. ● **deep-archive**: Deep Archive storage class (under limited beta testing). It is suitable for storing data that is barely (once every few years) accessed. This storage class costs less than the Archive storage class, but takes longer time (usually several hours) to restore data.
meta	Optional (additional parameter)	Indicates the standard and custom metadata that can be specified during file upload. This parameter should be configured in the following format: *key1***:***value1***#***key2***:***value2***#***key3***:***value3*. **NOTE** 1. The format example above indicates that the destination objects contain three groups of custom metadata: *key1:value1*, *key2:value2*, and *key3:value3*. 2. Standard metadata headers include **Content-Type**, **Content-Encoding**, **Cache-Control**, **Content-Disposition**, **Content-Language** and **Expires**.
ps	Optional (additional parameter)	Indicates the size of each part in a multipart upload task, in bytes. The value ranges from **100KB** to **5GB**. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes). ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source file size.

Parameter	Optional or Mandatory	Description
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart upload and saved to the **upload** subfolder. After the upload succeeds, its part record is deleted automatically. If the upload fails or is suspended, the system attempts to resume the task according to its part record when you perform the upload the next time.
r	Mandatory for uploading a folder (additional parameter) Optional for uploading multiple files/ folders	Indicates files and subfolders within the folder when uploading a folder recursively.
f	Optional for uploading a folder or multiple files/ folders (additional parameter)	Runs in force mode.
j	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the maximum number of concurrent tasks for uploading a folder. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
msm	Mandatory for uploading multiple files/ folders (additional parameter)	Enables the mode for uploading multiple files/folders. Possible values are **1** and **2**. **NOTE** ● If **msm** is set to **1**, the source URL indicates a list of file/folder names separated by commas. ● If **msm** is set to **2**, the source URL indicates a file containing a list of file/folder names. ● If the file or folder name already contains commas (,), do not set **msm** to **1**. ● If parameter **r** is not set, the folders in the list will not be uploaded.

Parameter	Optional or Mandatory	Description
exclude	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the file matching patterns that are excluded, for example: ***.txt**. **NOTE** - The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. - You can use **\*** to represent ***** and **\\?** to represent **?**. - If the name of the file to be uploaded matches the value of this parameter, the file is skipped. **NOTICE** - You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. - The matching pattern applies to the absolute file path (including the file name and file directory). - The matching pattern takes effect only for files in the folder. - Multiple **exclude** parameters can be specified, for example, **-exclude=*.xxx -exclude=*.xxx**.

Parameter	Optional or Mandatory	Description
include	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the file matching patterns that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use **\*** to represent ***** and **\\?** to represent **?**. ● Only after identifying that the name of the file to be uploaded does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is uploaded. If not, the file is skipped. **NOTE** Example of uploading files in a request with the **include** parameter contained: `./obsutil cp /localpath/ obs://test/ -include=/localpath/2022-12-09/* -f -r` This command uploads files that are under **localpath** and start with **/localpath/2022-12-09/** to bucket **test**. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute file path (including the file name and file directory). ● The matching pattern takes effect only for files in the folder. ● Multiple **include** parameters can be specified, for example, **-include=*.xxx -include=*.xxx**.
at	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates that only the files whose latest access time is within the value of **timeRange** are uploaded. **NOTE** ● This parameter must be used together with **timeRange**.
disableDirObject	Optional for uploading multiple folders (additional parameter)	Indicates the folders themselves are not uploaded as an object. Configuring this parameter can avoid uploading empty folders to a bucket. If a folder contains files, the files will be uploaded and the original path format is retained.

Parameter	Optional or Mandatory	Description
timeRange	Optional for uploading a folder or multiple files/ folders (additional parameter)	Indicates the time range matching pattern when uploading files. Only files whose latest modification time is within the configured time range are uploaded. This pattern has a lower priority than the file matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured file matching patterns. **NOTE** ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-***time2*, all files whose latest modification time is earlier than *time2* are matched. If it is set to *time1***-***, all files whose latest modification time is later than *time1* are matched. **NOTICE** Time in the matching pattern is the UTC time.
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on folders.

Parameter	Optional or Mandatory	Description
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** - The naming rule for result lists is as follows: **cp_{succeed | failed | warning}_report_***time***_TaskId.txt** - By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file. - If there are multiple folders and files and you need to confirm the detailed error information about a failed task, refer to the failure list **cp_failed_report_***time***_TaskId.txt** in the result list folder and the **log files** in the log path.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

Response

Field	Description
Parallel	Parameter **-p** in the request

Field	Description
Jobs	Parameter **-j** in the request
Threshold	Parameter **-threshold** in the request
PartSize	Parameter **-ps** in the request
Exclude	Parameter **-exclude** in the request
Include	Parameter **-include** in the request
TimeRange	Parameter **-timeRange** in the request
VerifyLength	Parameter **-vlength** in the request
VerifyMd5	Parameter **-vmd5** in the request
CheckpointDir	Parameter **-cpd** in the request
OutputDir	Parameter **-o** in the request
ArcDir	Parameter **-arcDir** in the request
Succeed count	Number of successful tasks
Failed count	Number of failed tasks
Skip count	Number of tasks that are skipped during incremental upload, download, or copy, and synchronous upload, download, or copy. **NOTE** Skipped tasks are recorded into successful tasks.
Warning count	Number of tasks that are executed successfully but contain warnings. **NOTE** ● The task for which a warning is generated may be a failure or a success, which needs to be further determined according to the corresponding result list. ● The number of tasks that generate warnings is independent of the number of successful or failed tasks. The total number of tasks is the number of successful tasks plus the number of failed tasks.
Succeed bytes	Number of bytes that are successfully uploaded or downloaded.
max cost	Maximum duration of all tasks, in ms
min cost	Minimum duration of all tasks, in ms
average cost	Average duration of all tasks, in ms
average tps	The average number of tasks completed per second
Task id	Unique ID of an operation, which is used to search for the result list generated in a batch task

4.2.6 Appendable Upload

Scenarios

After a streaming upload, browser-based upload, or multipart upload is complete, the uploaded object can be read but cannot be modified. If the object's content changes, you need to upload a new object with the same name to overwrite the original one.

There will be real-time streams when you upload logs, video surveillance data, live streaming data, and others. Using any of the foregoing uploads may cause problems such as complex software architecture, high server loads from frequent client requests, and high data latency. To solve such problems, you can use the appendable upload, which allows you to append data to the end of an object in a specific bucket. If there is no object with the same key value in the bucket, a new object will be created.

With appendable upload, newly generated video data can be appended to the same object in a timely manner. A client only needs to periodically obtain the length of the object and compare it with the length read last time. If there is new data that can be read, the client will read the newly uploaded data. In this way, the architecture can be simplified and its scalability can be enhanced.

Limits

- The last modification time of the object is updated each time an appendable upload is performed.
- If SSE-C encryption is used on the server side, you need to specify the request headers such as **x-obs-server-side-encryption** when you perform any appendable upload, which is the same as initiating multipart upload.
- For the server-side encryption (SSE-KMS), you only need to specify the request header such as **x-obs-server-side-encryption** when the file is uploaded for the first time and there is no object with the same name in the bucket.
- The length of each appendable upload cannot exceed the upper limit (5 GB) of the object length.
- Each appendable object can be appended for up to 10,000 times.
- If the object storage class is **COLD** (Archive) or **DEEP_ARCHIVE** (Deep Archive), appendable upload API cannot be called.
- If cross-region replication is configured for a bucket, appendable upload API cannot be called.
- Appendable upload is not available for parallel file systems.
- If you perform the PUT operation on an existing appendable object, the appendable object is overwritten by the newly uploaded object and the object type changes to normal. If you perform the PUT operation on a normal object, an error occurs.
- An appendable object will be changed to a normal object after being copied. An appendable object cannot be copied and saved as an appendable object.

Important Notes

- If the file you want to append data to does not exist, calling the AppendObject API will create an appendable file.
- If a bucket has WORM enabled, an append operation on this bucket will fail, with a 403 error returned.
- The ETag returns the hash value of the data to be uploaded, not the hash value of the entire object.
- Suppose there is a file you want to append data to:
  - If the file is appendable and the position where data will be appended matches the file length, the data can be directly appended to the end of the file. If the position does not match the file length, the PositionNotEqualToLength exception will be returned.
  - If the file is not appendable, for example, the file type is **Normal**, the ObjectNotAppendable exception will be returned.

Ways to Append

You can use SDKs or APIs to append objects.

Using APIs

Appending an Object

Using SDKs

Java	**Python**	**C**	**Go**	**BrowserJS**	**.NET**	**Android**	**iOS**	PHP: not supported	**Node.js**

4.2.7 Authorizing a Third Party to Upload Objects

Scenarios

If a third-party user does not have the upload permission, you can grant temporary permissions to them for accessing OBS and uploading objects within a validity period. For example, some companies have user management systems that manage app users and local users. These users do not have IAM user permissions, so you can grant temporary permissions to allow these users to temporarily access OBS.

Use either of the following methods:

- Temporary security credentials
- A temporary URL

Important Notes

Method	Description
Temporary security credentials	The validity period of temporary security credentials is from 15 minutes to 24 hours.
	When obtaining temporary security credentials, you can send the policy parameter to request for the least temporary permissions that can be granted to IAM users.
Temporary URL	Only the bucket owner can grant a temporary URL to others. If the owner changes the validity period of a URL, OBS obtains the authentication information again to generate a new URL.
	The **obs:PutObject** permission is required for uploading an object using a temporary URL.

Using Temporary Security Credentials to Grant Upload Permissions

You can assign temporary security credentials (including an AK, an SK, and a security token) to a third-party application or an IAM user, so that they can access OBS only for a specified period of time.

The temporary AK/SK and the security token must be used together to call an API for authentication and the **x-obs-security-token** field must be added to the request header.

For details, see **Accessing OBS Using Temporary Access Keys**.

Using a Temporary URL to Grant Upload Permissions

ObsClient allows you to create a URL with Query parameters that carry authentication information by specifying the security credentials, request method, and request parameters. You can share this URL with other users for them to make a temporary access. You need to specify the validity period of the temporary URL to restrict the allowed access duration.

Temporary URLs allow third-party users to upload objects without security credentials or authorization. OBS stores the objects uploaded by third-party users in a specified bucket.

For details, see **Accessing OBS Using a Temporary URL**.

4.2.8 Checking the Upload Progress

Scenarios

You can use OBS SDKs to obtain the upload progress of a specified object to estimate the remaining time.

SDKs Available for Progress Check

You can use the SDKs listed below to check your upload progress.

Java	**Python**	C: not supported	Go: not supported	**BrowserJS**	**.NET**	**Android**	iOS: not supported	PHP: not supported	Node.js: not supported

4.3 Downloading an Object

Scenarios

You can download objects from OBS to your local computer.

OBS supports batch download of objects. For details, see **Does OBS Support Batch Download?**

Limitations and Constraints

- Objects in the Archive or Deep Archive storage class can be downloaded only when they are in the **Restored** state.
- Batch download is not supported on OBS Console. To batch download files or folders, you can use OBS Browser+ or obsutil.
  - **Downloading Files or Folders Using OBS Browser+**
  - **Downloading Objects Using obsutil**

Ways to Download an Object

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to download an object.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 Select the file you want to download. Then, click **Download** or **More** > **Download As** on the right.

You can also select multiple files and choose **More** > **Download** above the file list.

NOTE

In the **Download As** dialog box, right-click the object and choose **Copy Link Address** from the shortcut menu to obtain the object's download address.

----End

Using the API

Downloading an Object

Using SDKs

Java	**Python**	**C**	**Go**	**BrowserJS**	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**

Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Go to the target bucket, select the file or folder you want to download, and click **Download**.

You can select multiple files or folders and batch download them.

NOTE

- To download Archive files, you must restore them first. For details, see **Restoring an Object**.
- Only files or folders that have been listed can be downloaded in batches. You can drag the scroll bar to the bottom to update the number of listed files or folders. 100 files or folders can be updated at a time.

Step 3 In the displayed dialog box, select a path for saving the file or folder and click **Select Folder**.

----End

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
  - Downloading a single object
    ```
    obsutil cp obs://bucket/key file_or_folder_url [-tempFileDir=xxx] [-dryRun] [-u] [-vlength] [-vmd5] [-p=1] [-threshold=52428800] [-versionId=xxx] [-ps=auto] [-cpd=xxx][-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Downloading objects in batches
    ```
    obsutil cp obs://bucket[/key] folder_url -r [-tempFileDir=xxx] [-dryRun] [-f] [-flat] [-u] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
- In Linux or macOS
  - Downloading a single object
    ```
    ./obsutil cp obs://bucket/key file_or_folder_url [-tempFileDir=xxx] [-dryRun] [-u] [-vlength] [-vmd5] [-p=1] [-threshold=52428800] [-versionId=xxx] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Downloading objects in batches
    ```
    ./obsutil cp obs://bucket[/key] folder_url -r [-tempFileDir=xxx] [-dryRun] [-f] [-flat] [-u] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-ps=auto] [-include=*.xxx] [-
    ```

```
exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-
i=xxx] [-k=xxx] [-t=xxx]
```

Examples

- Take the Windows OS as an example. Run the **obsutil cp obs://bucket-test/key d:\temp\test.txt** command to download a single object.

```
obsutil cp obs://bucket-test/key d:\temp\test.txt

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx

[========================================] 100.00% 4.86 KB/s 8.46MB/8.46MB 0s
Download successfully, 8.46MB, obs://bucket-test/key --> d:\temp\test.txt
```

- Take the Windows OS as an example. Run the **obsutil cp obs://bucket-test/temp d:\ -f -r** command to download objects in batches.

```
obsutil cp obs://bucket-test/temp d:\ -f -r

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx

Task id: 3066a4b0-4d21-4929-bb84-4829c32cbd0f
OutputDir: xxxx

[====================================================] 100.00% tps:17.86 155.59 KB/s
7.20MB/7.20MB 0s
Succeed count is:   6          Failed count is:    0
Metrics [max cost:153 ms, min cost:129 ms, average cost:92.00 ms, average tps:17.86]

Task id: 3066a4b0-4d21-4929-bb84-4829c32cbd0f
```

- For more examples, see **Download**.

Parameter Description

Parameter	Optional or Mandatory	Description
file_or_folder_url	Mandatory for downloading an object	Local file/folder path
folder_url	Mandatory for downloading objects in batches	Local folder path
bucket	Mandatory	Bucket name

Parameter	Optional or Mandatory	Description
key	Mandatory for downloading an object Optional for downloading objects in a batch	Indicates the name of the object to be downloaded, or the name prefix of the objects to be downloaded in batches. This parameter cannot be left blank when downloading an object. The saving and naming rules are as follows: ● If this parameter specifies a file or folder path that does not exist, the tool checks whether the value ends with a slash (/) or backslash (\). If yes, a folder is created based on the path, and the object is downloaded to this newly created directory. ● If this parameter specifies a file or folder path that does not exist and the value does not end with a slash (/) or backslash (\), the object is downloaded to your local PC with the value of **key** as the file name. ● If this parameter specifies an existing file, the object is downloaded to your local PC overwriting the existing file, with the value of **key** as the file name. ● If this parameter specifies an existing folder, the object is downloaded to the directory specified by **file_or_folder_url** with the object name as the file name. During batch download, the saving rules are as follows: ● If this parameter is left blank, all objects in the bucket are downloaded to the directory specified by **folder_url**. ● If this parameter is configured, objects whose name prefix is the configured value in the bucket are downloaded to the directory specified by **folder_url**. **NOTE** ● If this parameter is configured but the **flat** parameter is not configured when downloading objects in a batch, the name of the downloaded file contains the name prefix of the parent object. If **flat** is configured, then the name of the downloaded file does not contain the name prefix of the parent object. ● For details about how to use this parameter, see **Download**.

Parameter	Optional or Mandatory	Description
fr	Optional for downloading an object (additional parameter)	Generates an operation result list when downloading an object.
flat	Optional for downloading objects in batches (additional parameter)	The name prefix of the parent object is excluded when downloading objects in batches.
tempFileDir	Optional (additional parameter)	Indicates the directory for storing temporary files during multipart download. The default value is the value of **defaultTempFileDir** in the configuration file. **NOTE** ● If this parameter is left blank and the **defaultTempFileDir** parameter in the configuration file is also left blank, temporary files generated during multipart download are saved in the directory where to-be-downloaded files are located and end with the suffix of **.obs.temp**. ● Temporary files generated during multipart download are stored in this directory. Therefore, ensure that the user who executes obsutil has the write permission on the path. ● The available space of the partition where the path is located must be greater than the size of the objects to be downloaded.
dryRun	Optional (additional parameter)	Conducts a dry run.
u	Optional (additional parameter)	Indicates incremental download. If this parameter is set, each object can be downloaded only when it does not exist in the local path, its size is different from the namesake one in the local path, or it has the latest modification time.
vlength	Optional (additional parameter)	Checks whether the sizes of the local files are the same as those of the objects in the bucket after the download is complete.

Parameter	Optional or Mandatory	Description
vmd5	Optional (additional parameter)	Checks whether MD5 values of the local files are the same as those of the objects in the bucket after the download is complete. **NOTE** Objects in the bucket must contain metadata **x-obs-meta-md5chksum**, or MD5 verification will be skipped.
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart download tasks when downloading an object. The default value is the value of **defaultParallels** in the configuration file.
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart download, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the object to be downloaded is smaller than the threshold, download the object directly. If not, a multipart download is required. ● If you download an object directly, no part record is generated, and resumable transmission is not supported. ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes).
versionId	Optional for downloading an object (additional parameter)	Source object version ID that can be specified when downloading an object
ps	Optional (additional parameter)	Indicates the size of each part in a multipart download task, in bytes. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes). ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source object size.

Parameter	Optional or Mandatory	Description
fs	Optional (additional parameter)	Specifies whether the method of listing parallel file systems is applied. If you are listing parallel file systems, you are recommended to add this parameter. **CAUTION** ● With this method, the listing time required varies largely depending on the directory structures. ● After this parameter is enabled, **marker** and **limit** will be ignored. Then, the size of the bucket or parallel file system will be calculated. ● This parameter is only supported by obsutil 5.5.12 and later.
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart download and saved to the **down** subfolder. After the download succeeds, its part record is deleted automatically. If the download fails or is suspended, the system attempts to resume the task according to its part record when you perform the download the next time.
r	Mandatory for downloading objects in batches (additional parameter)	Copies objects in batches based on a specified object name prefix.
f	Optional for downloading objects in batches (additional parameter)	Runs in force mode.
j	Optional for downloading objects in batches (additional parameter)	Indicates the maximum number of concurrent tasks for downloading objects in a batch. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.

Parameter	Optional or Mandatory	Description
exclude	Optional for downloading objects in batches (additional parameter)	Indicates the matching patterns of source objects that are excluded, for example, ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use **\*** to represent ***** and **\\?** to represent **?**. ● If the name of the object to be downloaded matches the value of this parameter, the object is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/). ● Multiple **exclude** parameters can be specified, for example, **-exclude=*.xxx -exclude=*.xxx**.

Parameter	Optional or Mandatory	Description
include	Optional for downloading objects in batches (additional parameter)	Specifies what formats of objects can be downloaded. If this parameter is set to ***.jpg**, only objects in the **.jpg** format can be downloaded. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use **\*** to represent ***** and **\\?** to represent **?**. ● Only after identifying that the name of the file to be downloaded does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is downloaded. If not, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/). ● Multiple **include** parameters can be specified, for example, **-include=*.xxx -include=*.xxx**.

Parameter	Optional or Mandatory	Description
timeRange	Optional for downloading objects in batches (additional parameter)	Indicates the time range matching pattern when downloading objects. Only objects whose latest modification time is within the configured time range are downloaded. This pattern has a lower priority than the object matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured object matching patterns. **NOTE** ● The matching time range is represented in *time1***-***time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-***time2*, all files whose latest modification time is earlier than *time2* are matched. If it is set to *time1***-***, all files whose latest modification time is later than *time1* are matched. **NOTICE** ● Time in the matching pattern is the UTC time. ● This matching pattern applies only to objects whose names do not end with a slash (/).
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on objects whose names end with a slash (/).

Parameter	Optional or Mandatory	Description
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **cp_{succeed | failed | warning}_report_*time*_TaskId.txt** ● By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file. ● If there are multiple folders and files and you need to confirm the detailed error information about a failed task, refer to the failure list **cp_failed_report_*time*_TaskId.txt** in the result list folder and the **log files** in the log path.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

Response

Field	Description
Parallel	Parameter **-p** in the request
Jobs	Parameter **-j** in the request
Threshold	Parameter **-threshold** in the request
PartSize	Parameter **-ps** in the request
Exclude	Parameter **-exclude** in the request
Include	Parameter **-include** in the request
TimeRange	Parameter **-timeRange** in the request
VerifyLength	Parameter **-vlength** in the request
VerifyMd5	Parameter **-vmd5** in the request
CheckpointDir	Parameter **-cpd** in the request
OutputDir	Parameter **-o** in the request
ArcDir	Parameter **-arcDir** in the request
Succeed count	Number of successful tasks
Failed count	Number of failed tasks
Skip count	Number of tasks that are skipped during incremental upload, download, or copy, and synchronous upload, download, or copy. **NOTE** Skipped tasks are recorded into successful tasks.
Warning count	Number of tasks that are executed successfully but contain warnings. **NOTE** ● The task for which a warning is generated may be a failure or a success, which needs to be further determined according to the corresponding result list. ● The number of tasks that generate warnings is independent of the number of successful or failed tasks. The total number of tasks is the number of successful tasks plus the number of failed tasks.
Succeed bytes	Number of bytes that are successfully uploaded or downloaded.
max cost	Maximum duration of all tasks, in ms
min cost	Minimum duration of all tasks, in ms
average cost	Average duration of all tasks, in ms
average tps	The average number of tasks completed per second
Task id	Unique ID of an operation, which is used to search for the result list generated in a batch task

4.4 Limiting Download Speed By Single-Connection Bandwidth Throttling

Scenarios

To prevent a client from using a large amount of bandwidth to access objects, you can use single-connection bandwidth throttling to limit the download speed, allowing other applications to have enough bandwidth.

You can limit download speed by:

- **SDK**
- **Object URL**
- **Signed URL**
- **Header Field**

SDK

When you use an SDK to send a request, you can only limit the download speed by single-connection bandwidth throttling. For details, see **Single-Connection Bandwidth Throttling**.

Object URL

Item	Description
Object	Public read or public read/write object
Method	Add the bandwidth limit parameter in the **x-obs-traffic-limit=***value* format to the end of the URL of the shared object. The value can be from **819200** to **838860800**, in bit/s.
Scenarios	Object downloads
Example	https://bucketname.obs.cn-north-4.myhuaweicloud.com/object1?**x-obs-traffic-limit=819200** indicates that the maximum download speed of **object1** is 100 KB/s.

Signed URL

Item	Description
Object	Private object

Item	Description
Method	When you use an SDK to generate a signed URL, add the bandwidth limit parameter in the **x-obs-traffic-limit=***value* format to the signature for calculation. The value can be from **819200** to **838860800**, in bit/s.
Scenarios	Object downloads
Example	https://bucketname.obs.cn-north-4.myhuaweicloud.com/object1?**x-obs-traffic-limit=819200&Signature=signature**.

Header Field

Item	Description
Object	Private object
Method	Add the bandwidth limit parameter in the **x-obs-traffic-limit:***value* format to the header field. The value can be from **819200** to **838860800**, in bit/s.
Scenarios	Object downloads
Example	GET /object01 HTTP/1.1 User-Agent: curl/7.29.0 Host: examplebucket.obs.cn-north-4.myhuaweicloud.com Host: examplebucket.obs.*region*.myhuaweicloud.com Accept: */* Date: WED, 01 Jul 2015 04:24:33 GMT Authorization: OBS H4IPJX0TQTHTHEBQQCEC:NxtSMS0jaVxlLnxlO9awaMTn47s= **x-obs-traffic-limit:819200**

4.5 Previewing Objects

Scenarios

For security and compliance purposes, OBS prohibits the preview of objects in a bucket by using the default OBS domain name (**a bucket domain name or static website domain name**). When you use such a domain name to access objects (such as videos, images, and web pages) in a browser, no content will be displayed. Instead, the content is downloaded as an attachment.

To review objects in a bucket, you can use a user-defined domain name.

Solution 1: Use a User-Defined Domain Name

Solution 2: Use a User-Defined Domain Name and CDN

Solution 3: Use Private Bucket Retrieval on CDN

NOTE

If you have bound a user-defined domain name to a bucket and enabled CDN acceleration, but the objects are still directly downloaded without object preview, handle the problem by referring to **With CDN Acceleration Enabled, Why Are the Objects in My OBS Bucket Directly Downloaded When I Access Them?**

This restriction takes effect in different regions at the following two points in time:

January 1, 2022: CN North-Beijing1, CN North-Beijing4, CN East-Shanghai1, CN East-Shanghai2, and CN South-Guangzhou

March 25, 2022: CN-Hong Kong, AP-Bangkok, AP-Singapore, AF-Johannesburg, LA-Mexico City1, LA-Mexico City2, LA-Sao Paulo1, AP-Jakarta, and LA-Santiago

Solution 1: Use a User-Defined Domain Name

NOTE

Solution advantages and restrictions

- The bucket policy must be public read, so that all objects in the bucket can be accessed by anonymous users.
- As required by the China's Ministry of Industry and Information Technology (MIIT), you must complete the Internet Content Provider (ICP) filing, if the bucket to which your user-defined domain name is bound is in any of the following regions:

  CN North-Beijing1, CN North-Beijing4, CN East-Shanghai1, CN East-Shanghai2, and CN South-Guangzhou
- Only HTTP requests are supported.

Procedure

Step 1 Bind a user-defined domain name to the OBS bucket.

For details, see **Configuring a User-Defined Domain Name**. During the configuration, add a CNAME record for the OBS bucket domain name.

Step 2 Check whether objects can be previewed in a browser online.

Share an object link formatted in **http://***User-defined domain name***/***Object access path* with other users for them to preview the object.

If the object still cannot be previewed, check whether the value of the object metadata **ContentType** is supported by the browser.

For example, **ContentType** should be set to **video/mp4** for an MP4 video or set to **image/jpeg** for a JPG image. For details about all supported **ContentType** values, see **Object Metadata Content-Type**.

You can configure **ContentType** on OBS Console, or using APIs and SDKs.

Table 4-2 Modifying object metadata

Tool	Reference
OBS Console	**Configuring Object Metadata**
API	**Modifying Object Metadata**
SDK	Section about setting object attributes in the **SDK Reference**

----End

Solution 2: Use a User-Defined Domain Name and CDN

NOTE

Solution advantages and restrictions

- The bucket policy must be public read, so that all objects in the bucket can be accessed by anonymous users.
- As required by the China's Ministry of Industry and Information Technology (MIIT), you must complete the Internet Content Provider (ICP) filing, if the bucket to which your user-defined domain name is bound is in any of the following regions:

  CN North-Beijing1, CN North-Beijing4, CN East-Shanghai1, CN East-Shanghai2, and CN South-Guangzhou
- HTTPS requests are supported. To use HTTPS, you need to import a certificate to CDN. For details, see **HTTPS Certificates**.

Procedure

Step 1 On the CDN console, add the prepared user-defined domain name as an acceleration domain name.

For details, see **Adding a Domain Name**. Set **Origin Server Address** to **Domain name** and set **Domain name** to an OBS bucket domain name, with the HTTP port and HTTPS port set to **80** and **443** respectively. On the **Advanced Settings** tab, click **Edit** next to **HTTP Header**. Then, add response header **Content-Disposition** and set its value to **inline**.

Step 2 Bind the user-defined domain name to the OBS bucket.

For details, see **Binding a User-Defined Domain Name**. During the configuration, use the CNAME record set allocated by CDN.

Step 3 Check whether objects can be previewed in a browser online.

Share an object link formatted in **http://***User-defined domain name***/***Object access path* with other users for them to preview the object.

If the object still cannot be previewed, check whether the value of the object metadata **ContentType** is supported by the browser.

For example, **ContentType** should be set to **video/mp4** for an MP4 video or set to **image/jpeg** for a JPG image. For details about all supported **ContentType** values, see **Object Metadata Content-Type**.

You can configure **ContentType** on OBS Console, or using APIs and SDKs.

Step 4 (Optional) If the preview still fails, resolve the issue by referring to **Method 1** under **Solutions** and try again.

----End

Solution 3: Use Private Bucket Retrieval on CDN

NOTE

Solution advantages and restrictions

- The bucket policy can be public read or private.
- You only need to configure an acceleration domain name on CDN. No user-defined domain name of OBS is required. As required by the China's Ministry of Industry and Information Technology (MIIT), the used acceleration domain name must complete the Internet Content Provider (ICP) filing, if your bucket is in any of the following regions:

  CN North-Beijing1, CN North-Beijing4, CN East-Shanghai1, CN East-Shanghai2, and CN South-Guangzhou
- HTTPS requests are supported. To use HTTPS, you need to import a certificate to CDN. For details, see **HTTPS Certificates**.

Procedure

Step 1 On the CDN console, add the prepared user-defined domain name as an acceleration domain name.

For details, see **Adding a Domain Name**. Set **Origin Server Address** to **OBS bucket** and select the corresponding bucket domain name.

Step 2 Enable **Private Bucket Retrieval**.

For details, see **OBS Private Bucket Access**.

Step 3 Configure the CNAME record set on DNS.

For details, see the CNAME record configuration in **Binding a User-Defined Domain Name**. Use the CNAME record set allocated by CDN.

Step 4 Check whether objects can be previewed in a browser online.

Share an object link formatted in **http://***User-defined domain name***/***Object access path* with other users for them to preview the object.

If the object still cannot be previewed, check whether the value of the object metadata **ContentType** is supported by the browser.

For example, **ContentType** should be set to **video/mp4** for an MP4 video or set to **image/jpeg** for a JPG image. For details about all supported **ContentType** values, see **Object Metadata Content-Type**.

You can configure **ContentType** on OBS Console, or using APIs and SDKs.

----End

4.6 Object Management

4.6.1 Listing Objects

On OBS Console, when you go to the object list page of a bucket, objects are displayed by name by default. You can also sort objects by their size or last modification time.

If your bucket has versioning enabled, on the **Objects** page, you can choose whether to display historical versions of objects.

Constraints

- Listing objects by specifying a page number is not allowed.
- Objects cannot be listed by time when they were uploaded. You can search for objects by prefix only.
- The size and last modification time in the object list sort only objects on the current page.

Ways to List Objects

You can use OBS Console, APIs, SDKs, or obsutil to list objects.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 View the displayed objects. All objects in the bucket will be listed and each page has 50 objects displayed by default.

Step 4 (Optional) If versioning is enabled for a bucket, you can enable or disable **Historical Versions** above the object list to determine whether to display historical versions of objects.

- **Enabled**: Current and historical versions of objects (including deleted objects) are displayed.
- **Disabled**: Only current versions of objects are displayed.

----End

Using the API

Listing Objects in a Bucket

Using SDKs

Java	**Python**	**C**	**Go**	**BrowserJS**	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
```
obsutil ls obs://bucket[/prefix] [-s] [-d][-fs] [-v] [-du] [-marker=xxx] [-versionIdMarker=xxx] [-bf=xxx] [-limit=1] [-format=default] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```
- In Linux or macOS

```
./obsutil ls obs://bucket[/prefix] [-s][-fs] [-d] [-v] [-du] [-marker=xxx] [-versionIdMarker=xxx] [-
bf=xxx] [-limit=1] [-format=default] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

Examples

- Example 1: Take the Windows OS as an example. Run the **obsutil ls obs://bucket-test -limit=10** command to list objects in the bucket.

```
obsutil ls obs://bucket-test -limit=10

Folder list:
obs://bucket-test/api/

Object list:
key                                    LastModified              Size     StorageClass      ETag
obs://bucket-test/AUTHORS                        2018-11-16T02:15:49Z        33243     standard
"796393c1eaf502ef56a85c2ceb640aea"

obs://bucket-test/CONTRIBUTING.md               2018-11-16T02:15:49Z        1366
standard          "12d93325ba6131f852daecd18dd65edc"

obs://bucket-test/CONTRIBUTORS                  2018-11-16T02:15:49Z        45710
standard          "b486b5003e6215c9199e86ab3ccec9fa"

obs://bucket-test/LICENSE                      2018-11-16T02:15:49Z        1479      standard
"5d4950ecb7b26d2c5e4e7b4e0dd74707"

obs://bucket-test/PATENTS                      2018-11-16T02:15:49Z        1303      standard
"3a55d95595a6f9e37dee53826b4daff2"

obs://bucket-test/README.md                      2018-11-16T02:15:49Z        1399      standard
"97351fd7946b9ea021a31a86ba2a10ab"

obs://bucket-test/VERSION                      2018-11-16T02:15:49Z        7         standard
"43d93b553855b0e1fc67e31c28c07b65"

obs://bucket-test/api/README                    2018-11-16T02:15:49Z        521       standard
"4e9e63a87075df60cdf65c8ce9e92117"

obs://bucket-test/api/except.txt               2018-11-16T02:15:49Z        20194     standard
"8eb96de3f60447e2f09a7531c99fb3ee"

Next marker is: api/except.txt
Folder number is: 1
File number is: 9
```

- Example 2: Take the Windows OS as an example. Run the **obsutil ls obs://bucket-test2** command to list objects in the bucket.

```
obsutil ls obs://bucket-test2

Object list:
key                                    LastModified              Size     StorageClass      ETag
obs://bucket-test2/123                       2022-03-29T09:17:51Z        0B       standard
"d41d8cd98f00b204e9800998ecf8427e"

obs://bucket-test2/1_2-3.txt                  2022-03-29T09:17:51Z        0B       standard
"d41d8cd98f00b204e9800998ecf8427e"

obs://bucket-test2/1_2-3_33.txt                2022-03-29T09:17:51Z        200B      standard
"dcf204c11d791255adc63e61763c2426"

obs://bucket-test2/New text file.txt
                                     2022-03-29T09:17:51Z        0B       standard
"d41d8cd98f00b204e9800998ecf8427e"

Total size of bucket is: 200B
Folder number is: 0
File number is: 4
```

- Example 3: Take the Windows OS as an example. Run the **obsutil ls obs://bucket-test2/prefix** command to list objects whose name prefix is **prefix** in the bucket.

```
obsutil ls obs://bucket-test2/prefix

Object list:
key                                   LastModified              Size     StorageClass      ETag
obs://bucket-test2/prefix                   2022-03-29T09:17:51Z        0B       standard
"d41d8cd98f00b204e9800998ecf8427e"

obs://bucket-test2/prefix2                  2022-03-29T09:17:51Z        0B       standard
"d41d8cd98f00b204e9800998ecf8427e"

Total size of bucket is: 0B
Folder number is: 0
File number is: 2
```

- For more examples, see **Listing**.

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
prefix	Optional	Prefix of an object name for listing objects **NOTE** If this parameter is left blank, all objects in the bucket are listed.
s	Optional (additional parameter)	Displays simplified query result. **NOTE** In the simplified format, the returned result contains only the object name.
d	Optional (additional parameter)	Lists only objects and subdirectories in the current directory, instead of recursively listing all objects and subdirectories. In big data scenarios, parallel file systems usually have deep directory levels and each directory has a large number of files. In such case, you are advised to use this parameter to limit the scope to list. **NOTE** According to the naming conventions in OBS, a slash (/) is used as the directory separator.
v	Optional (additional parameter)	Lists versions of an object in a bucket. The result contains the latest version and historical versions (if any) of the object.
marker	Optional (additional parameter)	Object name to start with when listing objects in a bucket. All objects are listed in lexicographical order by object name. **NOTE** For details about how to use this parameter, see **Listing**.

Parameter	Optional or Mandatory	Description
versionIdMarker	Optional (additional parameter). It must be used together with the **v** and **marker** parameters.	Version ID to start with when listing versions of objects in a bucket. All versions and objects are listed in lexicographical order by object name and version ID. **NOTE** If the value of **versionIdMarker** is not a version ID specified by **marker**, **versionIdMarker** is invalid.
bf	Optional (additional parameter)	Display formats of bytes in the listing result. Possible values are: ● human-readable ● raw **NOTE** If this parameter is not configured, the display format of bytes in the result is determined by the **humanReadableFormat** parameter in the configuration file.
du	Optional (additional parameter)	Quickly returns the total size of listed objects, without displaying detailed object information. This parameter can be used together with other parameters. **NOTE** If there are too many objects listed, wait for a while. **CAUTION** ● This parameter is only supported by obsutil 5.4.6 and later.
fs	Optional (additional parameter)	Specifies whether the method of listing parallel file systems is applied. If you are listing parallel file systems, you are recommended to add this parameter. **CAUTION** ● With this method, the listing time required varies largely depending on the directory structures. ● After this parameter is enabled, **marker** and **limit** will be ignored. Then, the buckets or parallel file systems (including directories) will be calculated. ● This parameter is only supported by obsutil 5.5.12 and later.

Parameter	Optional or Mandatory	Description
limit	Optional (additional parameter)	Maximum number of objects that can be listed. If the value is less than or equal to 0, all objects are listed. If it is left blank, 1,000 objects are listed by default. **NOTE** If there are a large number of objects in a bucket, you are advised to set this parameter to limit the number of objects to be listed each time. If not all objects are listed, **marker** and **versionIdMarker** of the next request will be returned in the result, which you can use to list the remaining objects.
format	Optional (additional parameter)	Prints a listing result in the user-defined format. Currently, the value can only be **default**, indicating that the listing result is displayed in one row.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

Response

Field	Description
Key	Object name
LastModified	Time when the last modification was made to the object
Size	Object size

Field	Description
StorageClass	Storage class of an object ● **standard**: Standard storage class. It features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month) or data that is smaller than 1 MB. ● **warm**: Infrequent Access storage class. It is ideal for storing infrequently accessed (less than 12 times a year) data, but when needed, the access has to be fast. ● **cold**: Archive storage class. It provides storage for rarely-accessed (once a year) data.
ETag	ETag of an object, which is a Base64-encoded 128-bit MD5 digest. ETag is the unique identifier of the object content. It can be used to determine whether the object content is changed. For example, if the ETag value is **A** when an object is uploaded, but this value has changed to **B** when the object is downloaded, it indicates that the object content has been changed. The ETag reflects changes to the object content, rather than the object metadata. An uploaded object or copied object has a unique ETag.

4.6.2 Copying Objects

Scenarios

You can copy objects in OBS. An object of up to 5 GB can be copied by each operation. To copy an object larger than 5 GB, use the multipart upload API. Specifically, this copy operation allows you to:

- Create a copy for an object.
- Rename an object by creating a copy for it and deleting the source object.
- Edit object metadata.

NOTE

Each object has metadata, which is a set of name-value pairs. There is system-defined metadata and user-defined metadata. Some system-defined metadata can also be controlled by users. When you upload an object, you can set its metadata.

- After you upload the object, you can modify its metadata using an API. For details, see **Modifying Object Metadata**.
- You can also create an object copy and set the metadata. In the copy operation, set the target object to be the same as the source object.

Both user-controlled metadata and user-defined metadata are copied. OBS automatically resets the system-controlled metadata. In the copy request, you do not need to set such metadata. For example, when you copy an object, OBS resets the creation date of the copied object.

If you want to modify metadata, even only one piece of metadata, that can be configured by users (defined by users or the system), specify all the metadata that can be configured by users on the source object in the request.

Prerequisites

You have the read permission for the source object and the read and write permissions for the destination bucket.

Important Notes

- Objects cannot be copied across regions. For example, objects in a bucket in the CN North-Beijing1 region cannot be copied to a bucket in the CN North-Beijing4 region. For details about how to automatically copy objects across regions, see **Cross-Region Replication**.
- An appendable object cannot be copied.
- You can copy an object of up to 5 GB in a single operation. To copy an object larger than 5 GB, use the multipart upload API.
- If you want to copy a large number of Deep Archive objects, you can set the storage class of the copied objects to Standard and then transition them to the Deep Archive storage class through **lifecycle rules** to lower the costs on PUT requests.
- By default, an original object will be overwritten by its copy with the same name. After the copy is successful, only the new object can be downloaded. The original object will be deleted. To prevent data loss:
  - Before copying an object, ensure that there is no object with the same name as the object copy or the object with the same name is useless.
  - Enable versioning.

    If versioning is enabled, deleted or overwritten objects will be retained as historical versions. You can restore a historical version at any time.
- OBS allows you to copy your service data stored in OBS to a specified region, but Huawei Cloud does not detect the stored data. You need to ensure the legal compliance of your use of OBS on your own. If cross-border transfer is required, ensure that your use complies with relevant laws and regulations.
- You cannot determine whether a copy request is executed successfully only using **status_code** in the HTTP header. If 200 in **status_code** is returned, the server has received the request and starts to process the request. The body in the response shows whether the request is executed successfully. The request

is executed successfully only when the body contains ETag; otherwise, the request fails to be executed.

Ways to Copy Objects

You can use APIs, SDKs, OBS Browser+, or obsutil to copy objects.

Using the API

Copying Objects

Using SDKs

Java	**Python**	**C**	**Go**	**BrowserJS**	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**

Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Go to the target bucket. Right-click the file or folder and choose **Copy** from the shortcut menu.

Step 3 Right-click the path for saving the file or folder and choose **Paste** from the shortcut menu.

NOTE

- Files or folders in external buckets cannot be copied.
- The restore status of Archive objects cannot be copied.
- The source path and target path must be different.

----End

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
  - Copying a single object
    ```
    obsutil cp obs://srcbucket/key obs://dstbucket/[dest] [-dryRun][-u] [-crr] [-vlength] [-vmd5] [-p=1] [-threshold=52428800] [-versionId=xxx] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Copying objects in batches
    ```
    obsutil cp obs://srcbucket[/key] obs://dstbucket[/dest] -r [-dryRun][-f] [-flat] [-u] [-crr] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
- In Linux or macOS
  - Copying a single object
    ```
    ./obsutil cp obs://srcbucket/key obs://dstbucket/[dest] [-dryRun] [-u] [-crr] [-vlength] [-vmd5] [-p=1] [-threshold=52428800] [-versionId=xxx] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```

– Copying objects in batches

```
./obsutil cp obs://srcbucket[/key] obs://dstbucket[/dest] -r [-dryRun] [-f] [-flat] [-u] [-crr] [-
vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-
meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2]
[-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

NOTE

- The source path and destination path cannot be the same.
- The source path and destination path cannot be partly overlapped either. If the source path overlaps with the prefix of the destination path, recursive replication applies. If the destination path overlaps with the prefix of the source path, the replication may overwrite objects in the source path.

Examples

- Take the Windows OS as an example. Run the **obsutil cp obs://bucket-test/key obs://bucket-test2** command to copy a single object.

```
obsutil cp obs://bucket-test/key obs://bucket-test2

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx

[====================================================] 100.00% 6/s 0s
Copy successfully, 19B, obs://bucket-test/key --> obs://bucket-test2/key
ext.txt
```

- Take the Windows OS as an example. Run the **obsutil cp obs://bucket-test/temp/ obs://bucket-test2 -f -r** command to copy objects in batches.

```
obsutil cp obs://bucket-test/temp/ obs://bucket-test2 -r -f

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx
OutputDir: xxxx


[============================================================] 100.00% 10/s 0s
Succeed count is:   5          Failed count is:    0
Metrics [max cost:298 ms, min cost:192 ms, average cost:238.00 ms, average tps:9.71]
Task id is: 0476929d-9d23-4dc5-b2f8-0a0493f027c5
```

- For more examples, see **Copy**.

Parameter Description

Parameter	Optional or Mandatory	Description
srcbucket	Mandatory	Source bucket name
dstbucket	Mandatory	Destination bucket name
dest	Optional	Indicates the destination object name when copying an object, or the name prefix of destination objects when copying objects in batches.

Parameter	Optional or Mandatory	Description
key	Mandatory for copying an object. Optional for copying objects in batches.	Indicates the source object name when copying an object, or the name prefix of source objects when copying objects in batches. The rules are as follows: ● This parameter cannot be left blank when copying an object. If **dest** is left blank, the source object is copied to the root directory of the destination bucket. If the value of **dest** ends with a slash (/), the destination object name is the value of **dest** plus the source object name. Otherwise, the destination object name is the value of **dest**. ● If this parameter is left blank when copying objects in batches, all objects in the source bucket are copied. If not, objects whose name prefix is the set value in the source bucket are copied. The rules for confirming the name of the destination object are as follows: – If the value of **dest** ends with a slash (/), the destination object name is the value of **dest** plus the source object name. – If the value of **dest** does not end with a slash (/), the destination object name is *the value of **dest**/source object name*. **NOTE** ● If this parameter is configured but the **flat** parameter is not when copying objects in batches, the name of the source object contains the name prefix of the parent object. If **flat** is configured, then the name of the source object does not contain the name prefix of the parent object. ● For details about how to use this parameter, see **Copy**.
fr	Optional for copying an object (additional parameter)	Generates an operation result list when copying an object.
flat	Optional for copying objects in batches (additional parameter)	The name prefix of the parent object is excluded when copying objects in batches.

Parameter	Optional or Mandatory	Description
dryRun	Optional (additional parameter)	Conducts a dry run.
crr	Optional (additional parameter)	Enables the client-side cross-region replication function. In this mode, data is directly copied to the destination bucket from the source bucket through data stream. The buckets can by any two OBS buckets. **NOTE** ● If this parameter is configured, the configuration of client-side cross-region replication must be updated in the configuration file. For details, see **Updating a Configuration File**. ● The configurations of the source bucket and destination bucket are respectively **akCrr/skCrr/tokenCrr/endpointCrr** and **ak/sk/token/endpoint** in the configuration file. **NOTICE** When cross-region replication is enabled, the upload/download bandwidth, CPU, and memory resources of the host where commands are executed will be occupied, which may deteriorate the host performance.
vlength	Optional (additional parameter)	Verifies whether the object size in the destination bucket is the same as that in the source bucket after the copy task completes. **NOTE** This parameter must be used together with **crr**.
vmd5	Optional (additional parameter)	Verifies whether the MD5 value of the destination bucket is the same as that of the source bucket after the copy task completes. **NOTE** ● This parameter must be used together with **crr**. ● Objects in the source bucket must contain metadata **x-obs-meta-md5chksum**, or MD5 verification will be skipped. After the MD5 verification is successful, this parameter value is used for metadata **x-obs-meta-md5chksum** of the destination object, for later MD5 verification during download or copy.
u	Optional (additional parameter)	Indicates incremental copy. If this parameter is set, each object can be copied only when it does not exist in the destination bucket, its size is different from the namesake one in the destination bucket, or it has the latest modification time.

Parameter	Optional or Mandatory	Description
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart copy tasks when copying an object. The default value is the value of **defaultParallels** in the configuration file.
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart copy, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the object to be copied is smaller than the threshold, copy the object directly. If not, a multipart copy is required. ● If you copy an object directly, no part record is generated, and resumable transmission is not supported. ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes).
versionId	Optional for copying an object (additional parameter)	Source object version ID that can be specified when copying an object
acl	Optional (additional parameter)	Access control policies for destination objects that can be specified when copying objects. Possible values are: ● private ● public-read ● public-read-write ● bucket-owner-full-control **NOTE** The preceding four values indicate private read and write, public read, public read and write, and bucket owner full control.

Parameter	Optional or Mandatory	Description
sc	Optional (additional parameter)	Storage classes of the destination objects that can be specified when copying objects. Possible values are: ● **standard**: Standard storage class. It features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month) or data that is smaller than 1 MB. ● **warm**: Infrequent Access storage class. It is ideal for storing infrequently accessed (less than 12 times a year) data, but when needed, the access has to be fast. ● **cold**: Archive storage class. It provides secure, durable, and inexpensive storage for rarely-accessed (once a year) data. ● **deep-archive**: Deep Archive storage class (under limited beta testing). It is suitable for storing data that is barely (once every few years) accessed. This storage class costs less than the Archive storage class, but takes longer time (usually several hours) to restore data.
meta	Optional (additional parameter)	Standard or custom metadata that can be specified for destination objects in object replication. This parameter should be configured in the following format: *key1*:*value1*#*key2*:*value2*#*key3*:*value3*. **NOTE** 1. The format example above indicates that the destination objects contain three groups of custom metadata: *key1:value1*, *key2:value2*, and *key3:value3*. 2. Standard metadata headers include Content-Type, Content-Encoding, Cache-Control, Content-Disposition, Content-Language and Expires.
fs	Optional (additional parameter)	Specifies whether the method of listing parallel file systems is applied. If you are listing parallel file systems, you are recommended to add this parameter. **CAUTION** ● With this method, the listing time required varies largely depending on the directory structures. ● After this parameter is enabled, **marker** and **limit** will be ignored. Then, the buckets or parallel file systems (including directories) will be calculated. ● This parameter is only supported by obsutil 5.5.12 and later.

Parameter	Optional or Mandatory	Description
ps	Optional (additional parameter)	Indicates the size of each part in a multipart copy task, in bytes. The value ranges from **100KB** to **5GB**. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes). ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source object size.
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart copy and saved to the **copy** subfolder. After the copy succeeds, its part record is deleted automatically. If the copy fails or is suspended, the system attempts to resume the task according to its part record when you perform the copy the next time.
r	Mandatory for copying objects in batches (additional parameter)	Copies objects in batches based on a specified name prefix of objects in the source bucket.
f	Optional for copying objects in batches (additional parameter)	Runs in force mode.
j	Optional for copying objects in batches (additional parameter)	Indicates the maximum number of concurrent tasks for copying objects in batches. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.

Parameter	Optional or Mandatory	Description
exclude	Optional for copying objects in batches (additional parameter)	Indicates the matching patterns of source objects that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use **\*** to represent ***** and **\\?** to represent **?**. ● If the name of the object to be copied matches the value of this parameter, the object is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/). ● Multiple **exclude** parameters can be specified, for example, **-exclude=*.xxx -exclude=*.xxx**.

Parameter	Optional or Mandatory	Description
include	Optional for copying objects in batches (additional parameter)	Indicates the matching patterns of source objects that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use ***** to represent ***** and **\?** to represent **?**. ● Only after identifying that the name of the file to be copied does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is copied. If not, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/). ● Multiple **include** parameters can be specified, for example, **-include=*.xxx -include=*.xxx**.

Parameter	Optional or Mandatory	Description
timeRange	Optional for copying objects in batches (additional parameter)	Indicates the time range matching pattern when copying objects. Only objects whose latest modification time is within the configured time range are copied. This pattern has a lower priority than the object matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured object matching patterns. **NOTE** ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-***time2*, all files whose latest modification time is earlier than *time2* are matched. If it is set to *time1***-***, all files whose latest modification time is later than *time1* are matched. **NOTICE** ● Time in the matching pattern is the UTC time. ● This matching pattern applies only to objects whose names do not end with a slash (/).
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on objects whose names end with a slash (/).
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **cp_{succeed | failed | warning}_report_***time***_TaskId.txt** ● By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file. ● If there are multiple folders and files and you need to confirm the detailed error information about a failed task, refer to the failure list **cp_failed_report_***time***_TaskId.txt** in the result list folder and the **log files** in the log path.

Parameter	Optional or Mandatory	Description
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

Response

Field	Description
Parallel	Parameter **-p** in the request
Jobs	Parameter **-j** in the request
Threshold	Parameter **-threshold** in the request
PartSize	Parameter **-ps** in the request
Exclude	Parameter **-exclude** in the request
Include	Parameter **-include** in the request
TimeRange	Parameter **-timeRange** in the request
VerifyLength	Parameter **-vlength** in the request
VerifyMd5	Parameter **-vmd5** in the request
CheckpointDir	Parameter **-cpd** in the request
OutputDir	Parameter **-o** in the request
ArcDir	Parameter **-arcDir** in the request
Succeed count	Number of successful tasks
Failed count	Number of failed tasks

Field	Description
Skip count	Number of tasks that are skipped during incremental upload, download, or copy, and synchronous upload, download, or copy. **NOTE** Skipped tasks are recorded into successful tasks.
Warning count	Number of tasks that are executed successfully but contain warnings. **NOTE** ● The task for which a warning is generated may be a failure or a success, which needs to be further determined according to the corresponding result list. ● The number of tasks that generate warnings is independent of the number of successful or failed tasks. The total number of tasks is the number of successful tasks plus the number of failed tasks.
Succeed bytes	Number of bytes that are successfully uploaded or downloaded.
max cost	Maximum duration of all tasks, in ms
min cost	Minimum duration of all tasks, in ms
average cost	Average duration of all tasks, in ms
average tps	The average number of tasks completed per second
Task id	Unique ID of an operation, which is used to search for the result list generated in a batch task

4.6.3 Configuring and Viewing Object Metadata

Scenarios

Object metadata is a set of name-value pairs that describe an object and is used for object management. OBS has two types of metadata: **system-defined object metadata** and **user-defined object metadata**.

You can configure metadata for an object when uploading it or after the object is uploaded.

Metadata Introduction

System-defined Object Metadata

There are two types of system-defined metadata: system-controlled and user-controlled.

- System-controlled metadata: Metadata like **Last-Modified** is system controlled and cannot be modified.

- User-controlled metadata: Metadata like **ContentLanguage** (configured for an object) is user controlled. You can make an API call to modify such metadata. **Table 4-3** lists the metadata items that you can control.

NOTE

OBS uses KMS keys to encrypt objects. The checksum, together with the specified algorithm, are stored as part of the object's metadata. If server-side encryption is requested for the object, the checksum is stored in encrypted form. For more information about server-side encryption, see **Server-Side Encryption**.

Table 4-3 OBS metadata

Name	Description
ContentDisposition	Provides a default file name for the requested object. When the object is being downloaded or accessed, the object named after the default file name is opened in the browser or the object download window is displayed. Suppose that you configure the **ContentDisposition** metadata with the value of **attachment;filename="testfile.xls"** for an object. When you access this object by using a link, a download window is displayed and the object name is changed to **testfile.xls**. For details, see the definition about **ContentDisposition** in HTTP.
ContentLanguage	States the language or languages intended for the audience. For details, see the definition about **ContentLanguage** in HTTP.
WebsiteRedirectLocation	Redirects an object to another object or an external URL. The redirection is implemented using static website hosting. For example, you can redirect an object by doing as follows: 1. Configure the **WebsiteRedirectLocation** metadata for object **testobject.html** in the root directory of bucket **testbucket** and set the metadata value to **http://www.example.com**. NOTE OBS only supports redirection for objects in a bucket's root directory. Redirection for objects in the bucket's folders is not supported. 2. Configure static website hosting for bucket **testbucket**, and set object **testobject.html** as the default homepage. 3. Access object **testobject.html** through the hosted URL provided on the static website hosting page. The access will be redirected to **http://www.example.com**.

Name	Description
ContentEncoding	Specifies the content encoding format when an object is downloaded. The options are as follows: ● Standard: **compress**, **deflate**, **exi**, **identity**, **gzip**, and **pack200-gzip** ● Others: **br**, **bzip2**, **lzma**, **peerdist**, **sdch**, **xpress**, and **xz**
CacheControl	Specifies the caching behavior for a web page when an object is downloaded. ● Cacheability: **public**, **private**, **no-cache**, or **only-if-cached** ● Expiration time: **max-age=<*seconds*>**, **s-maxage=<*seconds*>**, **max-stale[=<*seconds*>]**, **min-fresh=<*seconds*>**, **stale-while-revalidate=<*seconds*>**, or **stale-if-error=<*seconds*>** ● Revalidation and reloading: **must-revalidate**, **proxy-revalidate**, or **immutable** ● Others: **no-store** or **no-transform**
Expires	Specifies the cache expiration time (GMT).
ContentType	Specifies the object type. For details, see **Content-Type**.

User-defined Object Metadata

You can add user-defined metadata whose name starts with **x-obs-meta-** for easy object management. When you retrieve or query an object's metadata, its user-defined metadata will be returned in the response. User-defined metadata is limited to 8 KB in size. The size of user-defined metadata is the sum of the number of bytes in the UTF-8 encoding of each key and value.

User-defined metadata keys are case insensitive, but OBS stores them in lowercase. The user-defined metadata values are case sensitive.

Example:

```
PUT /key HTTP/1.1
Host: bucket01.obs.myhuaweicloud.com
x-obs-meta-Test1: Test Meta1

HEAD /Key HTTP/1.1
Host: bucket01.obs.myhuaweicloud.com
x-obs-meta-test1: Test Meta1
```

Both user-defined metadata keys and their values must conform to US-ASCII. If non-ASCII or unrecognizable characters are necessary, they must be encoded or decoded in URL or Base64 on the client side. The server side does not perform any decoding.

Constraints

- In a versioning-enabled bucket, you can only configure metadata for the current version of an object, but not for noncurrent versions.
- Metadata cannot be configured for Archive or Deep Archive objects.
- The user-defined metadata is limited to 8 KB in size.

Configuring Object Metadata

You can use OBS Console, APIs, SDKs, or obsutil to configure object metadata.

Using OBS Console

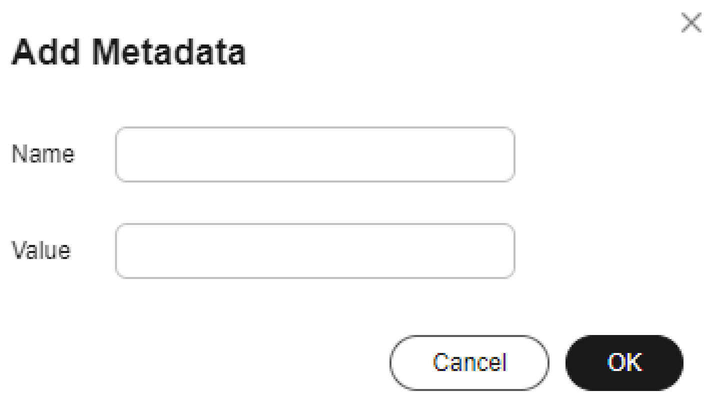
Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 Click the object to be operated, and then click the **Metadata** tab.

Step 4 Click **Add**. Then, configure metadata.

Figure 4-12 Adding metadata



Add Metadata

Name

Value

Step 5 Click **OK**.

----End

Using APIs

Modifying Object Metadata

Using SDKs

Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
  - Setting properties of a single object
    ```
    obsutil chattri obs://bucket/key [-meta=aaa:bbb#ccc:ddd] [-sc=xxx] [-acl=xxx] [-aclXml=xxx] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Setting properties of objects in batches
    ```
    obsutil chattri obs://bucket[/key] -r [-f] [-v] [-meta=aaa:bbb#ccc:ddd] [-sc=xxx] [-acl=xxx] [-aclXml=xxx] [-o=xxx] [-j=1] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
- In Linux or macOS
  - Setting properties of a single object
    ```
    ./obsutil chattri obs://bucket/key [-meta=aaa:bbb#ccc:ddd] [-sc=xxx] [-acl=xxx] [-aclXml=xxx] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Setting properties of objects in batches
    ```
    ./obsutil chattri obs://bucket[/key] -r [-f] [-v] [-meta=aaa:bbb#ccc:ddd] [-sc=xxx] [-acl=xxx] [-aclXml=xxx] [-o=xxx] [-j=1] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```

Examples

- Take the Windows OS as an example, run the **obsutil chattri obs://bucket-test/key -acl=public-read** command to set the access permission to an object to public read.
  ```
  obsutil chattri obs://bucket-test/key -acl=public-read

  Set the acl of object [key] in the bucket [bucket-test] to [public-read] successfully, request id
  [04050000016836DDFA73B2B5320E2651]
  ```
- Take the Windows OS as an example, run the **obsutil chattri obs://bucket-test -r -f -acl=public-read** command to set the access permission to all objects in the bucket to public read.
  ```
  obsutil chattri obs://bucket-test -r -f -acl=public-read

  [------------------------------------------------] 100.00% tps:155.15 5/5 233ms
  Succeed count is:   5         Failed count is:    0
  Metrics [max cost:177 ms, min cost:53 ms, average cost:102.40 ms, average tps:20.41]
  Task id is: 9d7f73ff-f747-4fdd-9b2a-815ba2dc3b07
  ```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
key	Mandatory when setting properties of an object. Optional when setting properties of objects in batches.	Indicates the name of the object whose properties are to be set, or the name prefix of objects whose properties are to be set in batches. **NOTE** If this parameter is left blank during batch operation, properties of all objects in the bucket are set.

Parameter	Optional or Mandatory	Description
meta	Optional (additional parameter)	Standard or custom metadata that can be specified for destination objects in object replication. This parameter should be configured in the following format: *key1*:*value1*#*key2*:*value2*#*key3*:*value3*. **NOTE** 1. This parameter takes effect only when it is used together with **direct**. 2. This value indicates that the destination objects contain three groups of custom metadata: *key1:value1*, *key2:value2*, and *key3:value3*. 3. Standard metadata headers include Content-Type, Content-Encoding, Cache-Control, Content-Disposition, Content-Language and Expires.
direct	Optional (additional parameter)	Metadata operation indicator. The value can be **REPLACE_NEW** or **REPLACE**. **REPLACE_NEW**: The existing metadata value is replaced with a new one, the metadata lacking a value is assigned one, and the metadata not specified keeps unchanged. **REPLACE**: The metadata is replaced with the header included in the current request and the metadata not specified is deleted. **NOTE** This parameter takes effect only when it is used together with **meta**.

Parameter	Optional or Mandatory	Description
sc	Optional (additional parameter)	Storage classes of objects. Possible values are: ● **standard**: Standard storage class. It features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month) or data that is smaller than 1 MB. ● **warm**: Infrequent Access storage class. It is ideal for storing infrequently accessed (less than 12 times a year) data, but when needed, the access has to be fast. ● **cold**: Archive storage class. It provides secure, durable, and inexpensive storage for rarely-accessed (once a year) data. ● **deep-archive**: Deep Archive storage class (under limited beta testing). It is suitable for storing data that is barely (once every few years) accessed. This storage class costs less than the Archive storage class, but takes longer time (usually several hours) to restore data. **NOTE** For an object whose storage class is **cold**, restore the object first and then specify its storage class. To restore an object, see **Restoring Objects from the Archive Storage**.
acl	Optional (additional parameter)	Access control policies that can be specified for objects. Possible values are: ● private ● public-read ● public-read-write ● bucket-owner-full-control **NOTE** The preceding four values indicate private read and write, public read, public read and write, and bucket owner full control.

Parameter	Optional or Mandatory	Description

aclXml	Optional (additional parameter)	Access control policy of the bucket, in XML format.
		`<AccessControlPolicy>`
		`    <Owner>`
		`        <ID>`*ownerid*`</ID>`
		`    </Owner>`
		`    <AccessControlList>`
		`        <Grant>`
		`            <Grantee>`
		`                <ID>`*userid*`</ID>`
		`            </Grantee>`
		`            <Permission>`*[WRITE|WRITE_ACP|READ|READ_ACP|FULL_CONTROL]*`</Permission>`
		`        </Grant>`
		`        <Grant>`
		`            <Grantee>`
		`                <Canned>Everyone</Canned>`
		`            </Grantee>`
		`            <Permission>`*[WRITE|WRITE_ACP|READ|READ_ACP|FULL_CONTROL]*`</Permission>`
		`        </Grant>`
		`    </AccessControlList>`
		`</AccessControlPolicy>`
		NOTE
		● **Owner**: Optional. Specify the object owner's ID.
		● In **AccessControlList**, the **Grant** field contains the authorized users. **Grantee** specifies the IDs of authorized users. **Canned** specifies the authorized user group (currently, only **Everyone** is supported).
		● The following permissions can be granted: WRITE (write), WRITE_ACP (write ACL), READ (read), READ_ACP (read ACL), and FULL_CONTROL (full control).
		NOTICE
		Because angle brackets (<) and (>) are unavoidably included in the parameter value, you must use quotation marks to enclose them for escaping when running the command. Use single quotation marks for Linux or macOS and quotation marks for Windows.

fs	Optional (additional parameter)	Specifies whether the method of listing parallel file systems is applied. If you are listing parallel file systems, you are recommended to add this parameter. **CAUTION** ● With this method, the listing time required varies largely depending on the directory structures. ● After this parameter is enabled, **marker** and **limit** will be ignored. Then, the buckets or parallel file systems (including directories) will be calculated. ● This parameter is only supported by obsutil 5.5.12 and later.

Parameter	Optional or Mandatory	Description
versionId	Optional when setting properties of an object (additional parameter)	Version ID of the object whose properties are to be set
fr	Optional when setting properties of an object (additional parameter)	Generates an operation result list when setting properties of an object.
f	Optional when setting properties of objects in batches (additional parameter)	Runs in force mode.
r	Mandatory when setting properties of objects in batches (additional parameter)	Sets properties of objects in batches based on a specified object name prefix.
v	Optional when setting properties of objects in batches (additional parameter)	Sets properties of versions of objects in batches based on a specified object name prefix.

Parameter	Optional or Mandatory	Description
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (including success and failure files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **chattri_{succeed | failed}_report_***time***_TaskId.txt** By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file. ● If there are multiple folders and files and you need to confirm the detailed error information about a failed task, refer to the failure list **chattri_failed_report_***time***_TaskId.txt** in the result list folder and the **log files** in the log path.
j	Optional when setting properties of objects in batches (additional parameter)	Indicates the maximum number of concurrent tasks for setting object properties in batches. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

NOTE

Only one from **acl**, **sc**, or **aclXml** can be set for each command.

Response

Field	Description
Parallel	Parameter **-p** in the request
Jobs	Parameter **-j** in the request
Threshold	Parameter **-threshold** in the request
PartSize	Parameter **-ps** in the request
Exclude	Parameter **-exclude** in the request
Include	Parameter **-include** in the request
TimeRange	Parameter **-timeRange** in the request
VerifyLength	Parameter **-vlength** in the request
VerifyMd5	Parameter **-vmd5** in the request
CheckpointDir	Parameter **-cpd** in the request
OutputDir	Parameter **-o** in the request
ArcDir	Parameter **-arcDir** in the request
Succeed count	Number of successful tasks
Failed count	Number of failed tasks
Skip count	Number of tasks that are skipped during incremental upload, download, or copy, and synchronous upload, download, or copy. **NOTE** Skipped tasks are recorded into successful tasks.
Warning count	Number of tasks that are executed successfully but contain warnings. **NOTE** ● The task for which a warning is generated may be a failure or a success, which needs to be further determined according to the corresponding result list. ● The number of tasks that generate warnings is independent of the number of successful or failed tasks. The total number of tasks is the number of successful tasks plus the number of failed tasks.
Succeed bytes	Number of bytes that are successfully uploaded or downloaded.
max cost	Maximum duration of all tasks, in ms
min cost	Minimum duration of all tasks, in ms

Field	Description
average cost	Average duration of all tasks, in ms
average tps	The average number of tasks completed per second
Task id	Unique ID of an operation, which is used to search for the result list generated in a batch task

Viewing Object Metadata

You can use OBS Console, APIs, SDKs, or obsutil to view object metadata.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 Click the object of which you want to view the metadata. On the displayed details page, view the object metadata on the **Metadata** tab page.

----End

Using APIs

Querying Object Metadata

Using SDKs

Java	**Python**	**C**	**Go**	**BrowserJS**	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
```
obsutil stat obs://bucket/key [-acl][-bf=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```
- In Linux or macOS
```
./obsutil stat obs://bucket/key [-acl][-bf=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

Examples

- Take the Windows OS as an example. Run the **obsutil stat obs://bucket-test/key** command to query the basic properties of an object.
```
obsutil stat obs://bucket-test/key

Start at 2024-09-25 04:48:10.1147483 +0000 UTC

Key:
  obs://bucket-test/key
LastModified:
  2018-11-16T02:15:49Z
Size:
  7
StorageClass:
  standard
ETag:
  43d93b553855b0e1fc67e31c28c07b65
ContentType:
  text/plain
Type:
  file
```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
key	Mandatory	Object name
acl	Optional	Queries the access control policies of the object at the same time.
bf	Optional (additional parameter)	Display format of the object size (in bytes). Possible values are: ● human-readable ● raw **NOTE** If this parameter is not configured, the display format of the object size (in bytes) is determined by the **humanReadableFormat** parameter in the configuration file.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

Response

Field	Description
Key	Object name
LastModified	Latest modification time of the object
Size	Object size, in bytes
StorageClass	Storage class of the object
MD5	Real MD5 of the object **NOTE** You can query this value only after running the **cp** command and configuring the **-vmd5** parameter.
ETag	ETag value of an object calculated on the server
ContentType	Content-Type of the object
Type	Object type
Metadata	Customized metadata of the object
Acl	Access control policy of the object

Object Metadata Content-Type

Content-Type

When you access an object through a web browser, the system specifies an application to open the object according to the value of **Content-Type**. You can modify the **Content-Type** of an object based on its file name extension.

NOTE

If you upload an object by calling an API, specify the value of **Content-Type**.

Table 4-4 Common Content-Type values

File Name Extension	Content-Type	File Name Extension	Content-Type
.* (binary stream, unknown file type)	application/octet-stream	.7z	application/x-7z-compressed
.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

File Name Extension	Content-Type	File Name Extension	Content-Type
.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
.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

File Name Extension	Content-Type	File Name Extension	Content-Type
.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
.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

File Name Extension	Content-Type	File Name Extension	Content-Type
.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	.m1v	video/x-mpeg
.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

File Name Extension	Content-Type	File Name Extension	Content-Type
.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	.mxp	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
.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

File Name Extension	Content-Type	File Name Extension	Content-Type
.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-realtext3d
.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
.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

File Name Extension	Content-Type	File Name Extension	Content-Type
.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-realtext
.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
.sty	application/x-sty	.svg	text/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

File Name Extension	Content-Type	File Name Extension	Content-Type
.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-vsd	.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
.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

File Name Extension	Content-Type	File Name Extension	Content-Type
.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

4.6.4 Sharing an Object

Scenarios

You can share temporary URLs of your objects with others for them to access your objects stored in OBS.

Background Information

You can share files with other users temporarily or permanently.

- Temporary sharing:

Share a file to other users through the temporary URL of the file. File sharing is temporary. All sharing URLs are only valid for a limited period of time.

A temporary URL consists of the access domain name and the temporary authentication information of a file. Example:

```
https://bucketname.obs.ap-southeast-1.myhuaweicloud.com:443/image.png?
AccessKeyId=xxx&Expires=xxx&x-obs-security-token=xxx&Signature=xxx
```

The temporary authentication information contains the **AccessKeyId**, **Expires**, **x-obs-security-token**, and **Signature** parameters. **AccessKeyId**, **x-obs-security-token**, and **Signature** are used for authentication. The **Expires** parameter specifies the validity period of the authentication.

For details about the temporary authentication method and parameters, see **Authentication of Signature in a URL** in the *Object Storage Service API Reference*.

- Permanent sharing:

  If you set the permission for an object to allow anonymous users to read it, anonymous users can access the object through the URL that you shared. For details about how to configure permissions, see **Granting Public Read Permissions on Objects to Anonymous Users**.

  The method of using a browser to access objects varies depending on the object type. You can directly open **.txt** and **.html** files using a browser. However, when you open **.exe** and **.dat** files using a browser, the files are automatically downloaded to your local computer.

  For details about how to obtain the object access URL, see **How Do I Obtain the Access Path to an Object?**

  If access to an object using the URL fails, fix the problem by referring to **Why Can't I Access an Object Through Its URL?**

Limitations and Constraints

- An authorization code is not required for temporarily sharing a file. Temporarily sharing a folder requires an authorization code when the method of sharing by access code is used. For details, see **Sharing a Folder**.
- The following table describes the URL validity period of the files that are temporarily shared in different ways.

Way	URL Validity Period
OBS Console	The URL is valid for 1 minute to 18 hours. After an object is shared, the system will generate a URL that contains the temporary authentication information, valid for five minutes since its generation by default. Each time you change the validity period of a URL, OBS obtains the authentication information again to generate a new URL for sharing. The new URL becomes valid at the moment the validity period is changed. If you need a longer validity period, use OBS Browser+ that allows a validity period of up to one year to share the object. If you want a shared object to be permanently valid, **use a bucket policy to grant anonymous users the read-only permission for the object**.

Way	URL Validity Period
SDKs	Configure parameter **Expires** to specify when a temporary authorization expires. The temporary authorization expires in 20 years in maximum.
API	Configure parameter **Expires** to specify when a temporary authorization expires. The temporary authorization expires in 20 years in maximum.
OBS Browser+	When you log in to OBS Browser+ using an account and password, a shared URL can be valid for 24 hours at most. The default validity period is 10 hours. If a longer validity period is required, use the permanent AK/SK for login.
obsutil	Use the additional parameter **e** to specify when an object download URL expires. The minimum value is 60s and the default one is 300s. There is no upper limit on the expiration time.

- Only buckets of version 3.0 support object sharing. For details about how to query the bucket version, see **Checking OBS Version (OBS 2.0 or OBS 3.0)**.
- Archive objects can be shared only after they have been restored.
- Deep Archive objects can be shared only after they have been restored.
- Object sharing is available in all regions except CN Southwest-Guiyang1.

Ways to Share an Object

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to share an object.

Using OBS Console

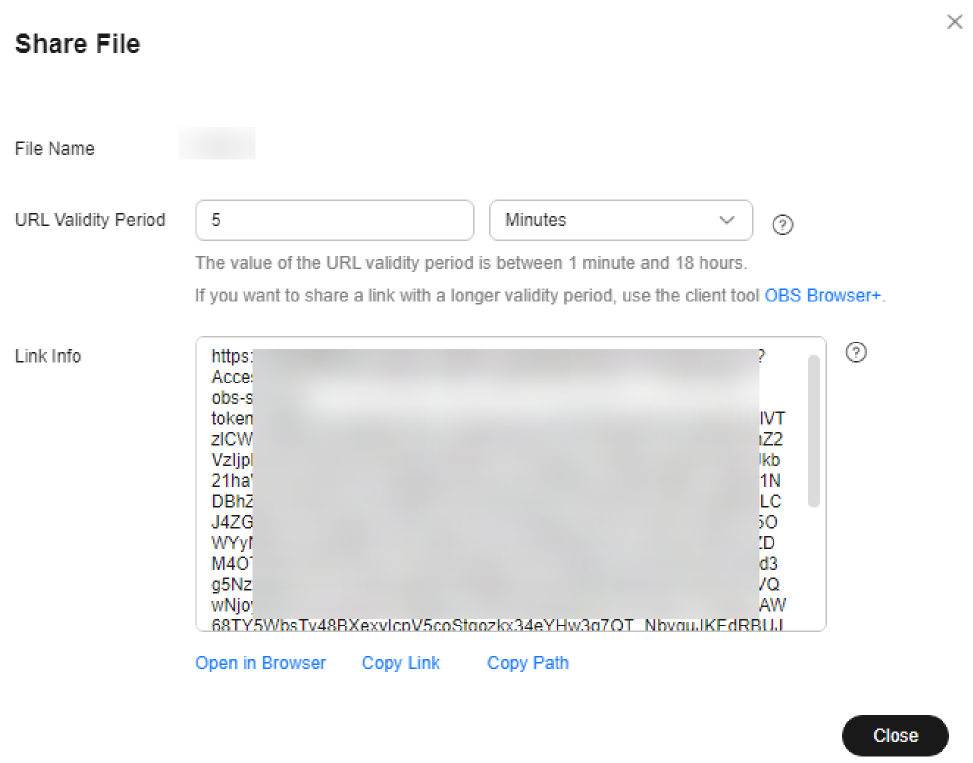
Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 Locate the file to be shared and click **Share** in the **Operation** column.

Once the **Share File** dialog box is opened, the URL is effective and valid for five minutes by default. If you change the validity period, the authentication information in the URL changes accordingly, and the URL's new validity period starts upon the change.

Figure 4-13 Sharing a file



Share File

File Name [redacted]

URL Validity Period 5 Minutes

The value of the URL validity period is between 1 minute and 18 hours.

If you want to share a link with a longer validity period, use the client tool OBS Browser+.

Link Info [redacted]

Step 4 Operate the URL as follows:

- Click **Open URL** to preview the file on a new page or directly download it to your default download path.
- Click **Copy Link** to share the link to others for them to access this file using a browser.
- Click **Copy Path** to share the file path to users who have access to the bucket. The users then can search for the file by pasting the shared path to the search box of the bucket.

NOTE

Within the URL validity period, anyone who has the URL can access the file.

----End

Using the API

Authentication of Signature in a URL

Using SDKs

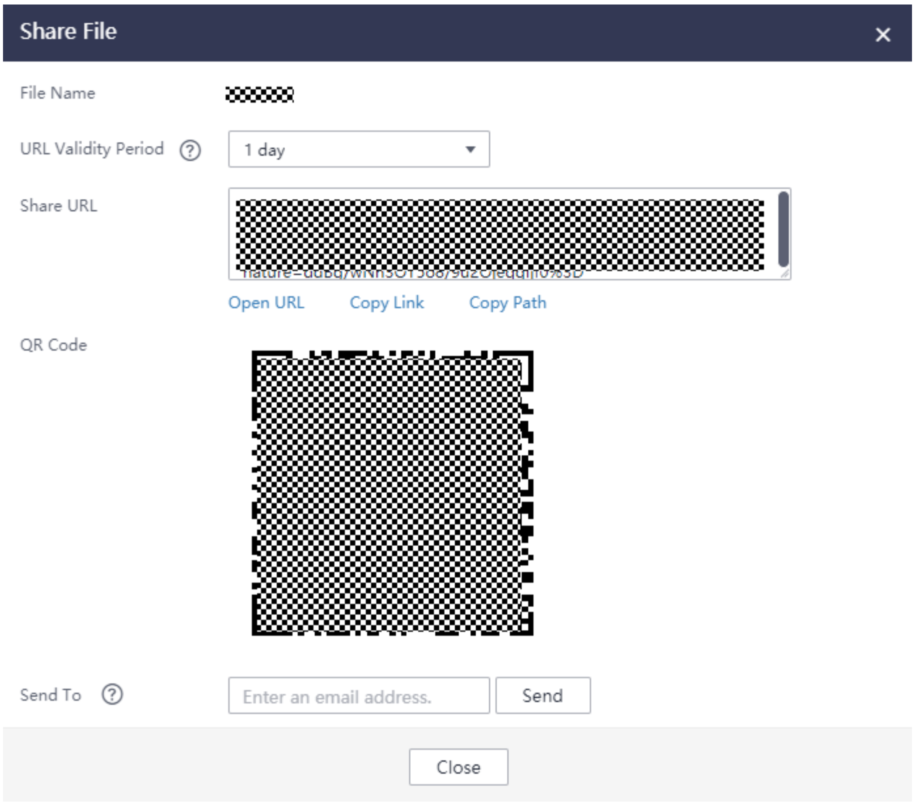
Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Locate the object you want to share and click next to the object. The window shown in **Figure 4-14** or **Figure 4-15** is displayed.

Figure 4-14 Sharing a file



Share File

File Name [redacted]

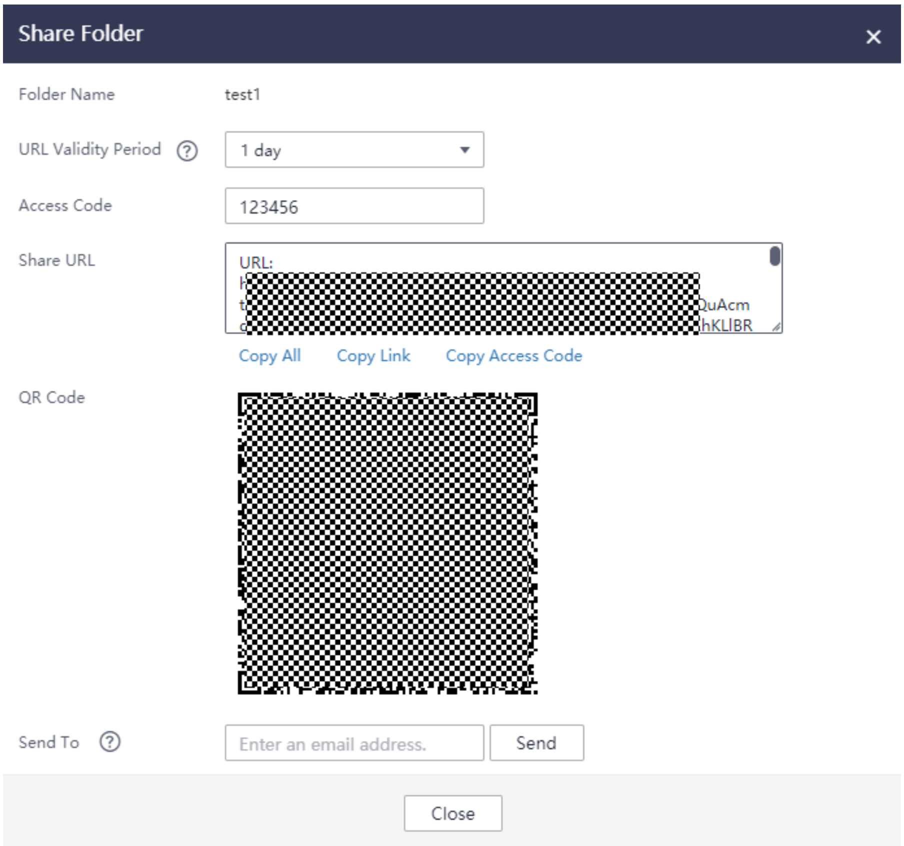
URL Validity Period 1 day

Share URL [redacted]

QR Code [redacted]

Send To Enter an email address.

Figure 4-15 Sharing a folder



Share Folder

Folder Name: test1

URL Validity Period: 1 day

Access Code: 123456

Share URL: URL: [redacted]

QR Code: [redacted]

Send To: Enter an email address.

Step 3 Click **Copy Link** to copy the object link to a browser and open it. Then you can access the object.

NOTE

To allow anonymous users to access Archive objects using URLs, ensure that these objects are in the **Restored** state.

----End

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
  - Generating the download link of a single object

    ```
    obsutil sign obs://bucket/key [-e=300] [-config=xxx] [-endpoint=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```

  - Generating the download links of objects in batches by object name prefix

    ```
    obsutil sign obs://bucket[/key] -r [-e=300] [-timeRange=time1-time2] [-include=*.xxx] [-exclude=*.xxx] [-o=xxx] [-config=xxx] [-endpoint=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```

- In Linux or macOS
  - Generating the download link of a single object

    ```
    ./obsutil sign obs://bucket/key [-e=300] [-config=xxx] [-endpoint=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```

  - Generating the download links of objects in batches by object name prefix

```
./obsutil sign obs://bucket[/key] -r [-e=300] [-timeRange=time1-time2] [-include=*.xxx] [-exclude=*.xxx] [-o=xxx] [-config=xxx] [-endpoint=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

Examples

- In Windows, run **obsutil sign obs://bucket-test/test.txt** to generate a single object download link:

```
obsutil sign obs://bucket-test/test.txt

Download url of [obs://bucket-test/test.txt] is:
  http://your-endpoint/bucket-test/test.txt?AccessKeyId=xxxx&Expires=1552548758&Signature=xxxx
```

- In Windows, run **obsutil sign obs://bucket-test/test/ -r** to generate object download links in batches:

```
obsutil sign obs://bucket-test/test/ -r

Generate download urls for objects .

Generate the download url(s) for the objects in the bucket [bucket-test] finished
Task id: af4dc692-6a88-4541-8156-ff1a889d2288
```

NOTE

- If there are a large number of objects, obsutil saves the object download links to a specific file. The file name is associated with the task ID. For instance, the task ID in the example above is **af4dc692-6a88-4541-8156-ff1a889d2288**, so the file name should be **sign_succeed_report_{*timestamp*}_af4dc692-6a88-4541-8156-ff1a889d2288.txt**.
- By default, the file is stored under folder **.obsutil_output** in your user directory. You can also use parameter **-o** to specify a new folder.

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
key	Optional	Object name used for generating the download link of a single object, or object name prefix used for generating download links of objects in batches
e	Optional (additional parameter)	Validity period of the generated download links of objects, in seconds. Minimum value: 60s. Default value: 300s
r	Mandatory when generating download links of objects in batches (additional parameter)	Generates the download links of objects in batches by a specified object name prefix.

Parameter	Optional or Mandatory	Description
exclude	Optional when generating download links of objects in batches (additional parameter)	Indicates the matching patterns of objects that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use **\*** to represent ***** and **\\?** to represent **?**. ● If the name of the object to be downloaded matches the value of this parameter, the object is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/). ● Multiple **exclude** parameters can be specified, for example, **-exclude=*.xxx -exclude=*.xxx**.

Parameter	Optional or Mandatory	Description
include	Optional when generating download links of objects in batches (additional parameter)	Indicates the matching patterns of objects that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use **\*** to represent ***** and **\\?** to represent **?**. ● Only after identifying that the name of the file to be downloaded does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is downloaded. If not, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/). ● Multiple **include** parameters can be specified, for example, **-include=*.xxx -include=*.xxx**.

Parameter	Optional or Mandatory	Description
timeRange	Optional (additional parameter)	Indicates the time range matching pattern when generating download links of objects. Only the download links of objects whose latest modification time is within the configured time range are generated. This pattern has a lower priority than the object matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured object matching patterns. **NOTE** ● Time in the matching pattern is the UTC time. ● This matching pattern applies only to objects whose names do not end with a slash (/). ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-***time2*, all files whose latest modification time is earlier than *time2* are matched. If it is set to *time1***-***, all files whose latest modification time is later than *time1* are matched.
o	Optional when generating download links of objects in batches (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success and failure files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **sign_{succeed | failed}_report_***time***_TaskId.txt** By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
endpoint	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.

Parameter	Optional or Mandatory	Description
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

4.6.5 Sharing Objects with Anonymous Users Using URLs

You can grant anonymous users the read permission for an object so they can access the object using the shared object URL.

Prerequisites

Anonymous users have the read permission for the object.

For details about permission granting, see **Granting All Accounts the Read Permission for Certain Objects**.

NOTE

Encrypted objects cannot be shared.

Procedure

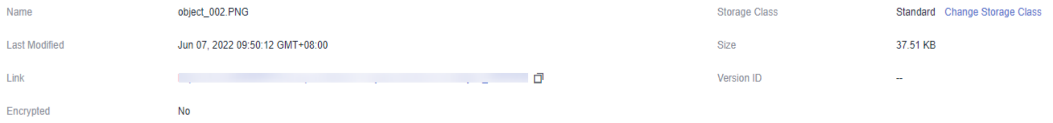
Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 Click the object to be shared. On the top part of the page, you can find the object's sharing link in the **Link** area.

Anonymous users can access the object by clicking this link. An object link (URL) is in the format of **https://***Bucket name.Domain name/Directory level/Object name*. If the object is stored in the root directory of the bucket, its URL does not contain any directory level. To learn more about domain names, see **OBS Domain Names**.

Figure 4-16 Object link



Name	object_002.PNG	Storage Class	Standard Change Storage Class
Last Modified	Jun 07, 2022 09:50:12 GMT+08:00	Size	37.51 KB
Link	[redacted]	Version ID	--
Encrypted	No		

NOTE

- To allow anonymous users to access objects in Archive or Deep Archive storage using URLs, ensure that these objects are in the **Restored** state.

----End

4.6.6 Restoring Archive or Deep Archive Objects

Scenarios

You must restore an object in the Archive or Deep Archive storage class before you can download it or access it with a URL.

To learn the costs involved in data restore, see **Product Pricing Details**.

NOTE

Within the validity period of a restored object, you can restore the object again. The validity period is then extended because it will start again when the latest restore is complete.

If a restored object is restored again, its expiration time should be later than the time set for the previous restore. Assume that an object is restored on January 1 and will expire 30 days later (on January 30). If the object is restored again on January 10 and is made to be expired earlier than January 30 (less than 20 days later), this restore action is considered invalid.

Prerequisites

You have the obs:object:RestoreObject permission. For details, see **Granting Permissions to an IAM User Under the Current Account**.

Important Notes

- If an Archive or a Deep Archive object is being restored, its restore speed cannot be changed and also the restore task cannot be suspended or deleted.
- After an object is restored, an object copy in the Standard storage class will be generated. This way, there is an Archive or a Deep Archive object and a Standard object copy in the bucket at the same time. After an Archive or Deep Archive object is restored, the object status displays **Restored**, and the generated object copy in Standard storage class is not displayed in the bucket.
- During the restore validity period, you will be billed for the space taken up by both the object and its copy. The copy will be automatically deleted once the restore expires.
- If versioning is enabled and there are different versions of an object in different storage classes, the current version of the object will be restored. To restore a specific object version, you can specify **versionId**.
- If you restore an object that is being restored, error 409 will be reported. Wait for the current restore to finish before attempting to restore again.
- If the RestoreObject API is called for the first time, 202 will be returned, indicating that the server has received the request. If the RestoreObject API is called again after a successful object restore, 200 OK will be returned.

Object Restore Option and Time Required

OBS provides the following restore options.

Storage Class	Restore Option	Time Required
Archive	Expedited	1 to 5 minutes
Archive	Standard	3 to 5 hours
Deep Archive	Expedited	3 to 5 hours
Deep Archive	Standard	5 to 12 hours

NOTE

To restore a large number of objects from the Deep Archive storage class, you are advised to use the standard restore. The restore time spent depends on the object size.

Ways to Restore Objects

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to restore Archive or Deep Archive objects.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

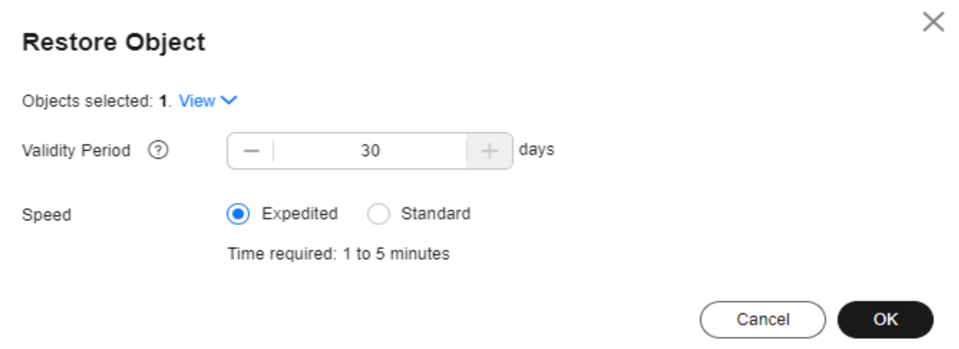
Step 3 Select the file you want to restore, and click **Restore** on the right.

You can select multiple files and choose **More** > **Restore** above the file list to batch restore them.

NOTE

Objects that are being restored cannot be added for batch restore.

Figure 4-17 Restore Object



Restore Object

Objects selected: 1. View

Validity Period: 30 days

Speed: ● Expedited ○ Standard

Time required: 1 to 5 minutes

Step 4 Configure the validity period and speed of the restore.

Table 4-5 Parameters for restoring objects

Parameter	Description
Validity Period	The number of days that the object will remain in the **Restored** state. It starts once the object is restored. The value is an integer from 1 to 30 (days). The default value is **30**. For example, if you set **Validity Period** to **20**, 20 days after the object is successfully restored, its status will change from **Restored** to **Unrestored**.
Speed	The time required for restoring an object. ● **Expedited**: Archive objects can be restored within 1 to 5 minutes, and Deep Archive objects can be restored within 3 to 5 hours. ● **Standard**: Archive objects can be restored within 3 to 5 hours, and Deep Archive objects can be restored within 5 to 12 hours.

Step 5 Click **OK**.

NOTE

The system checks the file restore status at UTC 00:00 every day. The system starts counting down the expiration time from the time when the latest check is complete.

----End

Using APIs

Restoring Archive or Deep Archive Objects

Using SDKs

Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

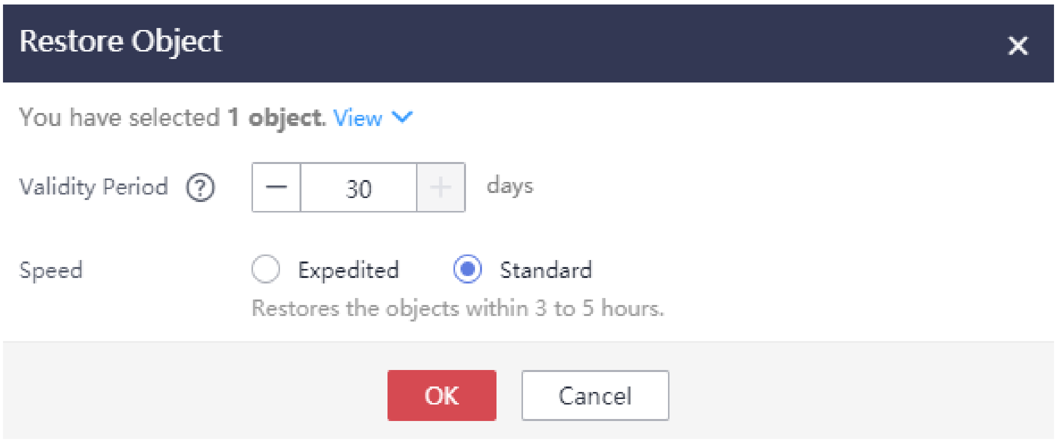
Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Go to the object list in the target bucket.

Step 3 Select the file you want to restore and choose **More** > **Restore Object**. The window shown in **Figure 4-18** is displayed.

Figure 4-18 Restoring an object


Restore Object

You have selected **1 object.** View

Validity Period: 30 days

Speed: ○ Expedited ● Standard

Restores the objects within 3 to 5 hours.

To restore an object, you must configure the validity period and restore speed. **Table 4-6** describes relevant parameters.

Table 4-6 Restoring an object

Parameter	Description
Object Name	Name of the object to be restored
Validity Period	Time duration when an object remains in the **Restored** state after it has been restored. The validity period starts when the object is restored. You can set the validity period to an integer ranging from 1 to 30 (days). The default value is **30**. For example, you set **Validity Period** to **20** when restoring an object. 20 days after the object is restored, its status will change from **Restored** to **Unrestored**.
Speed	How fast an object will be restored. ● **Expedited**: Data smaller than 250 MB can be restored within 1 to 5 minutes. ● **Standard**: All Archive data can be restored within 3 to 5 hours.

Step 4 Click **OK** to confirm the restored file.

To view the file restore status, click ··· next to the object and choose **Object Properties** to view the restoration status. You can download the file only after it has been restored.

----End

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
  - Restoring an object
    ```
    obsutil restore obs://bucket/key [-d=1] [-t=xxx] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-token=xxx]
    ```
  - Restoring objects in batches
    ```
    obsutil restore obs://bucket[/key] -r [-f] [-v] [-d=1] [-t=xxx] [-o=xxx] [-j=1] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-token=xxx]
    ```
  - Restoring all objects in a specific directory at a time
    ```
    obsutil restore obs://bucket/folder/ -r [-f] [-v] [-d=1] [-t=xxx] [-o=xxx] [-j=1] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-token=xxx]
    ```
- In Linux or macOS
  - Restoring an object
    ```
    ./obsutil restore obs://bucket/key [-d=1] [-t=xxx] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-token=xxx]
    ```
  - Restoring objects in batches
    ```
    ./obsutil restore obs://bucket[/key] -r [-f] [-v] [-d=1] [-t=xxx] [-o=xxx] [-j=1] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-token=xxx]
    ```
  - Restoring all objects in a specific directory at a time
    ```
    ./obsutil restore obs://bucket/folder/ -r [-f] [-v] [-d=1] [-t=xxx] [-o=xxx] [-j=1] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-token=xxx]
    ```

Examples

- Take the Windows OS as an example. Run the **obsutil restore obs://bucket-test/key** command to restore a single object whose storage class is **cold**.
  ```
  obsutil restore obs://bucket-test/key

  Start to restore object [key] in the bucket [bucket-test] successfully!
  ```
- Take the Windows OS as an example. Run the **obsutil restore obs://bucket-test -r -f** command to restore objects whose storage class is **cold** in the bucket in batches.
  ```
  obsutil restore obs://bucket-test -r -f

  [==================================================] 100.00% 3s
  Succeed count is:   12        Failed count is:    0
  Metrics [max cost:264 ms, min cost:54 ms, average cost:119.33 ms, average tps:19.70]
  Task id is: 96f104ee-d0bf-40ff-95dd-31dec0d8f4f4
  ```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
key	Mandatory for restoring a single object whose storage class is **cold** Optional for batch restoring objects whose storage class is **cold**	Indicates the name of the object to be restored or the name prefix of the objects to be restored in batches. **NOTE** If this parameter is left blank when batch restoring objects, all objects whose storage class is **cold** in the bucket are restored.

Parameter	Optional or Mandatory	Description
d	Optional (additional parameter)	Storage duration after objects whose storage class is **cold** are restored, in days. The value ranges from 1 to 30. The default value is **1**.
t	Optional (additional parameter)	Options for restoring objects. Possible values are: ● standard ● expedited **NOTE** ● **expedited** indicates that objects can be quickly restored from Archive storage within 1 to 5 minutes and from Deep Archive storage (under limited beta testing) within 3 to 5 hours. ● **standard** indicates that objects can be restored from Archive storage within 3 to 5 hours and from Deep Archive storage (under limited beta testing) within 5 to 12 hours. ● If this parameter is not configured, an expedited restore is used by default.
versionId	Optional for restoring a single object whose storage class is **cold** (additional parameter)	Version ID of the to-be-restored object whose storage class is **cold**
fs	Optional (additional parameter)	Specifies whether the method of listing parallel file systems is applied. If you are listing parallel file systems, you are recommended to add this parameter. **CAUTION** ● With this method, the listing time required varies largely depending on the directory structures. ● After this parameter is enabled, **marker** and **limit** will be ignored. Then, the buckets or parallel file systems (including directories) will be calculated. ● This parameter is only supported by obsutil 5.5.12 and later.
fr	Optional for restoring a single object whose storage class is **cold** (additional parameter)	Generates an operation result list when restoring a single object whose storage class is **cold**.

Parameter	Optional or Mandatory	Description
f	Optional for batch restoring objects whose storage class is **cold** (additional parameter)	Runs in force mode.
r	Mandatory for batch restoring objects whose storage class is **cold** (additional parameter)	Restores objects whose storage class is **cold** in batches by object name prefix.
v	Optional for batch restoring objects whose storage class is **cold** (additional parameter)	Restores versions of objects whose storage class is **cold** in batches by object name prefix.
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success and failure files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **restore_{succeed | failed}_report_***time_TaskId***.txt** By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.
j	Optional for batch restoring objects whose storage class is **cold** (additional parameter)	Maximum number of concurrent tasks for batch restoring objects whose storage class is **cold**. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.

Parameter	Optional or Mandatory	Description
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
token	Optional (additional parameter)	Specifies the user's security token.

Response

Field	Description
Parallel	Parameter **-p** in the request
Jobs	Parameter **-j** in the request
Threshold	Parameter **-threshold** in the request
PartSize	Parameter **-ps** in the request
Exclude	Parameter **-exclude** in the request
Include	Parameter **-include** in the request
TimeRange	Parameter **-timeRange** in the request
VerifyLength	Parameter **-vlength** in the request
VerifyMd5	Parameter **-vmd5** in the request
CheckpointDir	Parameter **-cpd** in the request
OutputDir	Parameter **-o** in the request
ArcDir	Parameter **-arcDir** in the request
Succeed count	Number of successful tasks
Failed count	Number of failed tasks
Skip count	Number of tasks that are skipped during incremental upload, download, or copy, and synchronous upload, download, or copy. **NOTE** Skipped tasks are recorded into successful tasks.

Field	Description
Warning count	Number of tasks that are executed successfully but contain warnings. **NOTE** ● The task for which a warning is generated may be a failure or a success, which needs to be further determined according to the corresponding result list. ● The number of tasks that generate warnings is independent of the number of successful or failed tasks. The total number of tasks is the number of successful tasks plus the number of failed tasks.
Succeed bytes	Number of bytes that are successfully uploaded or downloaded.
max cost	Maximum duration of all tasks, in ms
min cost	Minimum duration of all tasks, in ms
average cost	Average duration of all tasks, in ms
average tps	The average number of tasks completed per second
Task id	Unique ID of an operation, which is used to search for the result list generated in a batch task

4.6.7 Direct Reading Archive Objects

Scenarios

You can enable direct reading if you want to obtain Archive objects immediately or if there are interface compatibility issues when OBS is interconnected with other systems.

With direct reading enabled for a bucket, you can access objects in the Archive storage class without restoring them first. However, downloading or copying Archive objects will incur costs for directly reading these objects. For details, see **Product Pricing Details**.

This section describes how to enable direct reading for Archive objects in a created bucket. If you want to enable direct reading for Archive objects when creating a bucket, refer to **Creating a Bucket**.

NOTE

Direct reading is only available in some regions. For details, see **Function Overview**.

Object Retrieval Comparison Before and After Direct Reading Is Enabled

Item	Direct Reading Disabled (Default)	Direct Reading Enabled
Retrieval method	Objects are restored and then read.	Objects are directly read.
Retrieval cost	Low	High
Retrieval time	Minutes	Milliseconds

Ways to Configure Direct Reading

You can use OBS Console or APIs to configure direct reading.

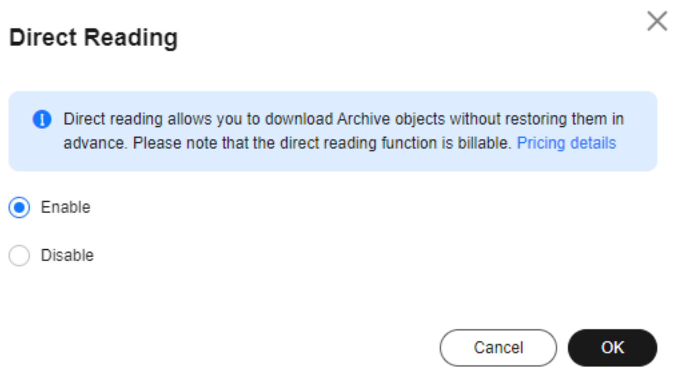
Using OBS Console

Step 1 In the navigation pane, choose **Overview**.

Step 2 In the **Basic Configurations** area, click **Direct Reading**. The **Direct Reading** dialog box is displayed.

Step 3 Select **Enable**.

Figure 4-19 Enabling direct reading



Direct Reading

Direct reading allows you to download Archive objects without restoring them in advance. Please note that the direct reading function is billable. Pricing details

● Enable

○ Disable

Step 4 Click **OK**.

----End

Using APIs

Setting the Direct Reading Policy for Archive Objects in a Bucket

4.7 Deleting Objects

4.7.1 Deleting an Object

Scenarios

To save space and money, you can use OBS Console, the API, SDKs, OBS Browser+, or obsutil to delete unneeded objects.

The following describes how to manually delete objects. For details about the comparison between manual and automatic deletion, see **Table 4-7**.

Table 4-7 Object deletion methods

Method	Description	Rule
Manual deletion	Manual deletion is recommended for deleting a few or irregular objects.	● Individual deletion ● Batch deletion
Automatic deletion	To delete a large number of objects at a time or delete some objects periodically, you can **configure lifecycle rules**.	

NOTE

In big data scenarios, parallel file systems usually have deep directory levels and each directory has a large number of files. In such case, deleting directories from parallel file systems may time out. To address this problem, you are advised to delete directories in either of the following ways:

1. On the Hadoop client that has OBSA, an OBS client plugin, embedded, run the **hadoop fs - rmr obs://{***Name of a parallel file system***}/{***Directory name***}** command.
2. Configure **a lifecycle rule** for directories so that they can be deleted in background based on the preset lifecycle rule.

Deleting Objects from a Bucket with Versioning Disabled, Suspended, or Enabled

Deleting objects may cause data loss. Exercise caution when performing this operation. Versioning can reduce the risk by restoring objects that are accidentally deleted or overwritten.

Scenario	Deleting the Current Object Version	Deleting a Noncurrent Object Version
A versioning-disabled bucket	OBS permanently deletes objects. Deleted objects cannot be recovered.	N/A
A versioning-suspended bucket	● If an object's current version does not have a delete marker and its version ID is not null, the object deletion operation turns the current version into a noncurrent version and inserts a delete marker with an ID of null. The delete marker then becomes the current version. ● If an object's current version does not have a delete marker and its version ID is null, the object deletion operation inserts a delete marker with an ID of null that overwrites the current version. The delete marker then becomes the current version, and the overwritten version cannot be recovered. ● If the current version has a delete marker, no action is taken. **NOTE** ● After versioning is suspended, a newly uploaded object will have a version ID of null. If the newly uploaded object has the same name as an existing object with a null version ID, this new object will overwrite the existing object. For more information about versioning, see **Versioning**. ● By enabling **Historical Versions** in the object list, you can view the version ID of an object.	Deleted noncurrent versions cannot be recovered.
A versioning-enabled bucket	● If an object's current version does not have a delete marker, the object deletion operation inserts a delete marker. Then, the current version becomes a noncurrent version and the delete marker becomes the current version. ● If the current version has a delete marker, no action is taken. **NOTE** ● To permanently delete an object, delete it again from the **Deleted Objects** list. ● To recover a deleted object, undelete it from the **Deleted Objects** list. For details, see **Undeleting an Object**.	Deleted noncurrent versions cannot be recovered.

Ways to Delete Objects

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to delete objects (folders included).

Using OBS Console

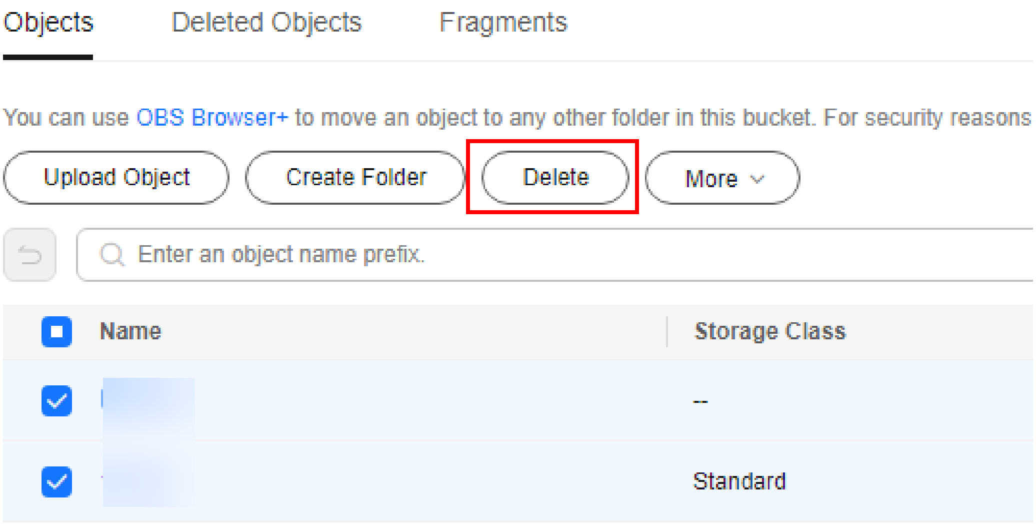
Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 Select the object or folder you want to delete and choose **More** > **Delete** on the right.

You can select multiple objects or folders and click **Delete** above the object list to batch delete them.

Figure 4-20 Deleting a file or folder

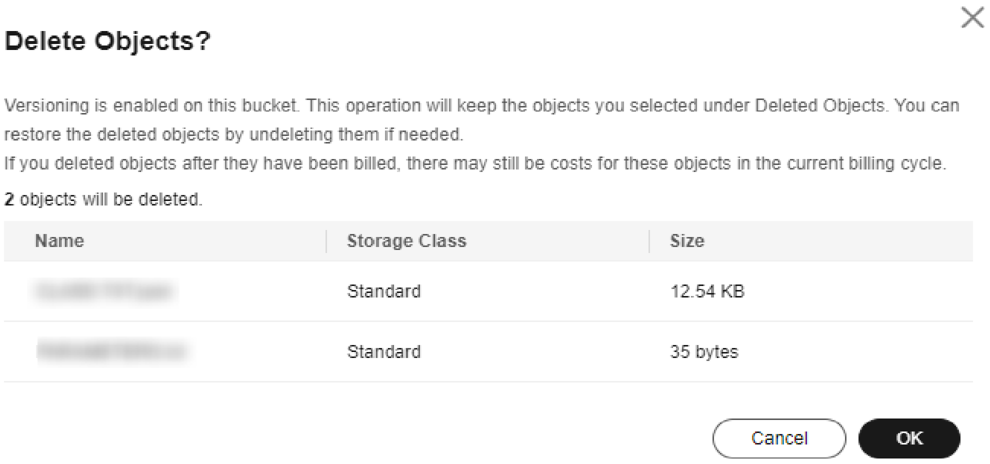


You can use OBS Browser+ to move an object to any other folder in this bucket. For security reasons

	Name	Storage Class
☑		--
☑		Standard

Step 4 Click **OK** to confirm the deletion.

Figure 4-21 Deleting objects



Delete Objects?

Versioning is enabled on this bucket. This operation will keep the objects you selected under Deleted Objects. You can restore the deleted objects by undeleting them if needed.
If you deleted objects after they have been billed, there may still be costs for these objects in the current billing cycle.

2 objects will be deleted.

Name	Storage Class	Size
	Standard	12.54 KB
	Standard	35 bytes

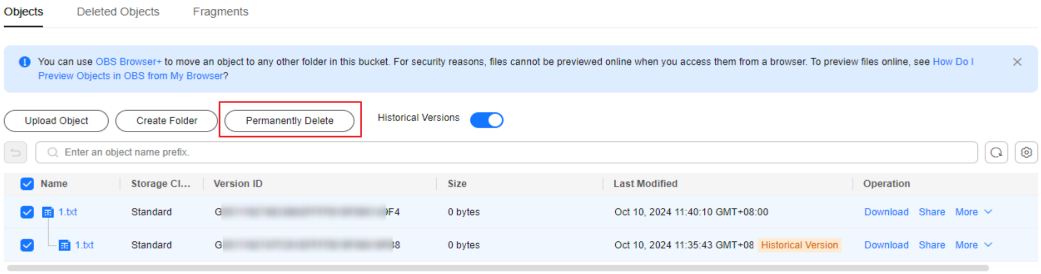
CAUTION

If you delete an object from a bucket with versioning enabled, the object is not permanently deleted but retained in the **Deleted Objects** list. All versions of the object are still kept in the bucket and are billed for storage. If you need to permanently delete the object, see the following steps.

Step 5 Permanently delete an object or a folder from an OBS bucket with versioning enabled:

1. Enable **Historical Versions**.
2. Select all the versions of the object to be permanently deleted and click **Permanently Delete** above the search bar.

Figure 4-22 Permanently deleting an object



Objects Deleted Objects Fragments

You can use OBS Browser+ to move an object to any other folder in this bucket. For security reasons, files cannot be previewed online when you access them from a browser. To preview files online, see How Do I Preview Objects in OBS from My Browser?

Name	Storage Cl...	Version ID	Size	Last Modified	Operation
☑ 1.txt	Standard	G[redacted]F4	0 bytes	Oct 10, 2024 11:40:10 GMT+08:00	Download Share More
☑ 1.txt	Standard	G[redacted]8	0 bytes	Oct 10, 2024 11:35:43 GMT+08 Historical Version	Download Share More

3. Click **OK**.
4.

----End

Using APIs

Deleting an Object

Using SDKs

Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2

Step 3 Go to the bucket from which you want to delete a file or folder.

Step 4 Right-click the file or folder to be deleted and click **Delete**.

You can select multiple files or folders and choose **More** > **Delete** above the list to batch delete them.

Step 5 In the displayed dialog box, click **Yes**.

----End

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
  - Deleting a single object
    ```
    obsutil rm obs://bucket/key [-f] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Deleting objects in batches
    ```
    obsutil rm obs://bucket/[key] -r [-j=1] [-f] [-v] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
- In Linux or macOS
  - Deleting a single object
    ```
    ./obsutil rm obs://bucket/key [-f] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Deleting objects in batches
    ```
    ./obsutil rm obs://bucket/[key] -r [-j=1] [-f] [-v] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```

Examples

- Take the Windows OS as an example. Run the **obsutil rm obs://bucket-test/key -f** command to delete a single object named **key** in bucket **bucket-test**.
  ```
  obsutil rm obs://bucket-test/key -f

  Start at 2024-09-25 04:48:10.1147483 +0000 UTC

  Delete object [key] in the bucket [bucket-test] successfully, cost [152], request id
  [0000016979E1D2B2860BB5181229C72C]
  ```
- Take the Windows OS as an example. Run the **obsutil rm obs://bucket-test -r -f** command to delete all objects in bucket **bucket-test**.
  ```
  obsutil rm obs://bucket-test -r -f

  [=============================================] 100.00% 21s
  Succeed count is:   1313      Failed count is:    0
  Task id: 95936984-f81a-441a-bba0-1fd8254d9241
  ```
- Take the Windows OS as an example. Run the **obsutil rm obs://bucket-test/key -r -f** command to delete all objects and folders prefixed with **key** in bucket **bucket-test**.
  ```
  obsutil rm obs://bucket-test/key -r -f

  [=============================================] 100.00% 21s
  Succeed count is:   10      Failed count is:    0
  Task id: 79ab59ec-7e00-4f22-8c88-465faa834125
  ```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name

Parameter	Optional or Mandatory	Description
key	Mandatory for deleting a single object. Optional for deleting objects in batches.	Indicates the name of the object to be deleted, or the name prefix of the objects to be deleted in batches. **NOTE** If this parameter is left blank when deleting objects in batches, all objects in the bucket are deleted.
fr	Optional for deleting a single object (additional parameter)	Generates an operation result list when deleting an object.
f	Optional (additional parameter)	Runs in force mode.
versionId	Optional for deleting a single object (additional parameter)	Version ID of the object to be deleted.
r	Mandatory for deleting objects in batches (additional parameter)	Deletes objects in batches based on a specified object name prefix. **CAUTION** When you batch delete objects, all objects with the specified prefix will be deleted.
j	Optional for deleting objects in batches (additional parameter)	Indicates the maximum number of concurrent tasks for deleting objects in batches. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
v	Optional for deleting objects in batches (additional parameter)	Deletes versions of an object and the delete markers in batches based on a specified object name prefix.

Parameter	Optional or Mandatory	Description
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success and failure files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **rm_{succeed | failed}_report_***time***_TaskId.txt** ● By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file. ● If there are multiple folders and files and you need to confirm the detailed error information about a failed task, refer to the failure list **rm_failed_report_***time***_TaskId.txt** in the result list folder and the **log files** in the log path.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

Response

Field	Description
Parallel	Parameter **-p** in the request
Jobs	Parameter **-j** in the request

Field	Description
Threshold	Parameter **-threshold** in the request
PartSize	Parameter **-ps** in the request
Exclude	Parameter **-exclude** in the request
Include	Parameter **-include** in the request
TimeRange	Parameter **-timeRange** in the request
VerifyLength	Parameter **-vlength** in the request
VerifyMd5	Parameter **-vmd5** in the request
CheckpointDir	Parameter **-cpd** in the request
OutputDir	Parameter **-o** in the request
ArcDir	Parameter **-arcDir** in the request
Succeed count	Number of successful tasks
Failed count	Number of failed tasks
Skip count	Number of tasks that are skipped during incremental upload, download, or copy, and synchronous upload, download, or copy. **NOTE** Skipped tasks are recorded into successful tasks.
Warning count	Number of tasks that are executed successfully but contain warnings. **NOTE** ● The task for which a warning is generated may be a failure or a success, which needs to be further determined according to the corresponding result list. ● The number of tasks that generate warnings is independent of the number of successful or failed tasks. The total number of tasks is the number of successful tasks plus the number of failed tasks.
Succeed bytes	Number of bytes that are successfully uploaded or downloaded.
max cost	Maximum duration of all tasks, in ms
min cost	Minimum duration of all tasks, in ms
average cost	Average duration of all tasks, in ms
average tps	The average number of tasks completed per second
Task id	Unique ID of an operation, which is used to search for the result list generated in a batch task

4.7.2 Undeleting an Object

Scenarios

If versioning is not enabled for a bucket, deleted objects cannot be recovered.

After a bucket has **versioning** enabled, deleting a file from the **Objects** list does not permanently delete it. Instead, OBS inserts a delete marker and retains the deleted file in the **Deleted Objects** list. You can recover the deleted object by using the **Undelete** operation.

Important Notes

- Files can be undeleted, but folders cannot.

  After you undelete a deleted file, the file is recovered and will appear in the **Objects** list. Then you can perform basic operations on the file as you normally do on other objects. If the file was stored in a folder before deletion, it will be recovered to its original path after you undelete it.
- Deleted files in the **Deleted Objects** still keep multiple versions. When deleting these versions, keep the following points in mind:
  - Deleting a version with the delete marker actually recovers the file instead of permanently deleting it. For details, see **Related Operations**.
  - Deleting a version without the delete marker permanently deletes this version. This version will not be recovered, even if the file is recovered later.

Prerequisites

- Versioning has been enabled for the bucket. For details, see **Versioning**.
- The file to be recovered is in the **Deleted Objects** list and has at least one version without the delete marker.

Ways to Undelete Objects

You can use OBS Console to undelete objects.

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

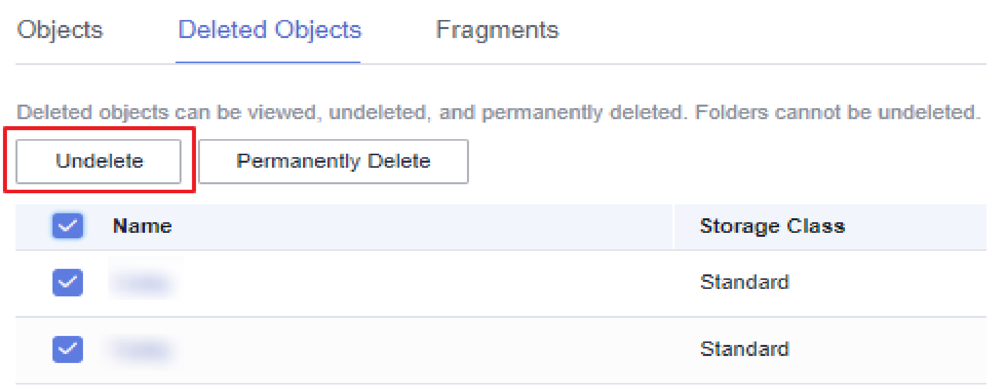
Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 Click **Deleted Objects**.

Step 4 In the row of the deleted file you want to recover, click **Undelete** on the right.

You can select multiple files and click **Undelete** above the object list to batch recover them.

Figure 4-23 Undeleting a file



Objects Deleted Objects Fragments

Deleted objects can be viewed, undeleted, and permanently deleted. Folders cannot be undeleted.

Undelete Permanently Delete

☑	Name	Storage Class
☑	[redacted]	Standard
☑	[redacted]	Standard

----End

Related Operations

Recover a file by deleting its version with the delete marker:

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

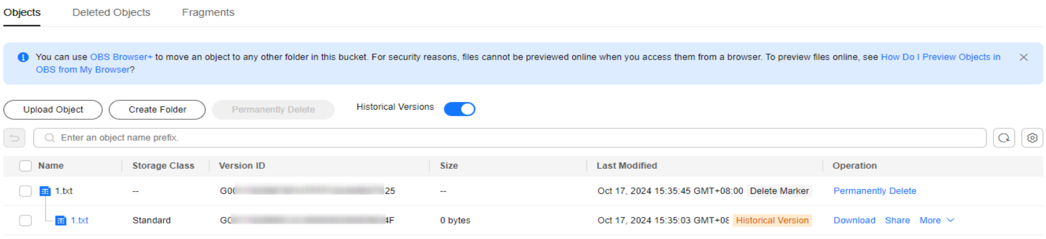
Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 Enable **Historical Versions**.

Step 4 Locate the file to be retrieved with all its historical versions.

- If you delete a version with the **Delete Marker**, the file will be recovered and retained in the **Objects** list.
- If you delete a version without the **Delete Marker**, that version will be permanently deleted.

Figure 4-24 Versions of files in the **Deleted Objects** list



Objects Deleted Objects Fragments

You can use OBS Browser+ to move an object to any other folder in this bucket. For security reasons, files cannot be previewed online when you access them from a browser. To preview files online, see How Do I Preview Objects in OBS from My Browser?

Upload Object Create Folder Permanently Delete Historical Versions

Enter an object name prefix.

□ Name	Storage Class	Version ID	Size	Last Modified	Operation
□ 1.txt	--	G0[redacted]25	--	Oct 17, 2024 15:35:45 GMT+08:00 Delete Marker	Permanently Delete
□ 1.txt	Standard	G0[redacted]4F	0 bytes	Oct 17, 2024 15:35:03 GMT+08 Historical Version	Download Share More

----End

4.7.3 Deleting Fragments

Scenarios

Fragments are the incomplete data in buckets. When data is uploaded to OBS using a multipart upload, there will be fragments generated if the multipart upload fails because of the following and other possible causes:

- Frequent disconnections from the OBS server due to poor network conditions

- Manual upload interruptions
- Device failures
- Sudden power outages

Fragments generated due to upload failures are stored in OBS. You can manually delete the fragments by following the instructions provided in this section. You can also **continue the interrupted or failed multipart upload tasks to delete the fragments** or **configure a lifecycle rule** to automatically delete the fragments periodically.

NOTE

- If there are fragments left in your bucket, your attempt to delete the bucket will fail. In such case, delete all of the fragments first.
- The storage of fragments is billable. If expenditures are stilled generated even if there are no objects in your bucket, check whether there are fragments in it and delete them (if any).

Billing

You will be billed for storing fragments in OBS.

Ways to Delete Fragments

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to delete fragments.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

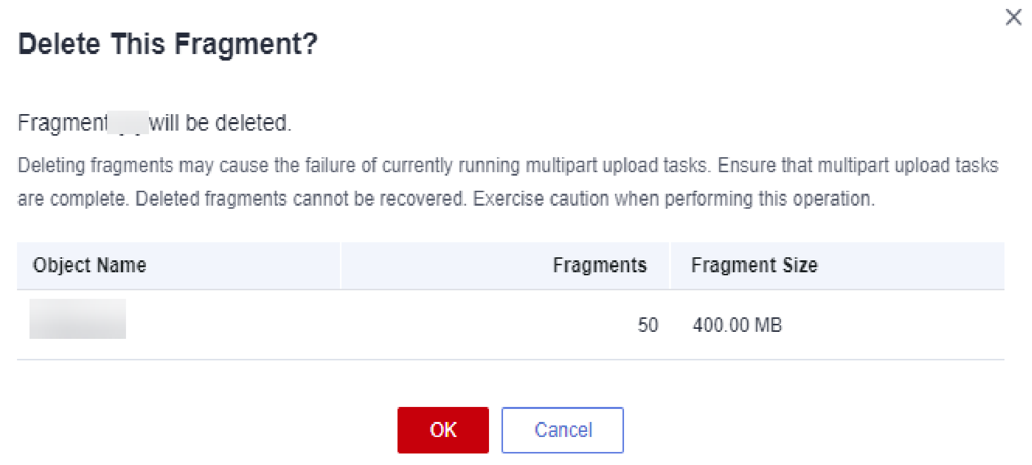
Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 Click **Fragments**, select the fragment that you want to delete, and click **Delete** on the right.

You can also select multiple fragments and click **Delete** above the fragment list to batch delete them.

Step 4 Click **OK** to confirm the deletion.

Figure 4-25 Deleting fragments



Delete This Fragment?

Fragment [redacted] will be deleted.

Deleting fragments may cause the failure of currently running multipart upload tasks. Ensure that multipart upload tasks are complete. Deleted fragments cannot be recovered. Exercise caution when performing this operation.

Object Name	Fragments	Fragment Size
[redacted]	50	400.00 MB

----End

Using APIs

You can use APIs to delete fragments from a bucket by doing as follows:

1. Use the **API for listing the initiated multipart uploads in a bucket** to list all the multipart uploads and obtain their upload IDs.
2. Use the **API for aborting a multipart upload** to abort the multipart uploads to delete all fragments.

Using SDKs

With an OBS SDK, fragments are generated only when a multipart upload is not completed. You can delete the fragments by aborting the multipart upload. The procedure is as follows:

1. Use the **ObsClient.listMultipartUploads** API to list all multipart uploads and obtain their upload IDs.
2. Use the **ObsClient.abortMultipartUpload** API to abort all of the listed multipart uploads to delete all fragments.

Java	**Python**	C: not supported	**Go**	**BrowserJS**	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**

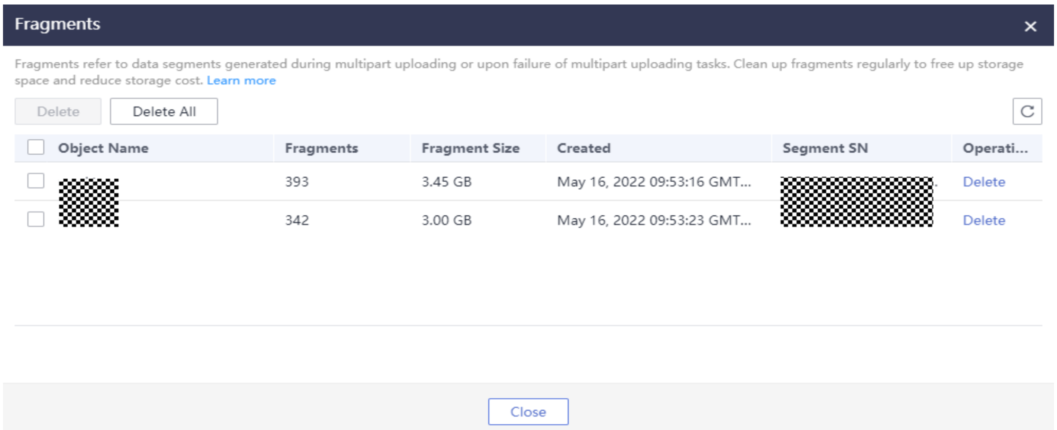
Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Select the bucket you want and click **Fragments**.

Step 3 In the **Fragments** window, select the unwanted fragments and click **Delete** above the list. You can also click **Delete All** above the list to delete all fragments. Click ↻ in the upper right corner to refresh the fragment list, as shown in **Figure 4-26**.

Figure 4-26 Fragment management



Fragments ×

Fragments refer to data segments generated during multipart uploading or upon failure of multipart uploading tasks. Clean up fragments regularly to free up storage space and reduce storage cost. Learn more

Delete | Delete All

□ Object Name	Fragments	Fragment Size	Created	Segment SN	Operati...
□ [redacted]	393	3.45 GB	May 16, 2022 09:53:16 GMT...	[redacted]	Delete
□ [redacted]	342	3.00 GB	May 16, 2022 09:53:23 GMT...	[redacted]	Delete

Close

Step 4 In the displayed **Warning** dialog box, confirm the delete information and click **Yes**.

Step 5 Click **Yes**.

The **Fragments** window is displayed. You can close this window to go back to the OBS Browser+ homepage.

----End

Using the CLI Tool - obsutil

With obsutil, you can delete fragments by deleting the corresponding multipart upload. If a bucket has more than one multipart uploads, you need to delete all the multipart uploads to delete all fragments in the bucket.

For details, see **Deleting a Multipart Upload**.

Related Operations

OBS allows you to manage tasks. If a multipart upload fails or is interrupted, you can resume it with the following tools. Once the upload is completed, fragments will disappear.

Table 4-8 Tools for deleting fragments

Tool	Operation Guide
OBS Browser+	On OBS Browser+, run the upload task that generates the fragments. Once it completes, fragments are cleared. For details, see **Managing Failed Tasks** and **Managing Suspended Tasks**.
obsutil	With obsutil, you can resume a failed upload based on its task ID recorded in the result list. After the upload is completed, the fragments will disappear. For details, see **Resuming a Failed Upload**.

Tool	Operation Guide
OBS SDKs	OBS SDKs provide resumable upload by recording the upload progress. You can read the recorded progress and resume an interrupted or failed multipart upload. For details, see the section about resumable upload in the *SDK Reference*.

4.8 Managing Folders

4.8.1 Creating a Folder

Scenarios

You can create a folder in a bucket to facilitate classification and management of data stored in OBS.

Unlike a file system, OBS does not involve the concepts of file and folder. For easy data management, OBS provides a method to simulate folders. On OBS Console or OBS Browser+, an object is simulated as a folder by adding a slash (/) to the end of the object name. When objects are listed by calling the API, the obtained object name is the path of the object, and the content following the last slash (/) is the actual object name. If the path ends with a slash (/), it indicates that the object is a folder. The hierarchical depth of the object does not affect the performance of accessing the object.

Ways to Create a Folder

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to create a folder.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 Click **Create Folder**, or click a folder in the object list to open it and click **Create Folder**.

Step 4 In the **Folder Name** text box, enter a name for the folder.

- You can create single-level or multi-level folders.
- The name cannot contain the following special characters: \:*?"<>|
- The name cannot start or end with a period (.) or slash (/).
- The folder's absolute path cannot exceed 1,023 characters.
- Any single slash (/) separates and creates multiple levels of folders at once.
- The name cannot contain two or more consecutive slashes (/).

Step 5 Click **OK**.

----End

Using the API

To create a folder in OBS is to create an object whose size is 0 and whose name ends with a slash (/), in essential. For details, see **Using APIs**.

Using SDKs

To create a folder in OBS is to create an object whose size is 0 and whose name ends with a slash (/), in essential. For details, see **Using SDKs**.

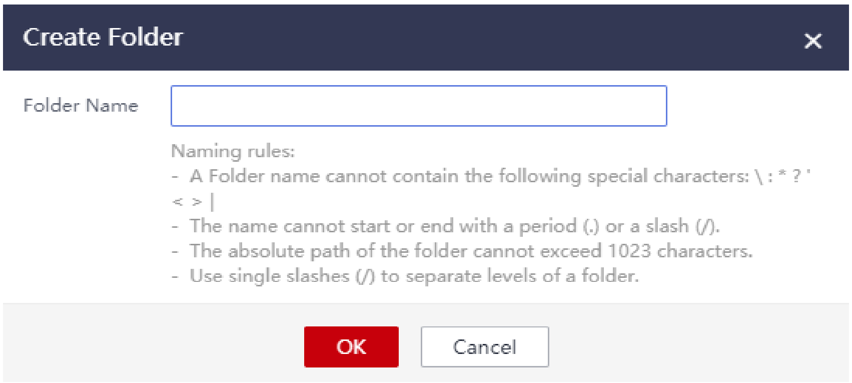
Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Click the bucket where you want to create a folder and click **Create Folder**.

Step 3 In the displayed dialog box, enter a folder name and click **OK**, as shown in **Figure 4-27**.

Figure 4-27 Creating a folder



Create Folder

Folder Name

Naming rules:
- A Folder name cannot contain the following special characters: \ : * ? ' < > |
- The name cannot start or end with a period (.) or a slash (/).
- The absolute path of the folder cannot exceed 1023 characters.
- Use single slashes (/) to separate levels of a folder.

- A folder name cannot contain the following special characters: \ : * ? ' < > |
- A folder name cannot start or end with a period (.) or slash (/).
- A folder name cannot exceed 1,023 bytes. The length of a folder name is the sum of the length of its own and the length of its upper-level directories. The total length cannot exceed 1,023 bytes. Directories of different levels are automatically separated by slashes (/). For example, if the upper-level directory of **folder01** is **folder02**, the name length of folder **folder01** is the length of **folder02/folder01/**.
- A single slash (/) separates and creates multiple levels of folders.

NOTE

If an access deny message is displayed when you are creating a folder, possible causes are as follows:

- Access to the bucket is restricted by an ACL. For example, you do not have the write permission for the bucket.
- Access to the bucket is restricted by a bucket policy. For example, you do not have the write permission for the bucket, or write operations cannot be performed on the bucket during the current period.

If such message is displayed, check ACL and policy settings of the bucket and resolve the problem accordingly.

Step 4 Click **OK**.

----End

Using the CLI Tool - obsutil

Command Line Structure

- In Windows
  - Creating a folder in a specified bucket
    ```
    obsutil mkdir obs://bucket/folder[/subfolder1/subfolder2] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Creating a folder in the local file system
    ```
    obsutil mkdir folder_url [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
- In Linux or macOS
  - Creating a folder in a specified bucket
    ```
    ./obsutil mkdir obs://bucket/folder[/subfolder1/subfolder2] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Creating a folder in the local file system
    ```
    ./obsutil mkdir folder_url [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```

Examples

- Take the Windows OS as an example. Run the **obsutil mkdir obs://bucket-test/folder1/folder2** command to create a folder in a bucket.
  ```
  obsutil mkdir obs://bucket-test/folder1/folder2

  The bucket [bucket-test] does not support POSIX, create folder(s) step by step
  Create folder [obs://bucket-test/folder1/] successfully, request id
  [0000016979E1D23C860BB3D8E4577C5E]
  Create folder [obs://bucket-test/folder1/folder2] successfully, request id
  [0000016979E1D2B2860BB5181229C72C]
  ```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory when creating a folder in a specified bucket	Bucket name

Parameter	Optional or Mandatory	Description
folder	Mandatory when creating a folder in a specified bucket	Folder path in the bucket. This value can contain multi-level folders. Separate each level with a slash (/).
folder_url	Mandatory when creating a folder in the local file system	Folder path in the local file system. The value can be an absolute path or a relative path.
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

Follow-up Procedure

You can click **Copy Path** on the right to copy the path of the folder and share it with others. Then they can open the bucket where the folder is stored and enter the path in the search box above the object list to find the folder.

4.8.2 Sharing a Folder

Scenarios

You can share your folders in OBS with others.

Folder can be shared temporarily or permanently.

Background Information

- Temporary sharing:

  Share a folder to other users through the temporary URL of the file. Folder sharing is temporary. All sharing URLs are only valid for a limited period of time.

  Folders can be temporarily shared by access code or URL:

  Sharing by access code

  You need to prepare a six-digit extraction code before sharing a folder. After the sharing task is created, OBS aggregates the download links of all objects in the folder to a static website that is hosted by a public OBS bucket. Then anyone who has the created temporary URL and access code can access the static website and download the shared files.

  A temporary URL consists of the access domain name and the temporary authentication information of a folder. Example:

  ```
  https://e-share.obs-website.ap-southeast-1.myhuaweicloud.com:443/image.png?token=xxx
  ```

  For details about the temporary authentication method and parameters, see **Authentication of Signature in a URL** in the *Object Storage Service API Reference*.

  Sharing by URL

  You can specify a validity period and then share the generated link with others. Anyone can use a signature to access all objects in the shared folder.

  A temporary URL for accessing an object in a folder consists of a bucket domain name (prefix), an object path, and signature information (suffix).

- Permanent sharing:

  If you set the permission for an object to allow anonymous users to read it, anonymous users can access the object through the URL that you shared. For details about how to configure permissions, see **Granting Public Read Permissions on Objects to Anonymous Users**.

  For details about how to obtain the object access URL, see **How Do I Obtain the Access Path to an Object?**

  If access to an object using the URL fails, fix the problem by referring to **Why Can't I Access an Object Through Its URL?**

Limitations and Constraints

- The following table describes the URL validity period of the folders that are temporarily shared in different ways.

Way	URL Validity Period
OBS Console	The URL is valid for 1 minute to 18 hours. After an object is shared, the system will generate a URL that contains the temporary authentication information, valid for five minutes since its generation by default. Each time you change the validity period of a URL, OBS obtains the authentication information again to generate a new URL for sharing. The new URL becomes valid at the moment the validity period is changed. If you need a longer validity period, use OBS Browser+ that allows a validity period of up to one year to share the object. If you want a shared object to be permanently valid, **use a bucket policy to grant anonymous users the public read permission for the folder**.
API	Configure parameter **Expires** to specify when a temporary authorization expires. The temporary authorization expires in 24 hours.
OBS Browser+	When you log in to OBS Browser+ using an account and password, a shared URL can be valid for 24 hours at most. The default validity period is 10 hours. If a longer validity period is required, use the permanent AK/SK for login.
obsutil	Use the additional parameter **vp** to specify how long an authorization code is valid. The default validity period is one day. The allowed units include m (months), w (weeks), d (days), h (hours), min (minutes), and s (seconds). If no time unit is specified, the value is calculated in seconds.

- Folder sharing is restricted to a few regions only. For details about the supported regions, see **Function Overview**.
- Only buckets of version 3.0 support folder sharing. For details about how to query the bucket version, see **Checking OBS Version (OBS 2.0 or OBS 3.0)**.
- Archive objects in a folder must be restored in the bucket before they can be downloaded.
- Deep Archive objects in a folder must be restored in the bucket before they can be downloaded.

Ways to Share a Folder

You can use OBS Console, OBS Browser+, or obsutil to share a folder.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

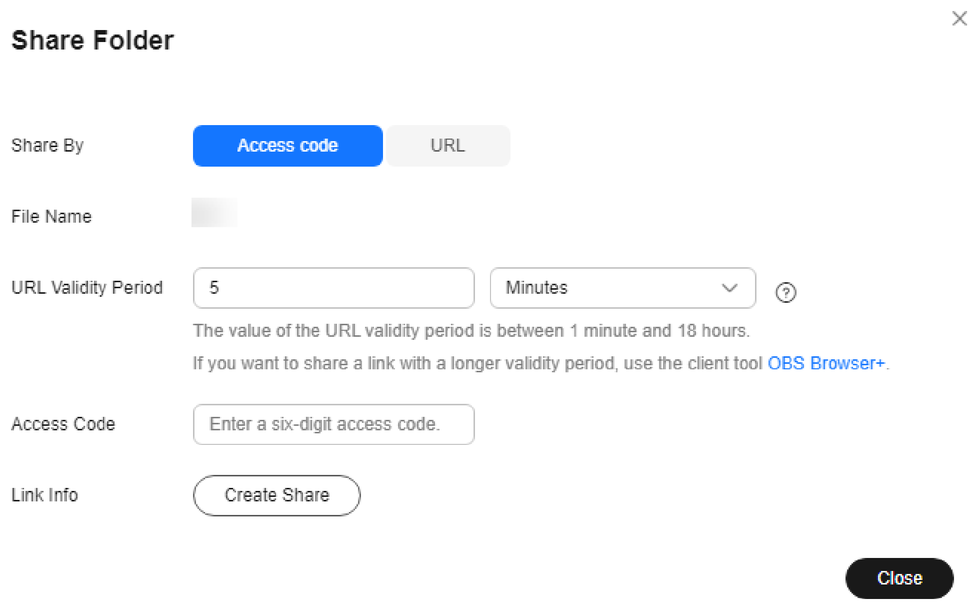
Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 Locate the folder you want to share and click **Share** in the **Operation** column. The **Share Folder** dialog box is displayed.

Step 4 Share the folder by access code or URL.

Step 5 Method 1: Share the folder by access code.

Figure 4-28 Sharing a folder



Share Folder

Share By: Access code | URL

File Name

URL Validity Period: 5 Minutes

The value of the URL validity period is between 1 minute and 18 hours.

If you want to share a link with a longer validity period, use the client tool OBS Browser+.

Access Code: Enter a six-digit access code.

Link Info: Create Share

1. Choose **Access code** for **Share By**.
2. Configure parameters.

Table 4-9 Parameters for sharing a folder with an access code

Parameter	Description
URL Validity Period	A validity period is from one minute to 18 hours. The default value is five minutes. Within the URL validity period, anyone who has the URL can access the folder.
Access Code	A six-digit code. An access code is required to access objects in the shared folder.

3. Click **Create Share** to generate sharing URL of the folder.
4. Send the URL and access code to others for them to access the folder.
5. Verify that other users can perform the following operations:
   a. Access the shared folder in a browser.
      i. Open the shared URL in a web browser.
      ii. In the dialog box that is displayed, enter the access code and access objects in the shared folder.

b. Access the shared folder on OBS Browser+.

i. Start OBS Browser+.

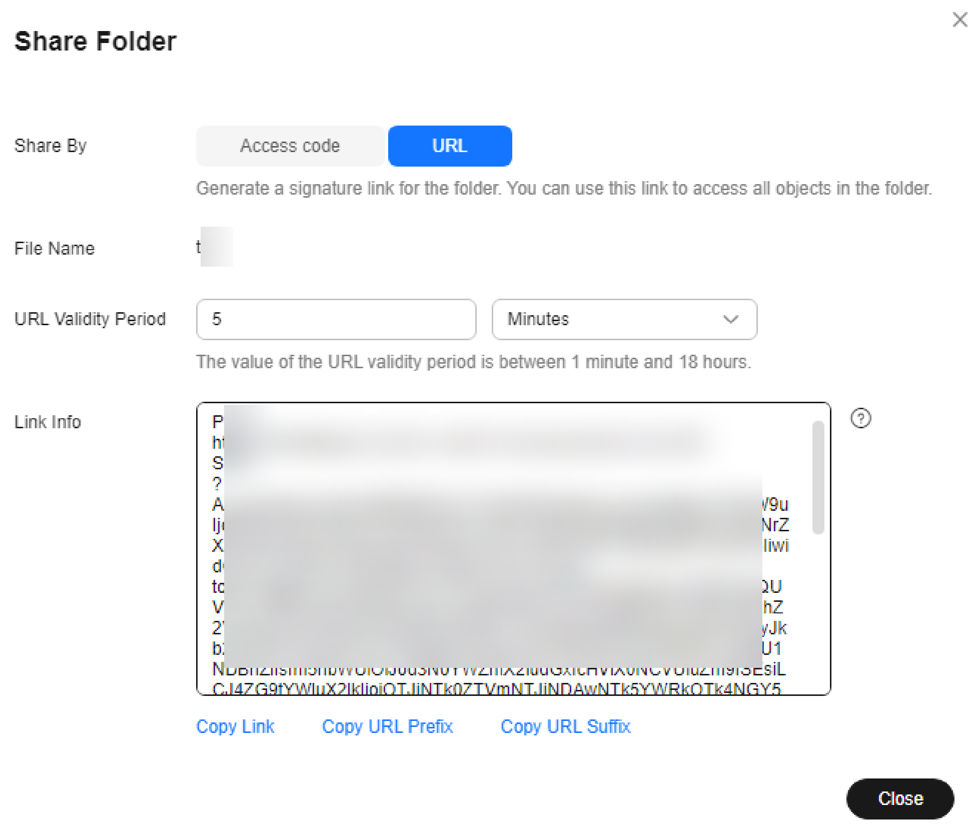
ii. On the login page, click **Authorization Code Login**.

iii. Enter the authorization code and access code.

iv. Click **Log In** to access the shared folder.

v. Method 2: Share the folder by URL.

Figure 4-29 Sharing by URL



Share Folder

Share By: Access code | URL (selected)

Generate a signature link for the folder. You can use this link to access all objects in the folder.

File Name: t

URL Validity Period: 5 Minutes

The value of the URL validity period is between 1 minute and 18 hours.

Link Info: [redacted]

Copy Link Copy URL Prefix Copy URL Suffix

Close

1) Choose **URL** for **Share By**.

2) Configure parameters.

Table 4-10 Parameters for sharing a folder by URL

Parameter	Description
URL Validity Period	A validity period is from one minute to 18 hours. The default value is five minutes. Within the URL validity period, anyone who has the URL can access the folder.

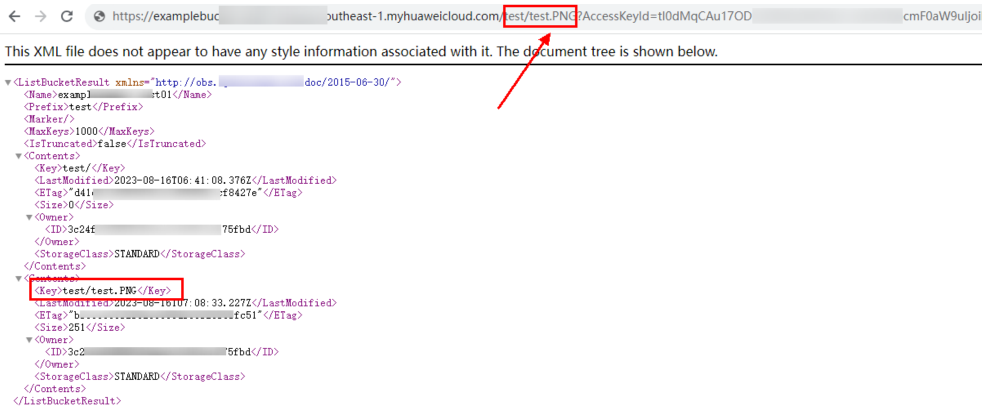
3) Click **Copy Link** and share the link with another user. The user then can use this link to access all objects in this folder. The sharing link consists of the bucket domain name (prefix) and

signature information (suffix). Users can add an object path after the prefix of a sharing link to access or download the specified object in a folder, as shown in **Figure 4-30**.

4) Verify that a user can use the sharing link to access all objects in the folder.

   1) Open a browser.
   2) Enter the sharing link in the address box and press **Enter** to list all objects in the folder.
   3) Copy the object path and paste it after the prefix.
   4) Press **Enter**. You can then access and download the specified object.

Figure 4-30 Accessing an object with a sharing link



https://examplebuc[redacted]outheast-1.myhuaweicloud.com/test/test.PNG?AccessKeyId=tl0dMqCAu17OD[redacted]cmF0aW9uljoi

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<ListBucketResult xmlns="http://obs.[redacted]doc/2015-06-30/">
   <Name>exampl[redacted]t01</Name>
   <Prefix>test</Prefix>
   <Marker/>
   <MaxKeys>1000</MaxKeys>
   <IsTruncated>false</IsTruncated>
 ▼<Contents>
     <Key>test/</Key>
     <LastModified>2023-08-16T06:41:08.376Z</LastModified>
     <ETag>"d41[redacted]f8427e"</ETag>
     <Size>0</Size>
   ▼<Owner>
       <ID>3c24f[redacted]75fbd</ID>
     </Owner>
     <StorageClass>STANDARD</StorageClass>
   </Contents>
 ▼<Contents>
     <Key>test/test.PNG</Key>
     <LastModified>2023-08-16T07:08:33.227Z</LastModified>
     <ETag>"b[redacted]fc51"</ETag>
     <Size>251</Size>
   ▼<Owner>
       <ID>3c2[redacted]5fbd</ID>
     </Owner>
     <StorageClass>STANDARD</StorageClass>
   </Contents>
</ListBucketResult>
```

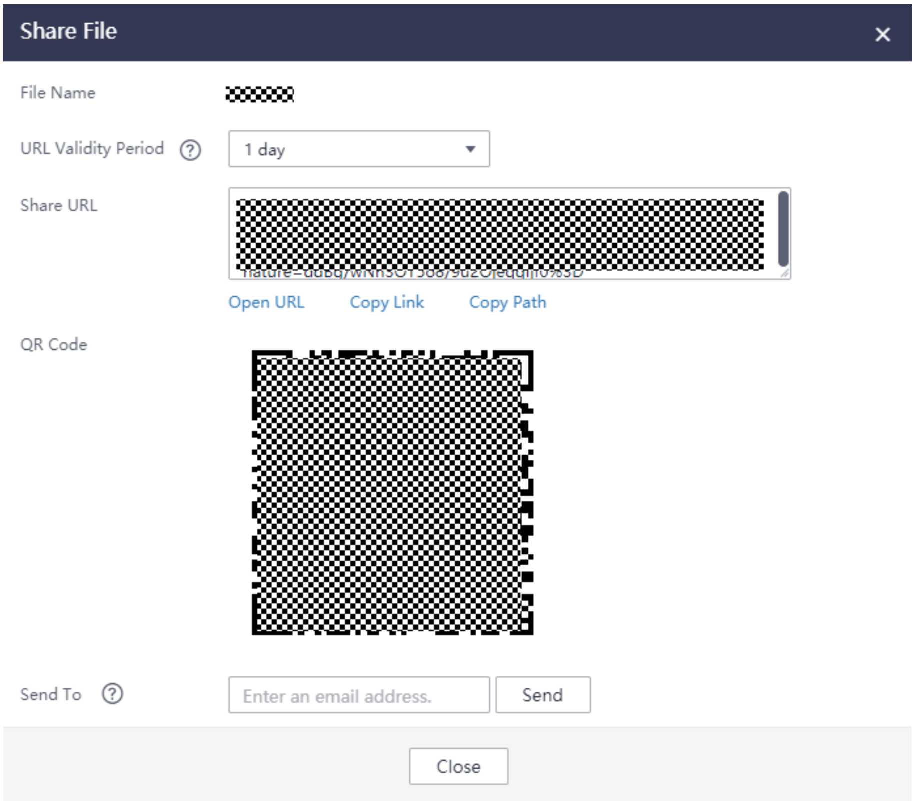
----End

Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Locate the object you want to share and click 🔗 next to the object. The window shown in **Figure 4-31** or **Figure 4-32** is displayed.

Figure 4-31 Sharing a file



Share File

File Name [redacted]

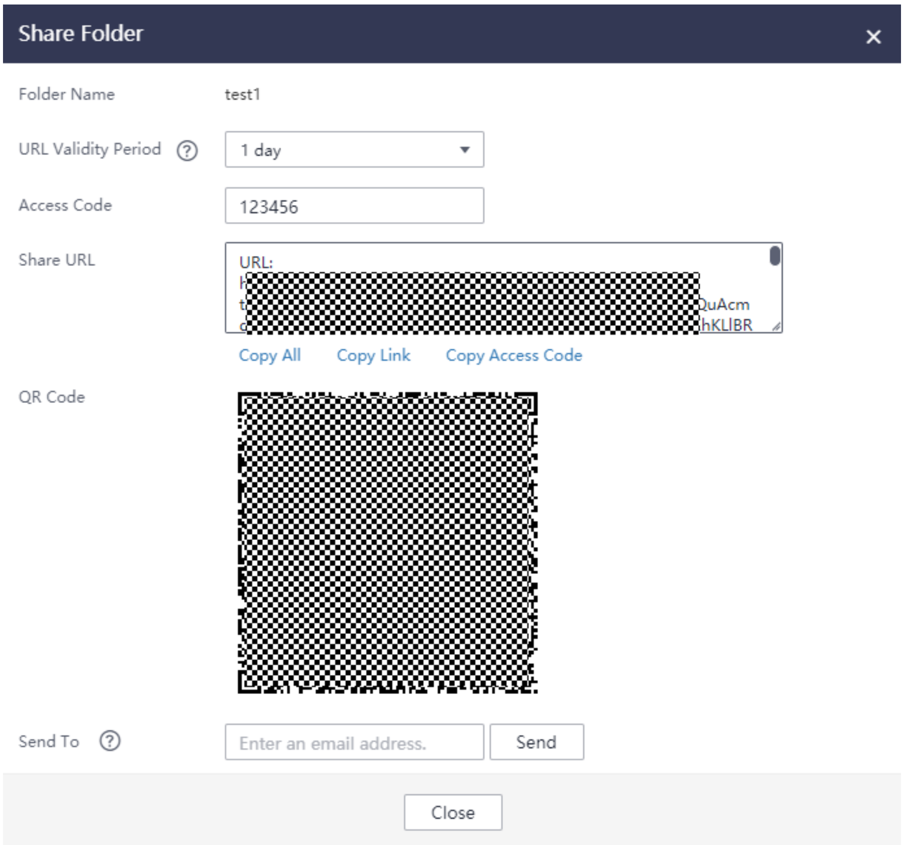
URL Validity Period: 1 day

Share URL [redacted]

QR Code [redacted]

Send To: Enter an email address.

Figure 4-32 Sharing a folder



Share Folder

Folder Name: test1

URL Validity Period: 1 day

Access Code: 123456

Share URL: URL: [redacted]

QR Code: [redacted]

Send To: Enter an email address.

Step 3 Click **Copy Link** to copy the object link to a browser and open it. Then you can access the object.

NOTE

To allow anonymous users to access Archive objects using URLs, ensure that these objects are in the **Restored** state.

----End

Using the CLI Tool - obsutil

You can use obsutil to specify the bucket name, object name prefix, and access code to create an authorization code for directory sharing.

Command Line Structure

- In Windows
```
obsutil create-share obs://bucket[/prefix] [-ac=xxx] [-vp=xxx] [-dst=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```
- In Linux or macOS
```
./obsutil create-share obs://bucket[/prefix] [-ac=xxx] [-vp=xxx] [-dst=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

Examples

- In Windows, you can run the **obsutil create-share obs://bucket/test/ -ac=123456 -vp=1m** command to create an authorization code that is valid within one month.

```
obsutil create-share obs://bucket/test/ -ac=123456 -vp=1m

Authorization Code:
token=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Access Code:
123456

Valid Until:
Sat, 26 Oct 2019 11:28:10 GMT +8000
```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Name of a bucket
prefix	Optional	Prefix of an object name. If this parameter is specified, objects starting with this prefix are shared. If this parameter is left blank, all objects in the bucket are shared. **NOTE** It is recommended that the value end with a slash (/).
ac	Optional (additional parameter)	Access code. **NOTE** - If no access code is passed using this parameter, obsutil tool prompts you to enter the access code in interactive mode. - An access code is a six-digit string.
vp	Optional (additional parameter)	Validity period of an authorization code. The default value is one day, indicating that the generated authorization code is valid for only one day. **NOTE** - This parameter supports different time units, including: **m** (month), **w** (week), **d** (day), **h** (hour), **min** (minute), and **s** (second). For example, **1d** indicates that the authorization code is valid within one day, **2w** indicates that the code is valid within two weeks, and **3h** indicates that the code is valid within three hours. - The default time unit is **s** (second), for example, **3600** indicates that the authorization code is valid within 3600 seconds.
dst	Optional (additional parameter)	Path for storing the generated authorization code.

Parameter	Optional or Mandatory	Description
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

Response

Field	Description
Authorization Code	Code for authorization
Access Code	Access code
Valid Until	Expiration time

4.8.3 Analyzing a Folder

You can use Folder Analysis to calculate the size of a folder and the number of objects in a folder.

NOTE

If you use Folder Analysis to calculate the size of a folder, you will be billed for the API calls used to list objects in the bucket. For details, see **Billing Description**.

Constraints

- Only folders with no more than 1 million objects can be analyzed. If your folder contains more objects than that, use the **listing objects** command of obsutil for analysis.

- It may take some time to analyze folders with a lot of objects.

Ways to Access Folder Analysis

You can access Folder Analysis through the OBS console or obsutil, but you cannot access it through an API, SDK, or OBS Browser+.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

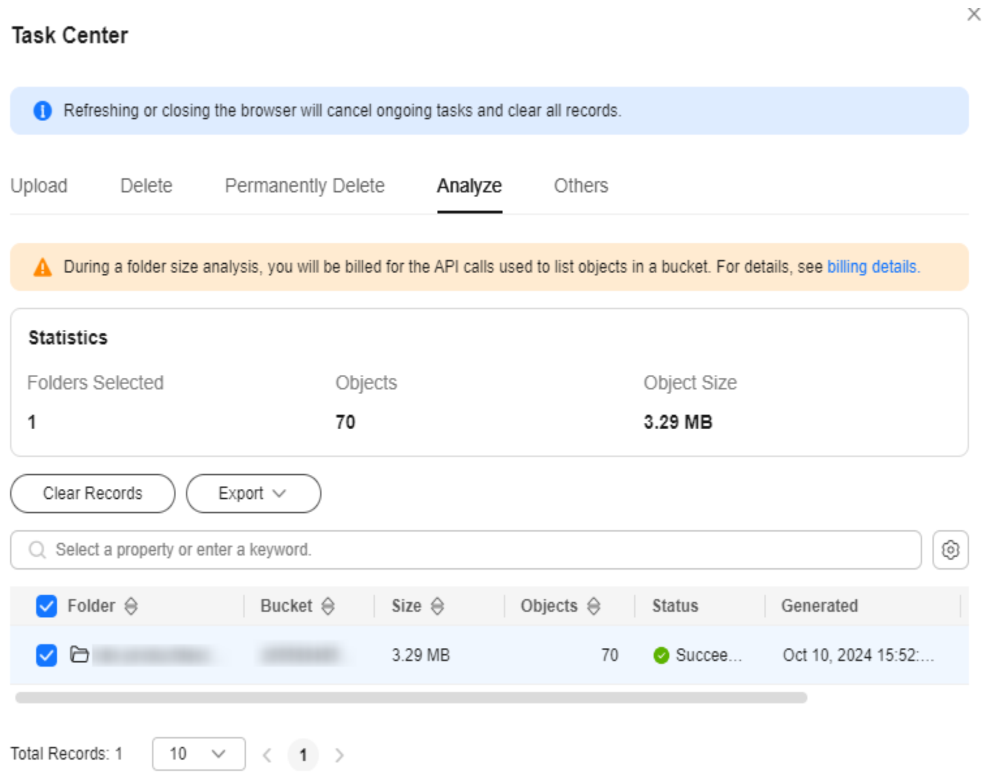
Step 2 In the bucket list, click the bucket you want to operate.

Step 3 Click **Analyze** in the **Operation** column of the folder, or click ↻ in the **Size** column.

Step 4 Click **Task Center** on the right corner of the page, and check the analysis result on the **Analyze** tab.

You can see information such as the bucket that a folder belongs to, the size of the folder, the number of objects in the folder, and the analysis status.

Figure 4-33 Folder Analysis



Task Center

Refreshing or closing the browser will cancel ongoing tasks and clear all records.

Upload | Delete | Permanently Delete | **Analyze** | Others

During a folder size analysis, you will be billed for the API calls used to list objects in a bucket. For details, see billing details.

Statistics

Folders Selected	Objects	Object Size
1	70	3.29 MB

Clear Records | Export

Select a property or enter a keyword.

☑ Folder	Bucket	Size	Objects	Status	Generated
☑ [redacted]	[redacted]	3.29 MB	70	Succee...	Oct 10, 2024 15:52:...

Total Records: 1 | 10 | 1

Step 5 (Optional) Click **Clear Records** to clear the analysis results.

Step 6 (Optional) Click **Export** to export all or selected analysis results.

Step 7 (Optional) Select a property to filter the analysis results.

----End

Using the CLI Tool - obsutil

- **Counting the number of objects in a folder**

obsutil cannot be directly used to obtain the number and size of objects in a folder. You need to **list objects**.

- **Querying the folder size**

Take Linux OS as an example. Run the **./obsutil ls obs://bucket-test/test/ -du -limit=0** command to query the size of the **test** folder in bucket **bucket-test**.

```
./obsutil ls obs://bucket-test/test/ -du -limit=0

Start at 2023-03-16 06:40:18.2773873 +0000 UTC

Listing objects .

Remove the -du parameter to view more information
[DU] Total prefix [test/] size: 990.85MB
```

⚠ CAUTION

- The **-du** parameter is only supported by obsutil 5.4.6 and later.

5 Permissions Control

5.1 Configuring IAM Permissions

5.1.1 Creating an IAM User and Granting OBS Permissions

You can use **IAM** for fine-grained access control over your OBS resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing OBS resources.
- Manage permissions on a principle of least permissions (PoLP) basis.
- Entrust a Huawei Cloud account or cloud service to perform efficient O&M on your OBS resources.

If your Huawei Cloud account does not require individual IAM users, skip this chapter.

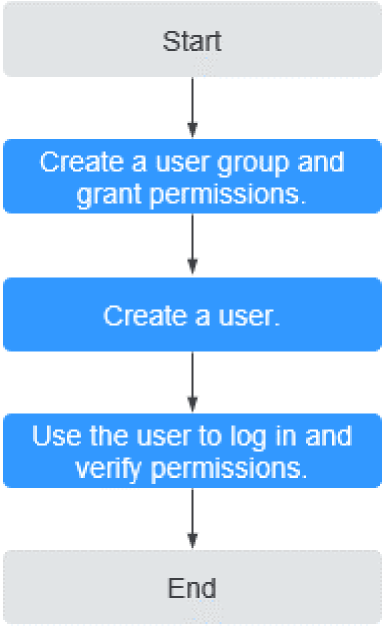
Figure 5-1 shows the procedure for granting permissions.

Prerequisites

You have learned about the **OBS permissions** that can be assigned to a user group.

Process

Figure 5-1 Process of granting an IAM user the OBS permissions


Start
Create a user group and grant permissions.
Create a user.
Use the user to log in and verify permissions.
End

The example here describes how to grant an IAM user the **Tenant Guest** permission for OBS.

1. **Create a user group and assign permissions**.

   Create a user group on the IAM console and assign the group the **Tenant Guest** permission.

2. **Create an IAM user and add it to the user group**.

   Create a user on the IAM console and add the user to the group created in **1**.

3. **Log in to the console** and verify permissions.

   Log in to OBS Console using the newly created user, and verify that the assigned permission has taken effect:

   - Choose **Object Storage Service** from the service list to go to the OBS homepage. If the list of buckets is displayed and you can view the basic information about any bucket, but you cannot create or delete buckets or perform any other operations, the granted **Tenant Guest** permission has already taken effect.
   - Go to an OBS bucket. If the list of objects is displayed and you can download objects, but you cannot upload or delete objects or perform any other operations, the **Tenant Guest** permission granted has already taken effect.

5.1.2 OBS Custom Policies

Custom policies can be created to supplement the system-defined policies of OBS. For the actions supported for custom policies, see **Bucket-Related Actions** and **Object-Related Actions**.

You can create custom policies in either of the following two ways:

- Visual editor: Select cloud services, actions, resources, and request conditions without the need to know policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**. The following provides examples of common OBS custom policies.

Example Custom Policies

- Example 1: Grant users all OBS permissions.

  This policy allows users to perform any operation on OBS.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "obs:*:*"
            ]
        }
    ]
}
```

- Example 2: Grant users all OBS Console permissions.

  This policy allows users to perform all operations on OBS Console.

  When a user logs in to OBS Console, the user may access resources of other services such as audit information in CTS, acceleration domain names in CDN, and keys in KMS. Therefore, in addition to the OBS permissions in example 1, you also need to configure the access permissions to other services. CDN is a global service, while CTS and KMS are regional ones. You need to configure the **Tenant Guest** permission for the global project and regional projects based on the services and regions that you use.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "obs:*:*"
            ]
        }
    ]
}
```

- Example 3: Grant users the read-only permission for all directories in a bucket.

  This policy allows users to list and download all objects in bucket **obs-example**.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "obs:object:GetObject",
                "obs:bucket:ListBucket"
            ],
            "Resource": [
                "obs:*:*:object:obs-example/*",
                "obs:*:*:bucket:obs-example"
            ]
        }
    ]
}
```

- Example 4: Grant users the read-only permission for a specified directory in a bucket.

  This policy allows users to download objects in only the **my-project/** directory of bucket **obs-example**. Objects in other directories can be listed but cannot be downloaded.

  ```
  {
      "Version": "1.1",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": [
                  "obs:object:GetObject",
                  "obs:bucket:ListBucket"
              ],
              "Resource": [
                  "obs:*:*:object:obs-example/my-project/*",
                  "obs:*:*:bucket:obs-example"
              ]
          }
      ]
  }
  ```

- Example 5: Grant users the read/write permissions for a specified directory in a bucket.

  This policy allows users to list, download, upload, and delete objects in the **my-project** directory of bucket **obs-example**.

  ```
  {
      "Version": "1.1",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": [
                  "obs:object:GetObject",
                  "obs:object:ListMultipartUploadParts",
                  "obs:bucket:ListBucket",
                  "obs:object:DeleteObject",
                  "obs:object:PutObject"
              ],
              "Resource": [
                  "obs:*:*:object:obs-example/my-project/*",
                  "obs:*:*:bucket:obs-example"
              ]
          }
      ]
  }
  ```

- Example 6: Grant users all permissions for a bucket.

  This policy allows users to perform any operation on bucket **obs-example**.

  ```
  {
      "Version": "1.1",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": [
                  "obs:*:*"
              ],
              "Resource": [
                  "obs:*:*:bucket:obs-example",
                  "obs:*:*:object:obs-example/*"
              ]
          }
      ]
  }
  ```

- Example 7: Grant users the permission to deny object upload.

  A deny policy must be used together with other policies. If the permissions assigned to a user contain both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

  If you grant the system policy OBS OperateAccess to a user but do not want the user to have the object upload permission (which is also a permission allowed by OBS OperateAccess), you can create a custom policy besides the OBS OperateAccess policy, to deny the user's upload permission. According to the authorization principle, the policy with the deny statement takes precedence. This means the user can perform all operations allowed by OBS OperateAccess except for the upload operation. The following is an example of a deny policy:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "obs:object:PutObject"
            ]
        }
    ]
}
```

5.1.3 OBS Resources

A resource is an object that exists within a service. OBS resources include buckets and objects. You can select these resources by specifying their paths.

Table 5-1 OBS resources and their paths

Resource Type	Resource Name	Path
Buckets	Bucket	[Format] **obs:*:*:bucket:***Bucket name* [Notes] IAM automatically generates the prefix **obs:*:*:bucket:** for bucket resource paths. By adding *Bucket name* to the end of the generated prefix, you can define a specific path. An asterisk ***** is allowed to indicate any bucket. An example is given as follows: **obs:*:*:bucket:***

Resource Type	Resource Name	Path
Objects	Object	[Format] **obs:*:*:object:***Bucket name*/*Object name* [Notes] IAM automatically generates the prefix **obs:*:*:object:** for object resource paths. By adding *Bucket name/Object name* to the end of the generated prefix, you can define a specific path. An asterisk ***** is allowed to any object in the bucket. An example is given as follows: **obs:*:*:object:my-bucket/my-object/*** (indicating any object in the **my-object** directory of bucket **my-bucket**)

5.1.4 OBS Request Conditions

Request conditions are useful in determining when a custom policy is in effect. A request condition consists of a condition key and an operator. Condition keys are either global or service-level and are used in the condition elements of a policy statement. **Global condition keys** (starting with **g:**) are available for operations of all services, while service-level condition keys (starting with a service name acronym like **obs:**) are available only for operations of a specific service. An operator is used together with a condition key to form a complete condition statement.

OBS has a group of predefined condition keys that can be used in IAM. For example, to define an allow permission, you can use the condition key **obs:SourceIp** to filter matching requesters by IP address.

The condition keys and operators supported by OBS are the same as those in the bucket policy. When configuring condition keys in IAM, start them with **obs:**. For details, see **Policy Format**.

5.2 Configuring a Bucket Policy

5.2.1 Creating a Bucket Policy with a Template

OBS Console provides bucket policy templates for eight typical scenarios. You can use these templates to quickly configure bucket policies.

Ways to Configure Bucket Policies

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to configure bucket policies.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

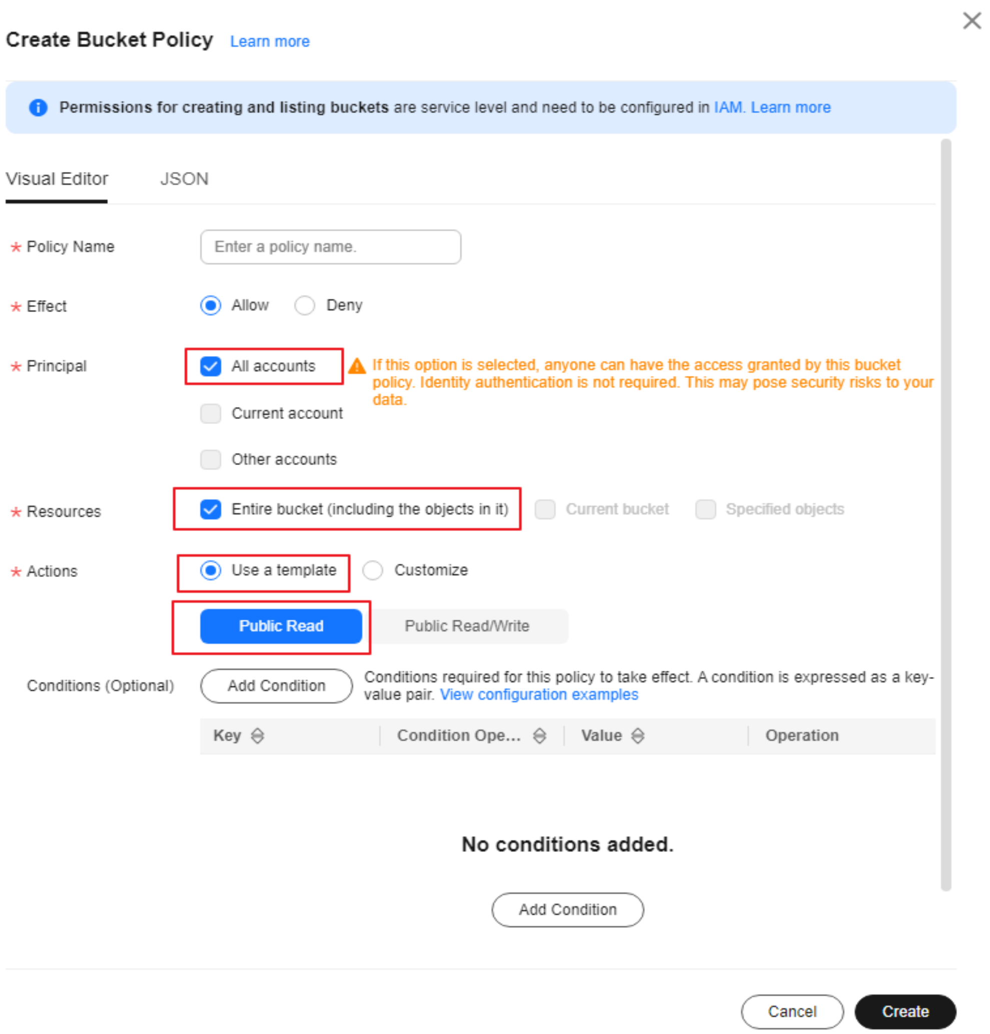
Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 4 Click **Create**.

Step 5 Choose a policy template. For details about the parameters, see **Bucket Policies**.

Figure 5-2 Choosing the Public Read template



Create Bucket Policy Learn more

Permissions for creating and listing buckets are service level and need to be configured in IAM. Learn more

Visual Editor | JSON

* Policy Name: Enter a policy name.

* Effect: ● Allow ○ Deny

* Principal: ☑ All accounts — If this option is selected, anyone can have the access granted by this bucket policy. Identity authentication is not required. This may pose security risks to your data.
☐ Current account
☐ Other accounts

* Resources: ☑ Entire bucket (including the objects in it) ☐ Current bucket ☐ Specified objects

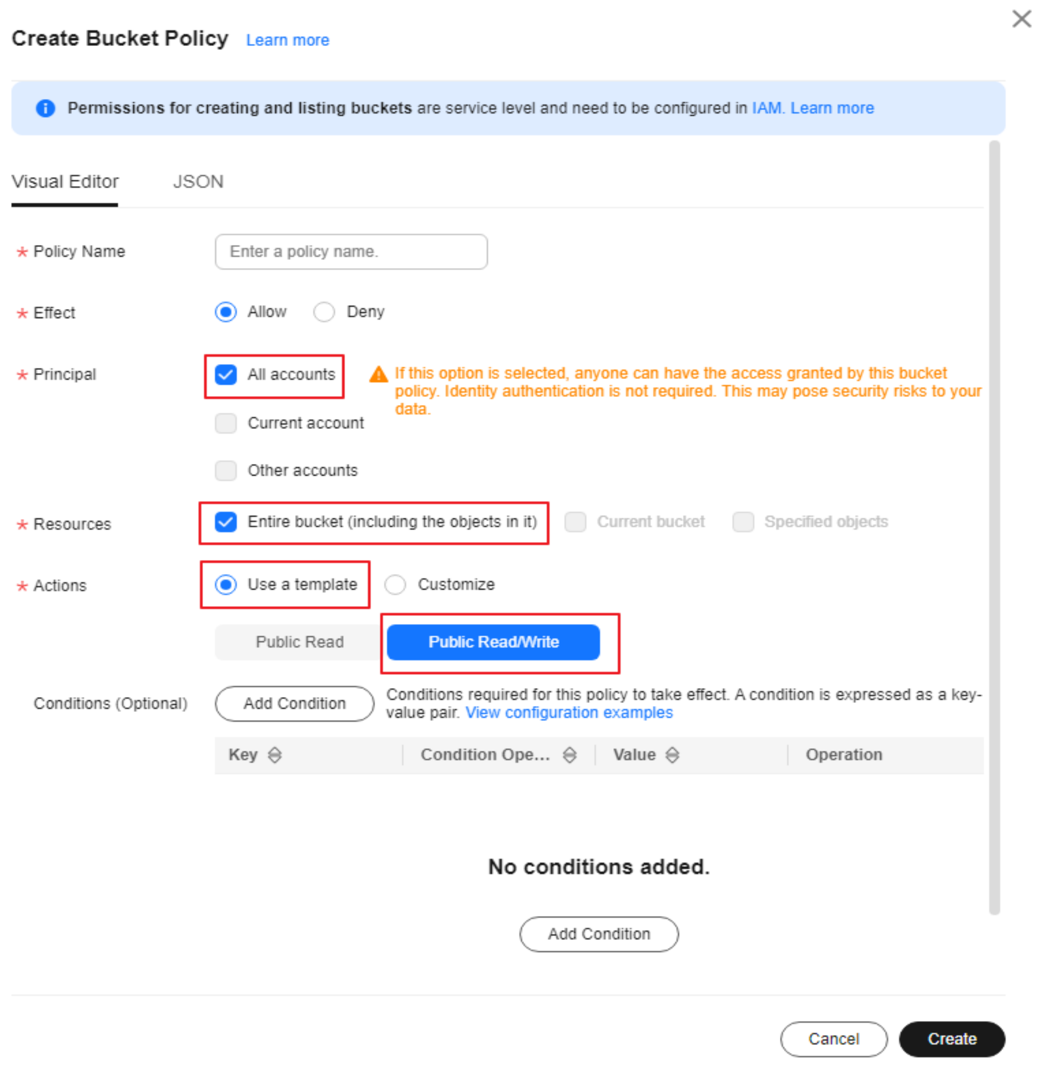
* Actions: ● Use a template ○ Customize
Public Read | Public Read/Write

Conditions (Optional): Add Condition — Conditions required for this policy to take effect. A condition is expressed as a key-value pair. View configuration examples

Key	Condition Ope...	Value	Operation

No conditions added.

Figure 5-3 Choosing the Public Read/Write template



Create Bucket Policy Learn more

Permissions for creating and listing buckets are service level and need to be configured in IAM. Learn more

* Policy Name: Enter a policy name.

* Effect: ● Allow ○ Deny

* Principal: ☑ All accounts □ Current account □ Other accounts

If this option is selected, anyone can have the access granted by this bucket policy. Identity authentication is not required. This may pose security risks to your data.

* Resources: ☑ Entire bucket (including the objects in it) □ Current bucket □ Specified objects

* Actions: ● Use a template ○ Customize

Public Read | **Public Read/Write**

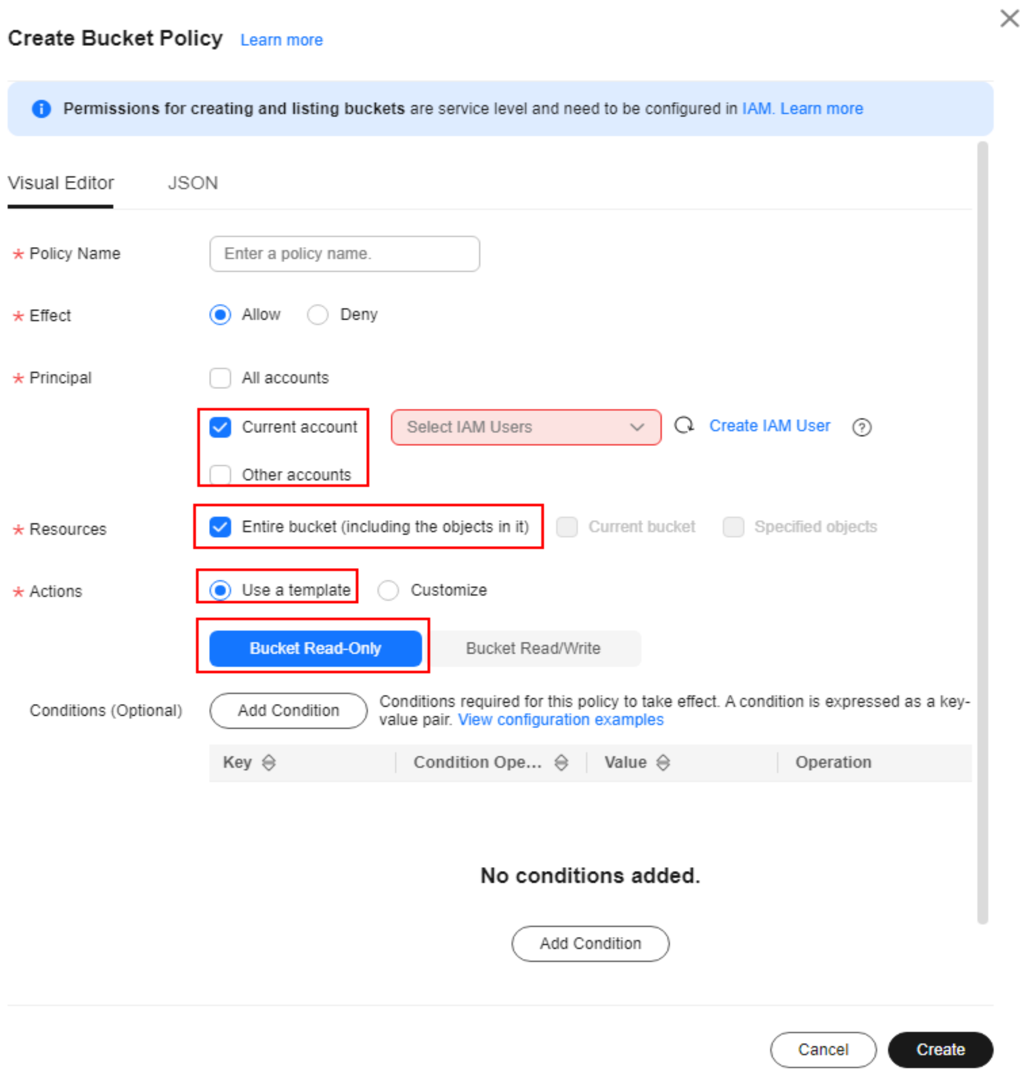
Conditions (Optional): Add Condition

Conditions required for this policy to take effect. A condition is expressed as a key-value pair. View configuration examples

Key	Condition Ope...	Value	Operation

No conditions added.

Figure 5-4 Choosing the Bucket Read-Only template



Create Bucket Policy Learn more

Permissions for creating and listing buckets are service level and need to be configured in IAM. Learn more

Visual Editor JSON

* Policy Name: Enter a policy name.

* Effect: ● Allow ○ Deny

* Principal: □ All accounts
☑ Current account Select IAM Users Create IAM User
□ Other accounts

* Resources: ☑ Entire bucket (including the objects in it) □ Current bucket □ Specified objects

* Actions: ● Use a template ○ Customize
Bucket Read-Only Bucket Read/Write

Conditions (Optional) Add Condition Conditions required for this policy to take effect. A condition is expressed as a key-value pair. View configuration examples

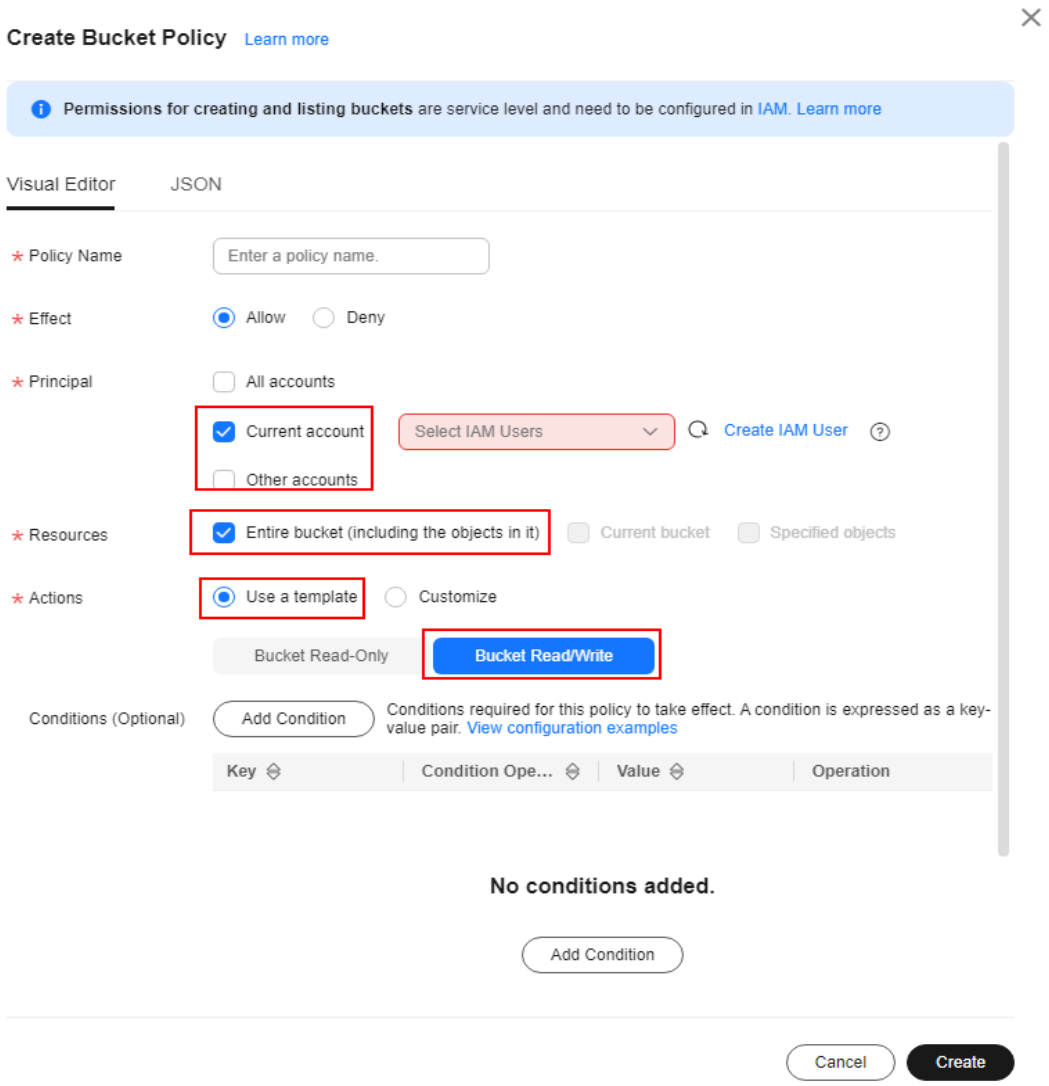
Key	Condition Ope...	Value	Operation

No conditions added.

Add Condition

Cancel Create

Figure 5-5 Choosing the Bucket Read/Write template



Create Bucket Policy Learn more

Permissions for creating and listing buckets are service level and need to be configured in IAM. Learn more

Visual Editor | JSON

* Policy Name: Enter a policy name.

* Effect: ● Allow ○ Deny

* Principal: □ All accounts
☑ Current account — Select IAM Users — Create IAM User
□ Other accounts

* Resources: ☑ Entire bucket (including the objects in it) □ Current bucket □ Specified objects

* Actions: ● Use a template ○ Customize
Bucket Read-Only | **Bucket Read/Write**

Conditions (Optional): Add Condition — Conditions required for this policy to take effect. A condition is expressed as a key-value pair. View configuration examples

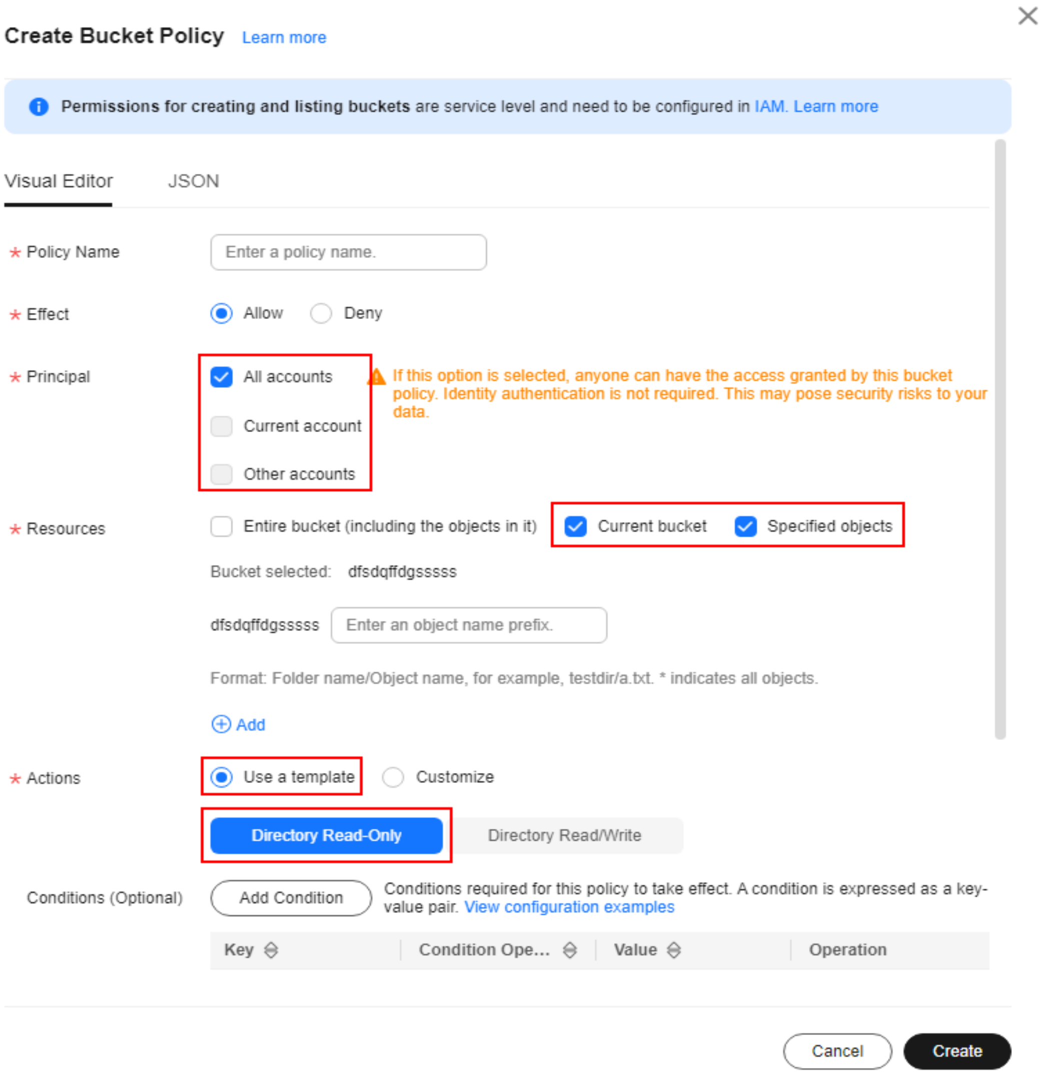
Key	Condition Ope...	Value	Operation

No conditions added.

Add Condition

Cancel | Create

Figure 5-6 Choosing the Directory Read-Only template



Create Bucket Policy Learn more

Permissions for creating and listing buckets are service level and need to be configured in IAM. Learn more

* Policy Name: Enter a policy name.

* Effect: ● Allow ○ Deny

* Principal: ☑ All accounts □ Current account □ Other accounts

If this option is selected, anyone can have the access granted by this bucket policy. Identity authentication is not required. This may pose security risks to your data.

* Resources: □ Entire bucket (including the objects in it) ☑ Current bucket ☑ Specified objects

Bucket selected: dfsdqffdgsssss

dfsdqffdgsssss Enter an object name prefix.

Format: Folder name/Object name, for example, testdir/a.txt. * indicates all objects.

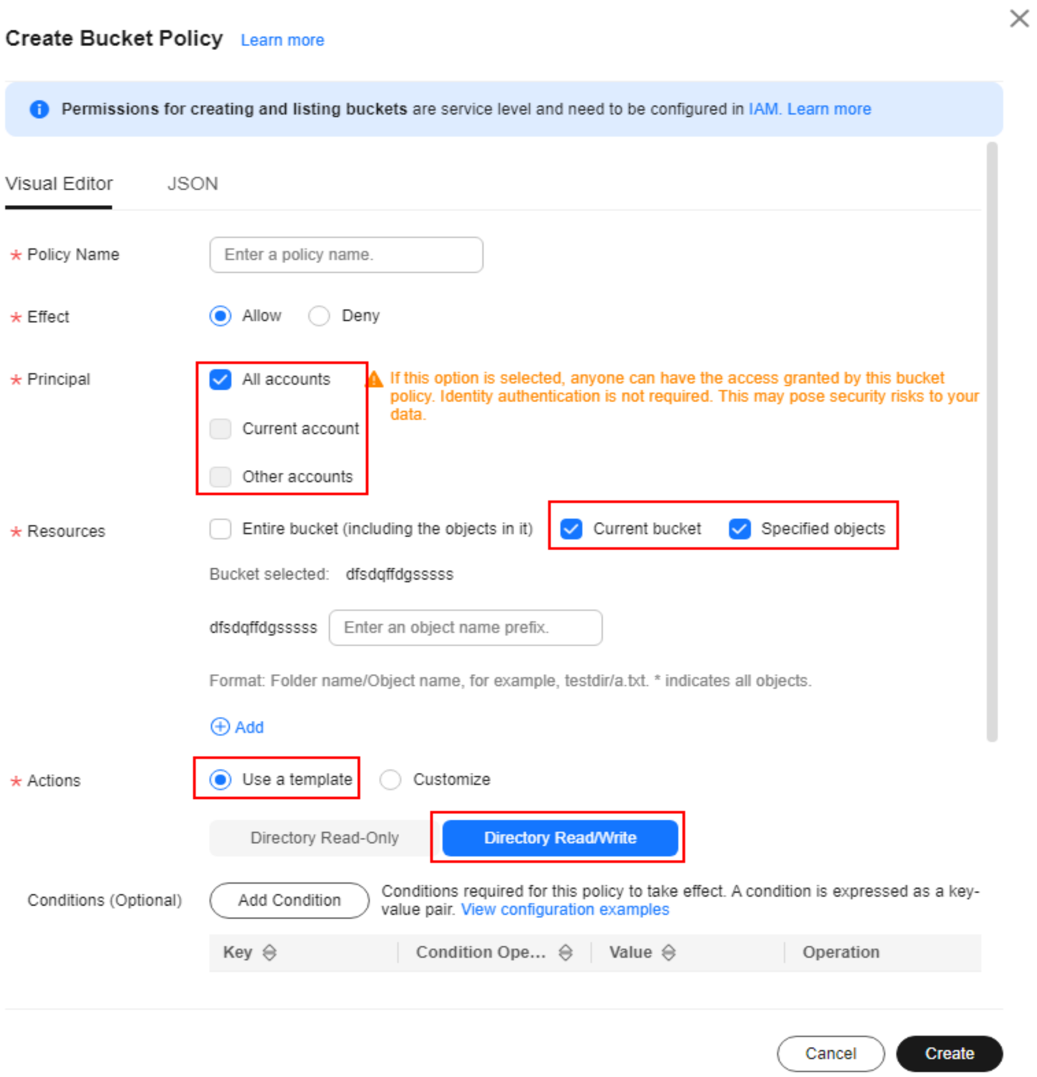
* Actions: ● Use a template ○ Customize

Conditions (Optional): Add Condition

Conditions required for this policy to take effect. A condition is expressed as a key-value pair. View configuration examples

Key	Condition Ope...	Value	Operation

Figure 5-7 Choosing the Directory Read/Write template



Create Bucket Policy Learn more

Permissions for creating and listing buckets are service level and need to be configured in IAM. Learn more

Visual Editor JSON

* Policy Name: Enter a policy name.

* Effect: ● Allow ○ Deny

* Principal: ☑ All accounts □ Current account □ Other accounts

If this option is selected, anyone can have the access granted by this bucket policy. Identity authentication is not required. This may pose security risks to your data.

* Resources: □ Entire bucket (including the objects in it) ☑ Current bucket ☑ Specified objects

Bucket selected: dfsdqffdgsssss

dfsdqffdgsssss Enter an object name prefix.

Format: Folder name/Object name, for example, testdir/a.txt. * indicates all objects.

Add

* Actions: ● Use a template ○ Customize

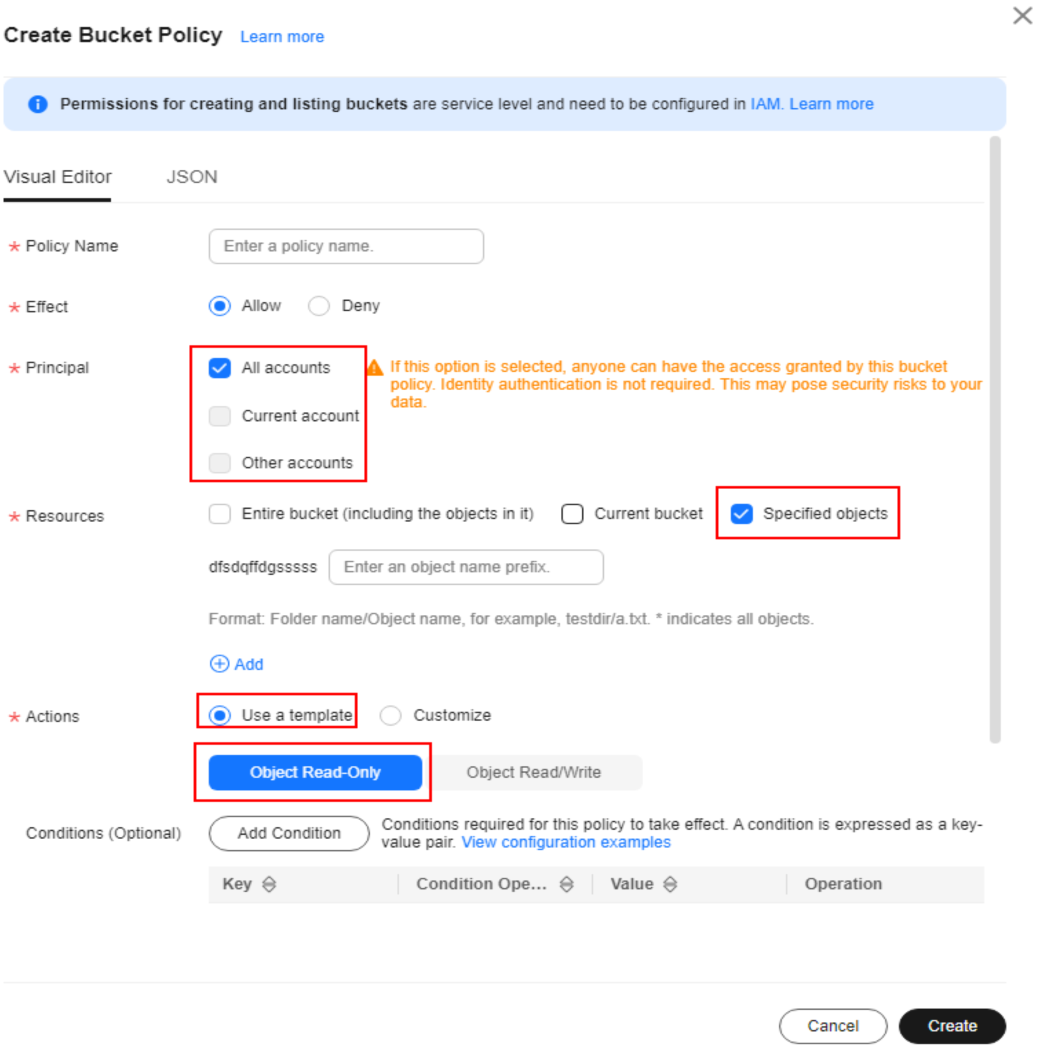
Directory Read-Only [✓ Directory Read/Write]

Conditions (Optional): Add Condition

Conditions required for this policy to take effect. A condition is expressed as a key-value pair. View configuration examples

Key	Condition Ope...	Value	Operation

Figure 5-8 Choosing the Object Read-Only template



Create Bucket Policy Learn more

Permissions for creating and listing buckets are service level and need to be configured in IAM. Learn more

Visual Editor | JSON

* Policy Name: Enter a policy name.

* Effect: ● Allow ○ Deny

* Principal: ☑ All accounts □ Current account □ Other accounts

If this option is selected, anyone can have the access granted by this bucket policy. Identity authentication is not required. This may pose security risks to your data.

* Resources: □ Entire bucket (including the objects in it) □ Current bucket ☑ Specified objects

dfsdqffdgsssss Enter an object name prefix.

Format: Folder name/Object name, for example, testdir/a.txt. * indicates all objects.

Add

* Actions: ● Use a template ○ Customize

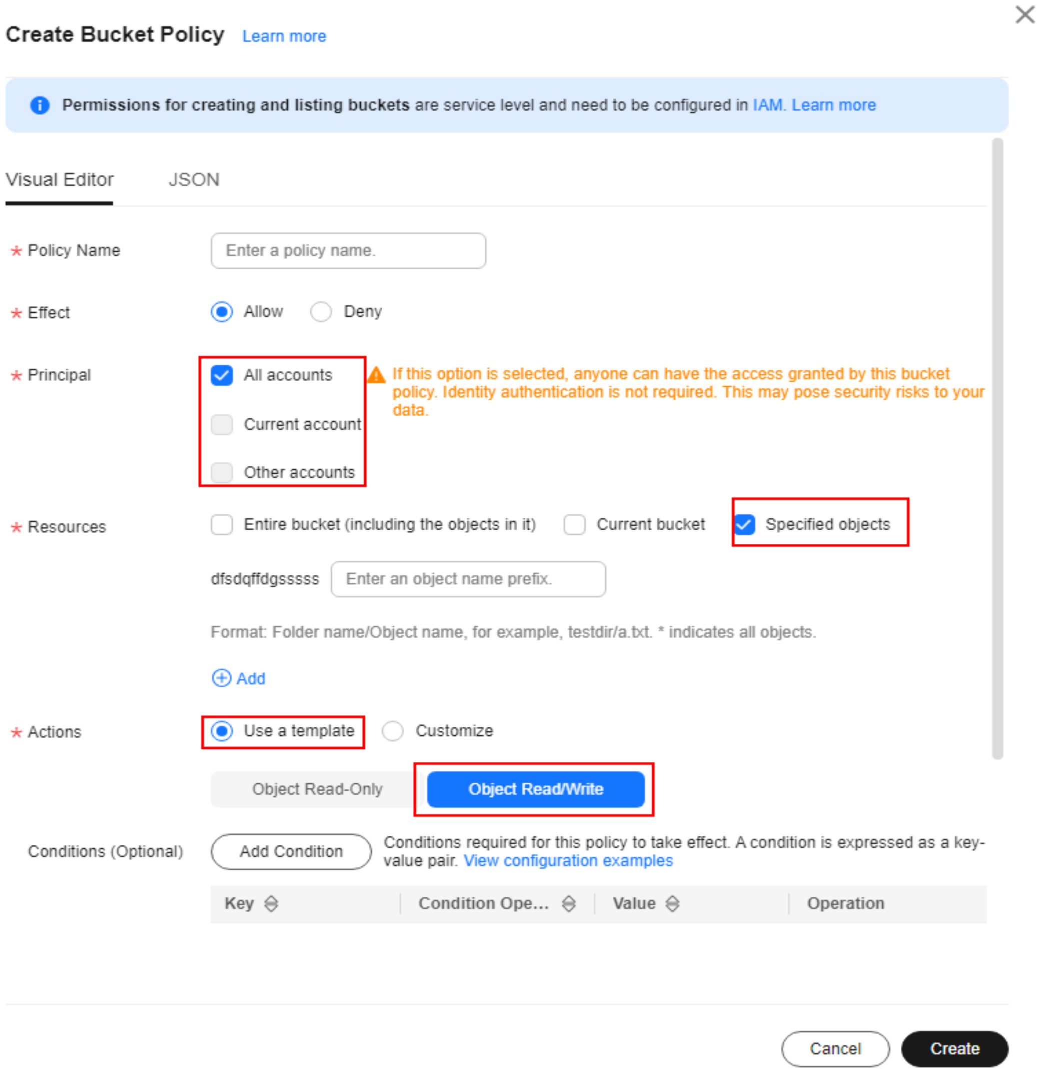
Object Read-Only | Object Read/Write

Conditions (Optional): Add Condition

Conditions required for this policy to take effect. A condition is expressed as a key-value pair. View configuration examples

Key	Condition Ope...	Value	Operation

Figure 5-9 Choosing the Object Read/Write template



Create Bucket Policy Learn more

Permissions for creating and listing buckets are service level and need to be configured in IAM. Learn more

Visual Editor | JSON

* Policy Name: Enter a policy name.

* Effect: ● Allow ○ Deny

* Principal: ☑ All accounts □ Current account □ Other accounts

If this option is selected, anyone can have the access granted by this bucket policy. Identity authentication is not required. This may pose security risks to your data.

* Resources: □ Entire bucket (including the objects in it) □ Current bucket ☑ Specified objects

dfsdqffdgsssss Enter an object name prefix.

Format: Folder name/Object name, for example, testdir/a.txt. * indicates all objects.

⊕ Add

* Actions: ● Use a template ○ Customize

Object Read-Only | **Object Read/Write**

Conditions (Optional): Add Condition

Conditions required for this policy to take effect. A condition is expressed as a key-value pair. View configuration examples

Key	Condition Ope...	Value	Operation

Table 5-2 Bucket policy templates

Principal	Resource	Template	Actions Allowed	Advanced Settings
All accounts	Entire bucket (including the objects in it)	Public Read See **Figure 5-2**	**Allows anonymous users to perform the following actions on a bucket and the objects in it:** HeadBucket (to check whether the bucket exists and obtain the bucket metadata) GetBucketLocation (to get the bucket location) GetObject (to obtain object content and metadata) RestoreObject (to restore objects from Archive storage) GetObjectVersion (to obtain the content and metadata of a specified object version)	Excluding the specified actions is not allowed.

Principal	Resource	Template	Actions Allowed	Advanced Settings
		Public Read/ Write See **Figure 5-3**	**Allows anonymous users to perform the following actions on a bucket and the objects in it:** ListBucket (to list objects in the bucket and obtain the bucket metadata) ListBucketVersions (to list object versions in the bucket) HeadBucket (to check whether the bucket exists and obtain the bucket metadata) GetBucketLocation (to get the bucket location) PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts) GetObject (to obtain object content and metadata) ModifyObjectMetaData (to modify object metadata) ListBucketMultipartUploads (to list multipart uploads) ListMultipartUploadParts (to list uploaded parts) AbortMultipartUpload (to abort multipart uploads) RestoreObject (to restore objects from Archive storage) GetObjectVersion (to obtain the content and metadata of a specified object version) PutObjectAcl (to configure the object ACL) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL)	Excluding the specified actions is not allowed.

Principal	Resource	Template	Actions Allowed	Advanced Settings
Current account/Other accounts/Delegated accounts	Entire bucket (including the objects in it)	Bucket Read-Only See **Figure 5-4**	**Allows specified accounts to perform the following actions on a bucket and the objects in it:** Get* (all GET actions) List* (all LIST actions) HeadBucket (to check whether the bucket exists and obtain the bucket metadata)	Excluding the specified actions is not allowed.
		Bucket Read/Write See **Figure 5-5**	**Allows specified accounts to perform all actions excluding the following ones on a bucket and the objects in it:** DeleteBucket (to delete the bucket) PutBucketPolicy (to configure a bucket policy) PutBucketAcl (to configure the bucket ACL)	The specified actions are excluded.

Principal	Resource	Template	Actions Allowed	Advanced Settings
All accounts/ Current account/ Other accounts/ Delegated accounts	Current bucket + Specified objects	Directory Read-Only See **Figure 5-6**	**Allows all accounts or specified accounts to perform the following actions on the current bucket and the specified resources in it:** GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL) RestoreObject (to restore objects from Archive storage) ListBucket (to list objects in the bucket and obtain the bucket metadata) ListBucketVersions (to list object versions in the bucket) HeadBucket (to check whether the bucket exists and obtain the bucket metadata) GetBucketLocation (to get the bucket location) **NOTE** If you apply the policy to **All accounts**, **ListBucket** and **ListBucketVersions** are not included in the template.	Excluding the specified actions is not allowed.

Principal	Resource	Template	Actions Allowed	Advanced Settings
		Directory Read/Write See **Figure 5-7**	**Allows all accounts or specified accounts to perform the following actions on the current bucket and the specified resources in it:** PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts) GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) ModifyObjectMetaData (to modify object metadata) ListBucketMultipartUploads (to list multipart uploads) ListMultipartUploadParts (to list uploaded parts) AbortMultipartUpload (to abort multipart uploads) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL) PutObjectAcl (to configure the object ACL) RestoreObject (to restore objects from Archive storage) ListBucket (to list objects in the bucket and obtain the bucket metadata) ListBucketVersions (to list object versions in the bucket) HeadBucket (to check whether the bucket exists and obtain the bucket metadata) GetBucketLocation (to get the bucket location)	Excluding the specified actions is not allowed.

Principal	Resource	Template	Actions Allowed	Advanced Settings
All accounts/ Current account/ Other accounts/ Delegated accounts	Specified objects	Object Read-Only See **Figure 5-8**	**Allows all accounts or specified accounts to perform the following actions on specified resources in the bucket:** GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL) RestoreObject (to restore objects from Archive storage)	Excluding the specified actions is not allowed.
		Object Read/ Write See **Figure 5-9**	**Allows all accounts or specified accounts to perform the following actions on specified resources in the bucket:** PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts) GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) ModifyObjectMetaData (to modify object metadata) ListMultipartUploadParts (to list uploaded parts) AbortMultipartUpload (to abort multipart uploads) GetObjectVersionAcl (to obtain the ACL of an object version) GetObjectAcl (to obtain the object ACL) PutObjectAcl (to configure the object ACL) RestoreObject (to restore objects from Archive storage)	Excluding the specified actions is not allowed.

Step 6 Complete the bucket policy configuration.

Some bucket policy templates require a configuration of principals or resources. You can also change the existing settings of a template, including the policy name, principals, resources, actions, and conditions. For details, see **Bucket Policy Parameters**.

Step 7 Click **Create** in the lower right corner.

----End

5.2.2 Creating a Custom Bucket Policy (Visual Editor)

You can also customize bucket policies based on your service needs. A custom bucket policy consists of five basic elements: effects, principals, resources, actions, and conditions. For details, see **Bucket Policy Parameters**.

Procedure

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

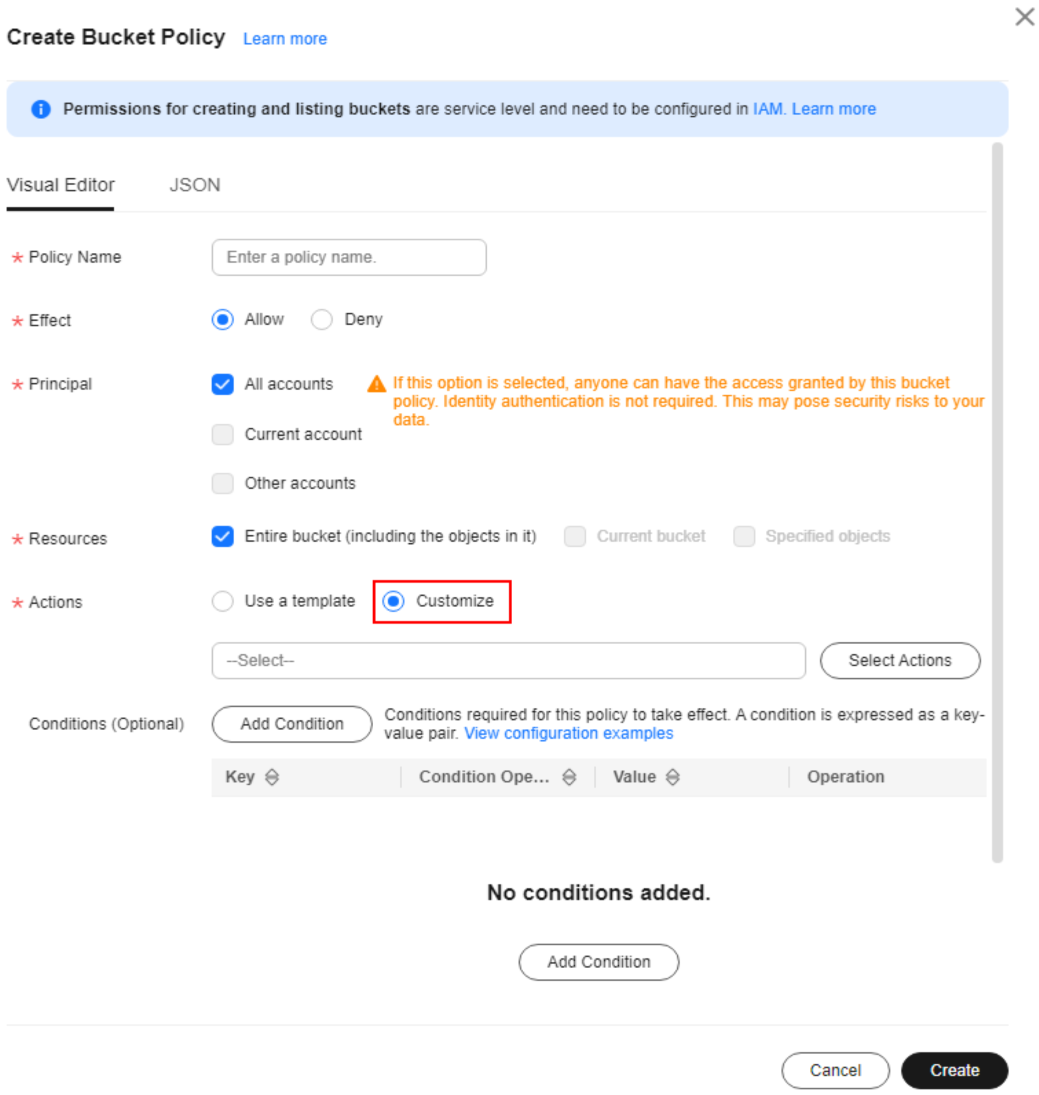
Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 4 Click **Create**.

Step 5 Configure a bucket policy.

Figure 5-10 Configuring a bucket policy



Permissions for creating and listing buckets are service level and need to be configured in IAM. Learn more

* Policy Name: Enter a policy name.

* Effect: ● Allow ○ Deny

* Principal: ☑ All accounts □ Current account □ Other accounts

If this option is selected, anyone can have the access granted by this bucket policy. Identity authentication is not required. This may pose security risks to your data.

* Resources: ☑ Entire bucket (including the objects in it) □ Current bucket □ Specified objects

* Actions: ○ Use a template ● Customize

--Select--

Conditions (Optional): Conditions required for this policy to take effect. A condition is expressed as a key-value pair. View configuration examples

Key	Condition Ope...	Value	Operation

No conditions added.

Table 5-3 Parameters for configuring a custom bucket policy

Parameter		Description
Method		Visual editor or JSON. The visual editor is used here. For details about configurations in the JSON view, see **Creating a Custom Bucket Policy (JSON View)**.
Policy Name		Enter a bucket policy name.
Policy content	Effect	- **Allow**: The policy allows the matched requests. - **Deny**: The policy denies the matched requests.

Parameter		Description
	Principals	- **All accounts**: The bucket policy applies to anonymous users. - **Current account**: Specify one or more IAM users under the current account. - **Other accounts**: Specify one or more accounts. **NOTE** The account ID and IAM user ID can be obtained from the **My Credentials** page. Accounts should be configured in the *Domain ID*/*IAM user ID* format, with each one on a separate line. *Account ID*/***** indicates that permission is granted to all IAM users under the account. - **Delegated accounts**: Delegated accounts can be added only after **Other accounts** is selected. **NOTE** Delegated accounts should be configured in the *ID of a delegating account*/*Agency name* format. Multiple delegated accounts are allowed, with each one on a separate line.
	Resources	- **Entire bucket (including the objects in it)**: The policy applies to the bucket and the objects in it. You can configure bucket and object actions in this policy. - **Current bucket**: The policy applies to the current bucket. You can configure bucket actions in this policy. - **Specified objects**: The policy applies to specified objects in the bucket. You can configure object actions in this policy. **NOTE** 1. Multiple resource paths can be specified. 2. A resource path should be configured in the *Folder name*/*Object name* format, for example, **testdir/a.txt**. To specify the **testdir** folder and all objects in it, enter **testdir/***. 3. You can specify a specific object, an object set, or a directory. ***** indicates all objects in the bucket. To specify a specific object, enter the object name. To specify a set of objects, enter *Object name prefix****, ****Object name suffix*, or *****. For example, **testdir/*** indicates objects in the **testdir** folder, and **testprefix*** indicates objects whose prefix is **testprefix**.

Parameter		Description
	Actions	● **Actions**: Choose **Customize**. ● **Select Actions**: See **Bucket Policy Parameters**. **NOTE** 1. If you select **Entire bucket (including the objects in it)** for **Resources**, common actions, bucket actions, and object actions will be available for you to choose from. 2. If you select **Current bucket** for **Resources**, common actions and bucket actions will be available for you to choose from. 3. If you select **Specified objects** for **Resources**, common actions and object actions will be available for you to choose from. 4. If you select both **Current bucket** and **Specified objects** for **Resources**, common actions, bucket actions, and object actions will be available for you to choose from.
	Conditions (Optional)	● **Key**: See **Bucket Policy Parameters**. ● **Conditional Operator**: See **Bucket Policy Parameters**. ● **Value**: The entered value is associated with the key.
	Advanced Settings > Exclude (Optional)	● **Specified principals**: By selecting this option, the bucket policy applies to users except the specified ones. **NOTE** If you do not select this option, the bucket policy applies to the specified users. ● **Specified resources**: By selecting this option, the bucket policy applies to resources except the specified ones. **NOTE** If you do not select this option, the bucket policy applies to the specified resources. ● **Specified actions**: By selecting this option, the bucket policy applies to actions except the specified ones. **NOTE** 1. If you do not select this option, the bucket policy applies to the specified actions. 2. By default, **Specified actions** is selected for **Exclude** in the bucket read/write template only. The action exclusion setting in bucket policy templates cannot be modified.

Step 6 Click **Create** in the lower right corner.

----End

5.2.3 Creating a Custom Bucket Policy (JSON View)

If you are familiar with the JSON syntax and OBS bucket policies, you can code a bucket policy in the JSON view. There is no limit on the number of bucket policies (statements) for a bucket, but the JSON descriptions of all bucket policies in a bucket cannot exceed 20 KB in total.

Ways to Create a Custom Bucket Policy

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to create a custom bucket policy.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 4 In the upper right corner of the page, click **JSON** and then **Edit**.

Step 5 Edit the bucket policy. Below gives a bucket policy example in JSON:

```
{
   "Statement":[
      {
         "Action":[
            "CreateBucket",
            "DeleteBucket"
         ],
         "Effect":"Allow",
         "Principal":{
            "ID":[
               "domain/account ID",
               "domain/account ID:user/User ID"
            ]
         },
         "Condition":{
            "NumericNotEquals":{
               "Referer":"sdf"
            },
            "StringNotLike":{
               "Delimiter":"ouio"
            }
         },
         "Resource":"000-02/key01"
      }
   ]
}
```

Table 5-4 Parameters for creating a bucket policy in JSON

Parameter	Description
Action	Actions the bucket policy applies to. For details, see **Bucket Policy Parameters**.

Parameter	Description
Effect	Effect of the bucket policy. For details, see **Bucket Policy Parameters**.
Principal	Users the bucket policy is applied to. You can obtain the user ID on the **My Credentials** page by logging in to the console as the user to be authorized. Principals should be configured as follows: - **domain/***Account ID* (indicating that the principal is an account) - **domain/***Account ID***:user/***User ID* (indicating that the principal is a user under an account)
Condition	Conditions under which the bucket policy takes effect. For details, see **Bucket Policy Parameters**.
Resource	Resources the bucket policy is applied to. For details, see **Bucket Policy Parameters**.

Step 6 Click **Create**.

----End

Using APIs

- **IAM Permissions Control**
- **Configuring a Bucket Policy**

Using SDKs

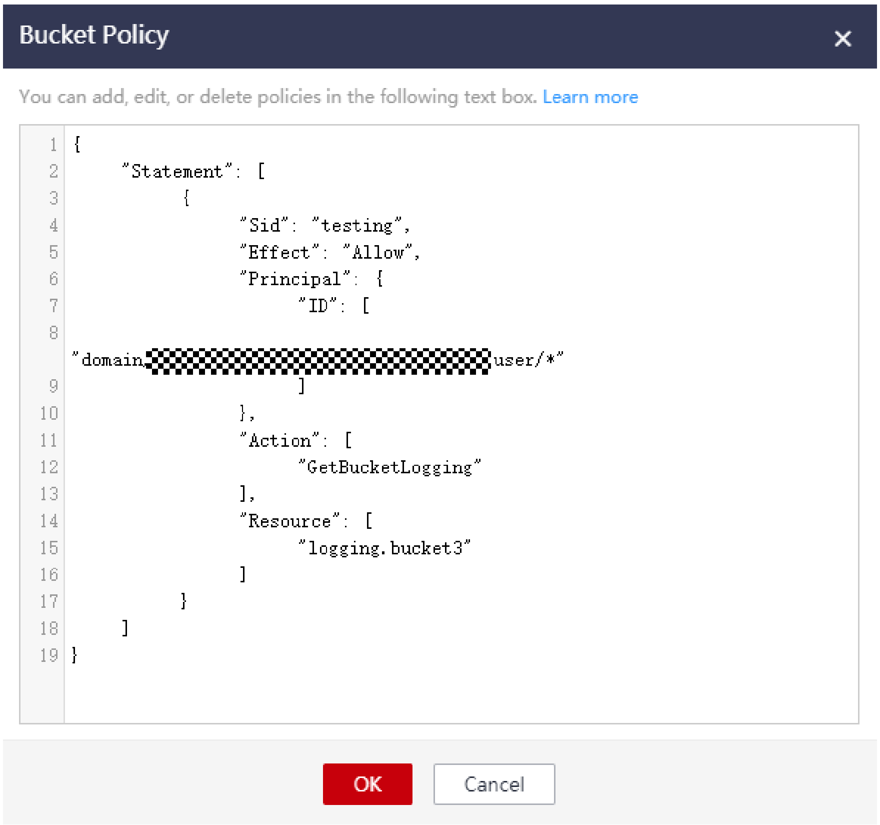
Java	**Python**	C: not supported	**Go**	**BrowserJS**	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**

Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Select the bucket you want and choose **More** > **Bucket Policy**. The window shown in **Figure 5-11** is displayed.

Figure 5-11 Configuring a bucket policy



You can add, edit, or delete policies in the following text box. Learn more

```
{
      "Statement": [
            {
                  "Sid": "testing",
                  "Effect": "Allow",
                  "Principal": {
                        "ID": [
"domain[redacted]user/*"
                        ]
                  },
                  "Action": [
                        "GetBucketLogging"
                  ],
                  "Resource": [
                        "logging.bucket3"
                  ]
            }
      ]
}
```

Step 3 Enter a bucket policy in the following format.

1. Grant permissions to an account. In the following example, the account (whose account ID is **783fc6652cf246c096ea836694f71855**) is granted the permission required to obtain the log management information about bucket **logging.bucket3**.

```
{
   "Statement": [
      {
         "Sid": "testing",
         "Effect": "Allow",
         "Principal": {
            "ID": [
               "domain/783fc6652cf246c096ea836694f71855:user/*"
            ]
         },
         "Action": [
            "GetBucketLogging"
         ],
         "Resource": [
            "logging.bucket3"
         ]
      }
   ]
}
```

   Table 5-5 describes the parameters that you need to manually modify in the example above:

Table 5-5 Parameter changes

Item to Modify	Description
GetBucketLogging	Value of the **Action** field that indicates all OBS-supported actions in the policy. The value is a case-insensitive string. The value can contain a wildcard character (*), for example, **"Action":["List*", "Get*"]**, to apply all actions to the resources. You need to change the value as needed. For the actions supported by OBS, see **Bucket-Related Actions**.
Allow	Value of the **Effect** field that indicates whether the permission in the policy is allowed or denied. The value must be **Allow** or **Deny**.
logging.bucket3	The bucket on which the policy works. You can change the bucket name as needed.
783fc6652cf246c096ea836694f71855	ID of an account. You can change it as needed. You can obtain the account ID on the bucket's **Basic Information** page.

2. Grant permissions to an IAM user. In the following example, the user (whose ID is **71f3901173514e6988115ea2c26d1999**) under the account (whose ID is **219d520ceac84c5a98b237431a2cf4c2**) is assigned the permission required to set log management for bucket **logging.bucket3**.

```
{
    "Statement": [
        {
            "Sid": "testing",
            "Effect": "Allow",
            "Principal": {
                "ID": [
                    "domain/219d520ceac84c5a98b237431a2cf4c2:user/
71f3901173514e6988115ea2c26d1999"
                ]
            },
            "Action": [
                "PutBucketLogging"
            ],
            "Resource": [
                "logging.bucket3"
            ]
        }
    ]
}
```

Table 5-6 describes the parameters that you need to manually modify in the example above:

Table 5-6 Parameter changes

Item to Modify	Description
PutBucketLogging	Value of the **Action** field that indicates all OBS-supported actions in the policy. The value is a case-insensitive string. The value can contain a wildcard character (*), for example, **"Action":["List*", "Get*"]**, to apply all actions to the resources. You need to change the value as needed. For the actions supported by OBS, see **Bucket-Related Actions**.
Allow	Value of the **Effect** field that indicates whether the permission in the policy is allowed or denied. The value must be **Allow** or **Deny**.
logging.bucket3	The bucket on which the policy works. You can change the bucket name as needed.
219d520ceac84c5a98b237431a2cf4c2	ID of an account. You can change it as needed. You can click ⓘ next to the target bucket to obtain the **Account ID** on the **Basic Information** page.
71f3901173514e6988115ea2c26d1999	ID of a user under the account. You can change it as needed. You can choose **My Credentials** from the username in the upper right corner of OBS Console to obtain the **IAM User ID**.

For details about bucket policy parameters, see **Bucket Policy Parameters**.

----End

Using the CLI Tool - obsutil

Command Line Structure

- In Windows

```
obsutil bucketpolicy obs://bucket -method=put -localfile=xxx [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

- In Linux or macOS

```
./obsutil bucketpolicy obs://bucket -method=put -localfile=xxx [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
```

Examples

- Take the Windows OS as an example. Run the **obsutil bucketpolicy obs://bucket -method=put -localfile=d:\temp\policy.json** command to set a bucket policy based on file **policy.json**.

```
obsutil bucketpolicy obs://bucket -method=put -localfile=d:\temp\policy.json

Put bucketPolicy succeed, requestId is [04050000016836C5DA6FB21F14A2A0C0]
```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
method	Mandatory	Specifies the method. Set this parameter to **put** when configuring a bucket policy.
localfile	Mandatory	Path of the local policy file to import
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

5.2.4 Replicating Bucket Policies

Scenarios

OBS allows you to replicate the existing bucket policies to a new bucket. When replicating policies, OBS automatically replaces the bucket name in the source bucket policies with the destination bucket name, for the policies to apply to the destination bucket.

Constraints

- The policies replicated from a source bucket will not overwrite existing policies in the destination bucket.
- The source policies with the same name as those in the destination bucket will not be replicated.
- Both source and destination buckets must be of the 3.0 version.

Procedure

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

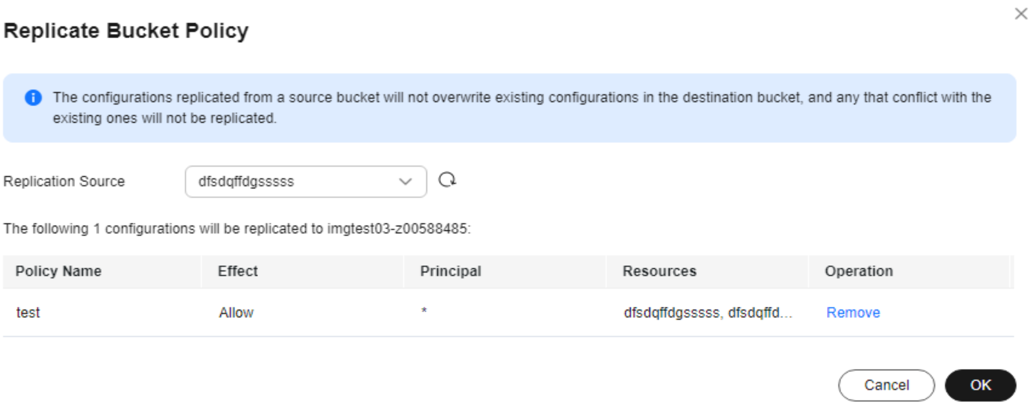
Step 3 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 4 Click **Replicate**.

Step 5 Select a replication source, which is the bucket whose policies you want to replicate.

After you select a replication source, all bucket policies with different name from those in the destination bucket are displayed. You can remove any that are not required.

Figure 5-12 Replicating bucket policies



Replicate Bucket Policy

The configurations replicated from a source bucket will not overwrite existing configurations in the destination bucket, and any that conflict with the existing ones will not be replicated.

Replication Source: dfsdqffdgsssss

The following 1 configurations will be replicated to imgtest03-z00588485:

Policy Name	Effect	Principal	Resources	Operation
test	Allow	*	dfsdqffdgsssss, dfsdqffd...	Remove

Step 6 Click **OK** to replicate the bucket policies to the destination bucket.

----End

5.3 Configuring an Object Policy

Object policies are applied to the objects in a bucket. With an object policy, you can configure conditions and actions for objects in a bucket.

Procedure

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the row containing the object for which you want to configure a policy, choose **More** > **Configure Object Policy** in the **Operation** column. The **Configure Object Policy** page is displayed.

You can customize a policy or use a preset template to configure one as needed.

- **Using a preset template**: The system has object policy templates preset for four typical scenarios. You can use these templates to quickly configure object policies. For details about each template, see **Bucket Policy Parameters**.
- **Customizing a policy**: You can also customize an object policy based on your needs. A custom object policy consists of five basic elements: effects, principals, resources, actions, and conditions, similar to a bucket policy. For details, see **Bucket Policy Parameters**. The resource is the selected object and is automatically configured by the system. For details about how to customize an object policy, see **Creating a Custom Bucket Policy (Visual Editor)**.
Different from customizing a bucket policy, to customize an object policy, you:
  a. Do not need to specify the resource.
  b. Can configure only object-related actions.

----End

5.4 Configuring a Bucket ACL

Prerequisites

You are the bucket owner or you have the permission to write the bucket ACL.

Ways to Configure a Bucket ACL

You can use OBS Console, APIs, SDKs, OBS Browser+, or obsutil to configure a bucket ACL.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

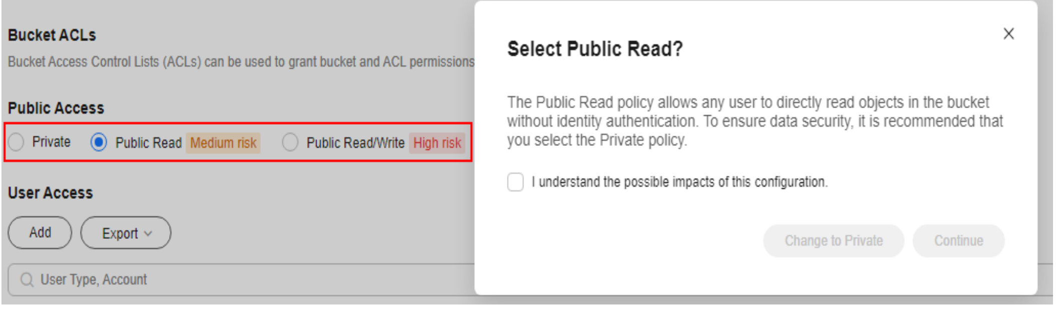
Step 3 In the navigation pane, choose **Permissions** > **Bucket ACLs**.

Step 4 On the **Bucket ACLs** page, choose a permission from **Private**, **Public Read**, and **Public Read/Write** to grant bucket ACL permission for anonymous users.

NOTE

1. After you change **Public Read** or **Public Read/Write** to **Private**, only the bucket owner or object owner has the access.
2. After you change **Private** to **Public Read**, anyone can read objects in the bucket. No identity authentication is required.
3. After you change **Private** to **Public Read/Write**, anyone can read, write, and delete objects in the bucket. No identity authentication is required.

Figure 5-13 Changing a public access permission



Bucket ACLs

Bucket Access Control Lists (ACLs) can be used to grant bucket and ACL permissions

Public Access

○ Private ● Public Read Medium risk ○ Public Read/Write High risk

User Access

User Type, Account

Select Public Read?

The Public Read policy allows any user to directly read objects in the bucket without identity authentication. To ensure data security, it is recommended that you select the Private policy.

□ I understand the possible impacts of this configuration.

Step 5 In the **Operation** column, click **Edit** to grant the owner, anonymous user, or log delivery user required ACL permissions for the bucket.

Step 6 In the middle of the page, click **Export** to get the bucket ACL configuration. The file includes the user type, account, bucket access, and ACL access.

Step 7 In the middle of the page, click **Add** to apply specific ACL permissions to an account.

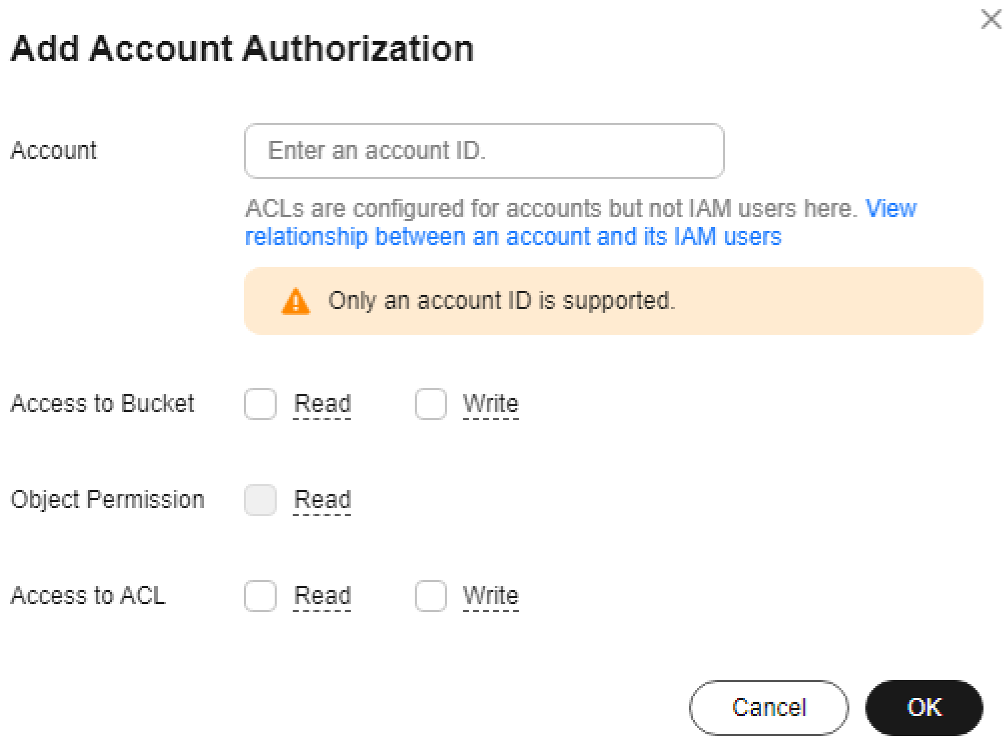
Enter an account ID and specify ACL permissions for the account. You can obtain the account ID from the **My Credentials** page.

Click **OK**.

NOTE

To select **Object read** for **Object Permission**, you must select **Read** for **Access to Bucket**.

Figure 5-14 Granting permissions



Add Account Authorization

Account: Enter an account ID.

ACLs are configured for accounts but not IAM users here. View relationship between an account and its IAM users

Only an account ID is supported.

Access to Bucket □ Read □ Write

Object Permission □ Read

Access to ACL □ Read □ Write

----End

Using APIs

Configuring a Bucket ACL

Using SDKs

Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

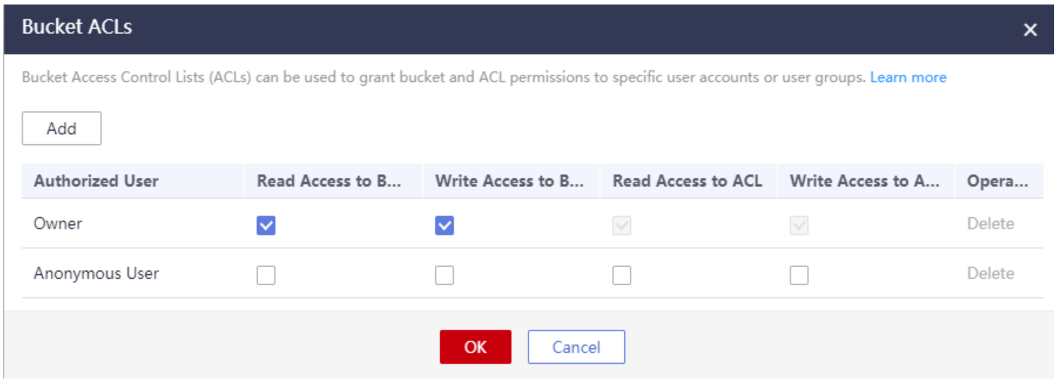
Using the GUI Tool - OBS Browser+

Step 1 Log in to OBS Browser+.

Step 2 Select the bucket you want and click **Bucket ACLs**.

Step 3 Configure the bucket ACL as needed and click **OK**, as shown in **Figure 5-15**.

Figure 5-15 Configuring an ACL



Bucket ACLs ×

Bucket Access Control Lists (ACLs) can be used to grant bucket and ACL permissions to specific user accounts or user groups. Learn more

Add

Authorized User	Read Access to B...	Write Access to B...	Read Access to ACL	Write Access to A...	Opera...
Owner	☑	☑	☑	☑	Delete
Anonymous User	☐	☐	☐	☐	Delete

OK Cancel

If no ACL permissions are configured for a new bucket, OBS Browser+ automatically disables the access to the bucket and its objects by any other users except the bucket owner.

----End

Using the CLI Tool - obsutil

Command Line Structure

- In Windows

```
obsutil chattri obs://bucket [-sc=xxx] [-acl=xxx] [-aclXml=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx]
[-t=xxx]
```

- In Linux or macOS

```
./obsutil chattri obs://bucket [-sc=xxx] [-acl=xxx] [-aclXml=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-
k=xxx] [-t=xxx]
```

Examples

- Take the Windows OS as an example. Run the **obsutil chattri obs://bucket-test -acl=private** command to change the access control policy of the bucket to private read and write.

```
obsutil chattri obs://bucket-test -acl=private
```

```
Set the acl of bucket [bucket-test] to [private] successfully, request id
[04050000016836C5DA6FB21F14A2A0C0]
```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
sc	Optional (additional parameter)	Default storage class of the bucket. Possible values are: ● **standard**: Standard storage class. It features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month) or data that is smaller than 1 MB. ● **warm**: Infrequent Access storage class. It is ideal for storing infrequently accessed (less than 12 times a year) data, but when needed, the access has to be fast. ● **cold**: Archive storage class. It provides secure, durable, and inexpensive storage for rarely-accessed (once a year) data. ● **deep-archive**: Deep Archive storage class (under limited beta testing). It is suitable for storing data that is barely (once every few years) accessed. This storage class costs less than the Archive storage class, but takes longer time (usually several hours) to restore data. **NOTE** If the multi-AZ mode is enabled for a bucket, the default storage class of the bucket cannot be set to **cold**.
acl	Optional (additional parameter)	Access control policies that can be specified for buckets. Possible values are: ● private ● public-read ● public-read-write **NOTE** The preceding three values indicate private read and write, public read, and public read and write.

<table>
<tr><th>Parameter</th><th>Optional or Mandatory</th><th>Description</th></tr>
<tr><td>aclXml</td><td>Optional (additional parameter)</td><td>Access control policy of the bucket, in XML format.
<AccessControlPolicy>
    <Owner>
        <ID><i>ownerid</i></ID>
    </Owner>
    <AccessControlList>
        <Grant>
            <Grantee>
                <ID><i>userid</i></ID>
            </Grantee>
            <Permission><i>[WRITE|WRITE_ACP|READ|READ_ACP|FULL_CONTROL]</i></Permission>
        </Grant>
        <Grant>
            <Grantee>
                <Canned>Everyone</Canned>
            </Grantee>
            <Permission><i>[WRITE|WRITE_ACP|READ|READ_ACP|FULL_CONTROL]</i></Permission>
        </Grant>
    </AccessControlList>
</AccessControlPolicy>
<b>NOTE</b>
● <b>Owner</b>: Optional. Specify the bucket owner's ID.
● In <b>AccessControlList</b>, the <b>Grant</b> field contains the authorized users. <b>Grantee</b> specifies the IDs of authorized users. <b>Canned</b> specifies the authorized user group (currently, only <b>Everyone</b> is supported).
● The following permissions can be granted: WRITE (write), WRITE_ACP (write ACL), READ (read), READ_ACP (read ACL), and FULL_CONTROL (full control).
<b>NOTICE</b>
Because angle brackets (<) and (>) are unavoidably included in the parameter value, you must use quotation marks to enclose them for escaping when running the command. Use single quotation marks for Linux or macOS and quotation marks for Windows.</td></tr>
<tr><td>config</td><td>Optional (additional parameter)</td><td>User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see <b>Configuration Parameters</b>.</td></tr>
<tr><td>e</td><td>Optional (additional parameter)</td><td>Specifies the endpoint.</td></tr>
<tr><td>i</td><td>Optional (additional parameter)</td><td>Specifies the user's AK.</td></tr>
</table>

Parameter	Optional or Mandatory	Description
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

NOTE

Only one from **sc**, **acl**, or **aclXml** can be set for each command.

Follow-up Procedure

After a specified account is granted the ACL permissions for a bucket, the authorized user can use the AK and SK to access that bucket by adding the bucket to OBS Browser+.

After certain permissions are granted to an anonymous user, the anonymous user can access the bucket without any authentication. The anonymous user can be either registered or non-registered. A registered anonymous user can use either of the methods above to access the bucket, while a non-registered anonymous user can access the bucket in any of the following ways:

- Access the bucket's domain name in a browser to view the objects in the bucket.
- Configure the bucket's domain name in a third-party system to directly connect to the bucket.

5.5 Configuring an Object ACL

Prerequisites

You are the object owner or you have the permission to write the object ACL.

An object owner is the account that uploads the object, but may not be the owner of the bucket that stores the object. For example, account **B** is granted the permission to access a bucket of account **A**, and account **B** uploads a file to the bucket. In that case, account **B**, instead of the bucket owner account **A**, is the owner of the object. By default, account A is not allowed to access this object and cannot read or modify the object ACL.

Procedure

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

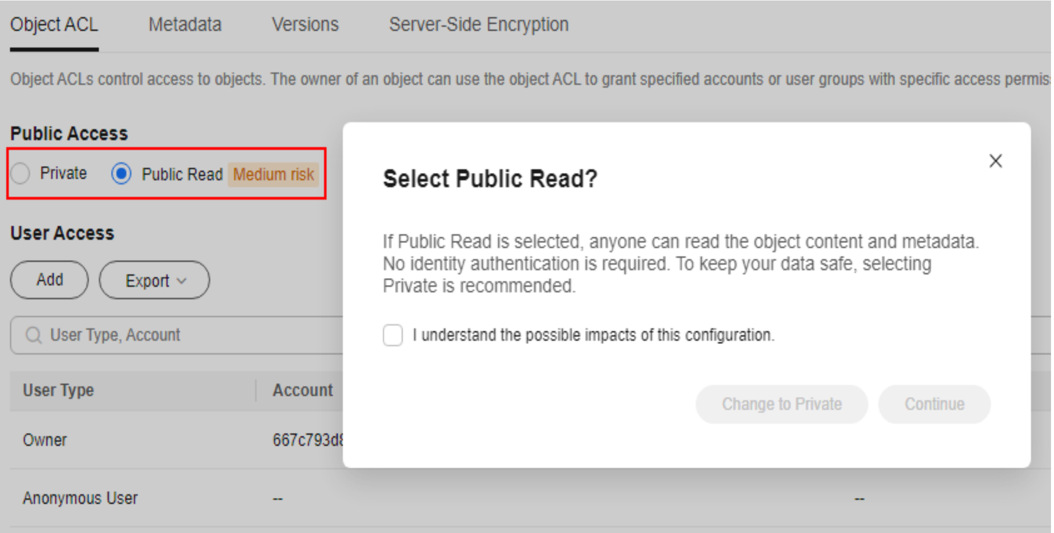
Step 3 Click a desired object.

Step 4 On the **Object ACL** page, choose a permission from **Private** and **Public Read** to grant object ACL permission for anonymous users.

NOTE

1. After you change **Public Read** to **Private**, only the bucket owner or object owner has the access.
2. After you change **Private** to **Public Read**, anyone can read the object content and metadata. No identity authentication is required.

Figure 5-16 Changing a public access permission



Object ACLs control access to objects. The owner of an object can use the object ACL to grant specified accounts or user groups with specific access permis

Public Access

○ Private ● Public Read Medium risk

Select Public Read?

If Public Read is selected, anyone can read the object content and metadata. No identity authentication is required. To keep your data safe, selecting Private is recommended.

□ I understand the possible impacts of this configuration.

Step 5 Click **Edit** to grant the owner, anonymous user, or other accounts required permissions for the object.

NOTE

ACL permissions for encrypted objects cannot be granted to registered users or anonymous users.

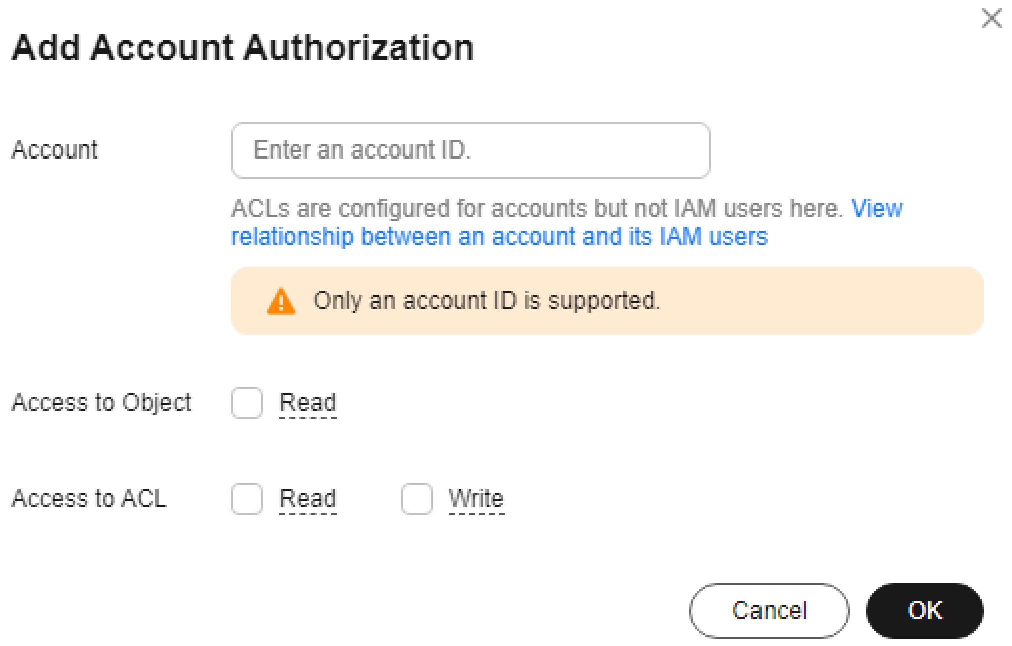
Click **Export** to get the object ACL configuration. The file includes the user type, account, object access, and ACL access.

Click **Add** to apply specific ACL permissions to an account.

Enter an account ID and specify ACL permissions for the account. You can obtain the account ID from the **My Credentials** page.

Click **OK**.

Figure 5-17 Granting permissions



Add Account Authorization

Account: Enter an account ID.

ACLs are configured for accounts but not IAM users here. View relationship between an account and its IAM users

Only an account ID is supported.

Access to Object: □ Read

Access to ACL: □ Read □ Write

----End

6 Data Security

6.1 Server-Side Encryption

You can use server-side encryption to encrypt data uploaded to OBS to improve storage security and compliance. With server-side encryption, the OBS server encrypts the objects uploaded from the client before storing them. When the objects are accessed, the OBS server decrypts the objects and then returns them to the client.

OBS supports three encryption methods: server-side encryption with KMS-managed keys (SSE-KMS), server-side encryption with OBS-managed keys (SSE-OBS), and server-side encryption with customer-provided keys (SSE-C).

Table 6-1 Three server-side encryption methods

Comparison Dimension	SSE-KMS	SSE-OBS	SSE-C
Scenarios	Scenarios that require high compliance and security. Keys are generated by third-party validated HSMs. Access to keys is controlled and all operations involving keys are traceable by logs.	Scenarios that require basic encryption and batch processing. OBS does not need to interact with KMS. SSE-OBS features lower access latency and better performance than SSE-KMS	Scenarios that users need to store and manage keys by themself
Key management	KMS generates and keeps keys, and OBS uses the keys to encrypt objects.	OBS generates and keeps keys, and uses the keys to encrypt objects.	A user generates and keeps keys, and OBS uses the keys to encrypt objects.

Comparison Dimension	SSE-KMS	SSE-OBS	SSE-C
Encryption algorithm	AES-256 and SM4	AES-256	AES-256
Encryption scope	Bucket level and object level	Bucket level and object level	Object level
Details	**SSE-KMS**	**SSE-OBS**	**SSE-C**

NOTE

- You can refer to the regions that support SSE-KMS on the console If you want to use the SM4 encryption algorithm when enabling SSE-KMS, choose the CN North-Ulanqab1 region.
- SSE-OBS is available on the entire Huawei Cloud Chinese Mainland website.
- SSE-C is available in the regions where the SSE-C API is supported.

Bucket-level and Object-level Server-Side Encryption

The following table compares bucket-level and object-level server-side encryption.

Table 6-2 Comparison between bucket-level and object-level server-side encryption

Item	Bucket-level Encryption	Object-level Encryption
Scenarios	You need to encrypt all objects when they are uploaded to a bucket.	You need to encrypt only some objects or select an encryption method and key each time you upload an object.
Encryption methods	SSE-KMS and SSE-OBS	SSE-KMS, SSE-OBS, and SSE-C
Configuration	You can change your encryption configuration at any time.	You can configure encryption during object upload and the configuration cannot be changed after the upload.

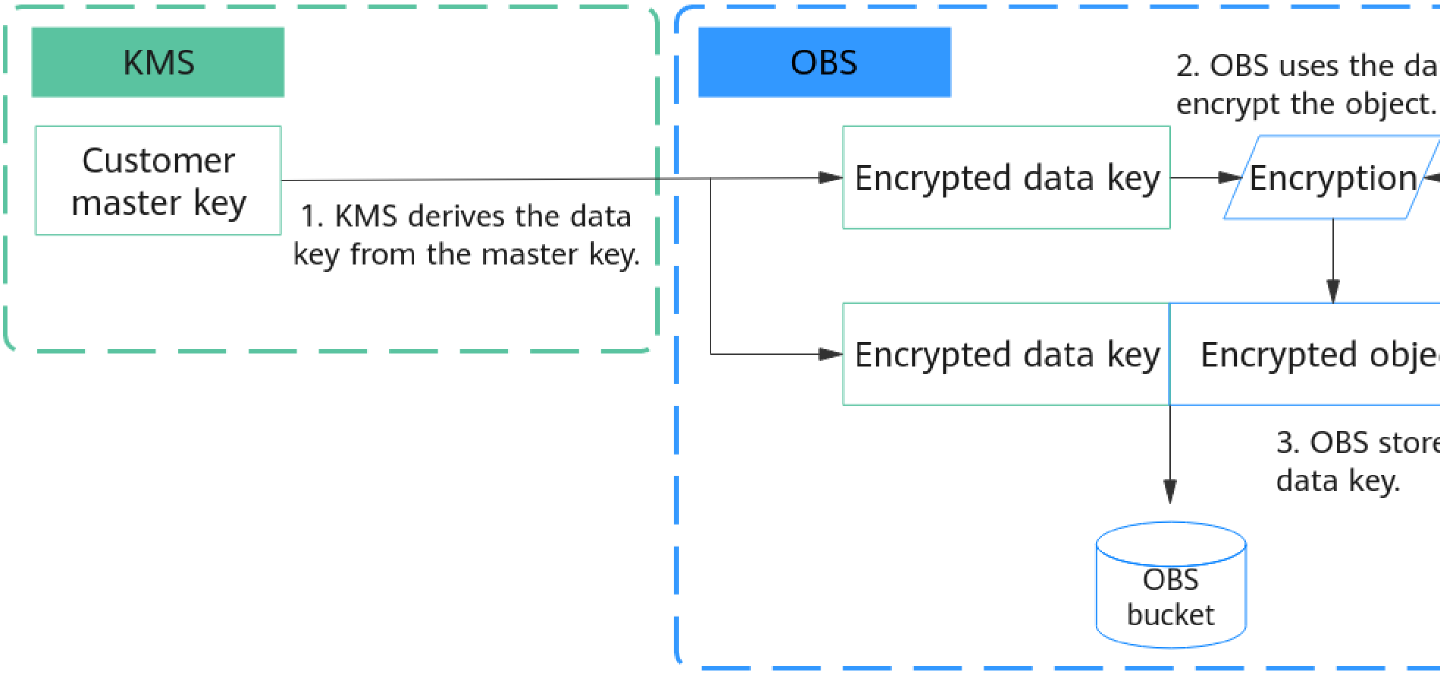
Item	Bucket-level Encryption	Object-level Encryption
Impacts	● If encryption is not enabled for a bucket, you can separately configure encryption when uploading objects to the bucket. ● If encryption is enabled for a bucket, OBS only encrypts the objects uploaded after encryption is enabled, and does not encrypt those uploaded before. When you upload objects to this bucket, they will inherit the bucket's encryption by default, but you can change the configuration if needed. ● If you change the encryption configuration for a bucket, only the objects uploaded after the change will inherit the new configuration. Objects uploaded before the change will retain the original configuration. ● Disabling encryption for a bucket does not change the encryption status of existing objects in the bucket. However, objects uploaded later will not be forcibly encrypted. If necessary, you can manually encrypt subsequent objects when you upload them to the bucket.	

Constraints

- Only one encryption method can be used each time an object is uploaded. The encryption configuration of an uploaded object cannot be changed.
- To use SSE-KMS to encrypt a bucket or the objects in it, you must have **kms:cmk:get**, **kms:cmk:list**, **kms:cmk:create**, **kms:dek:create**, and **kms:dek:crypto** permissions granted by using IAM, so that you can upload objects to or download objects from this bucket.
- If server-side encryption is disabled for a bucket, the encrypted objects can only be accessed over HTTPS.
- A key in use cannot be deleted, or the object encrypted with this key cannot be downloaded.

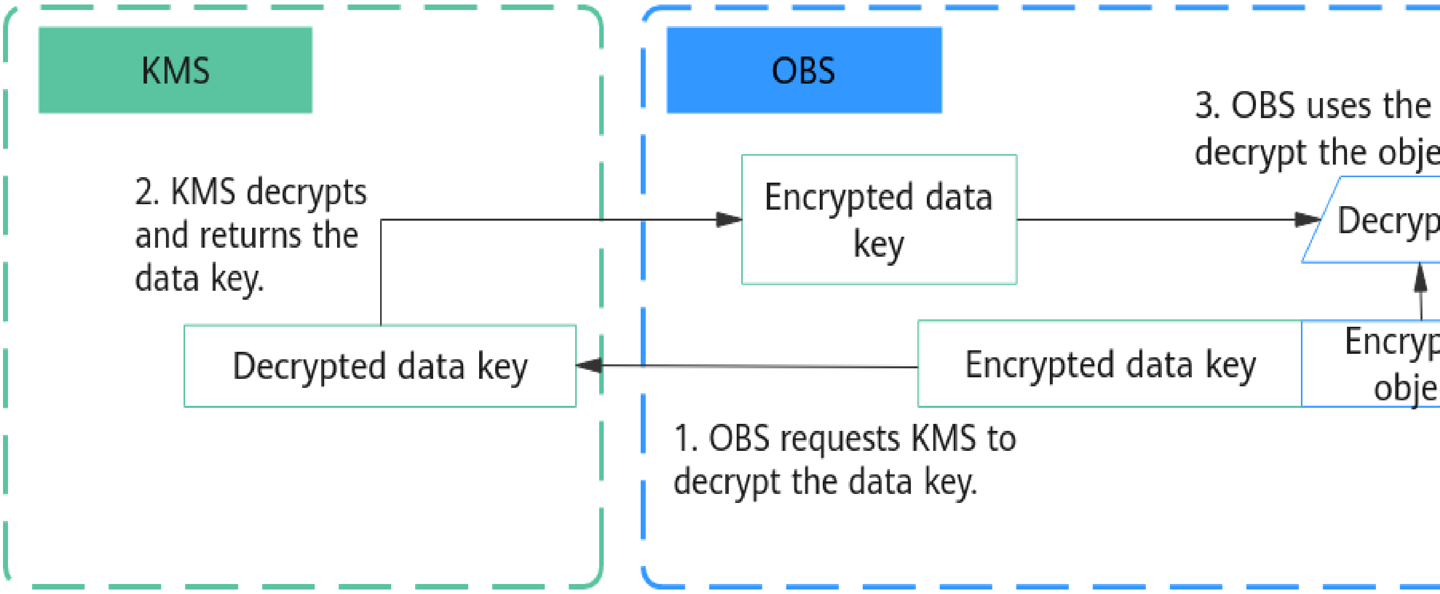
SSE-KMS

Figure 6-1 SSE-KMS encryption process



During the encryption in **Figure 6-1**, SSE-KMS supports envelope encryption. Customer master keys hosted by KMS are not directly used to encrypt data. Instead, data keys derived from them are used to encrypt data in OBS. OBS stores the encrypted objects and data keys.

Figure 6-2 SSE-KMS decryption process



During the decryption in **Figure 6-2**, KMS decrypts the data keys, uses them to decrypt the objects, and returns the objects to users.

KMS uses Hardware Secure Modules (HSMs) to ensure key security, enabling users to easily create and manage encryption keys. Keys are not displayed in plaintext outside HSMs, which prevents key disclosure. All operations performed on keys are controlled and logged, and usage of all keys is recorded, meeting regulatory compliance requirements.

Table 6-3 APIs that support SSE-KMS encryption

Resource	API	Header
Bucket	**Creating a Bucket**	● **x-obs-server-side-encryption**: Encryption method of a bucket ● **x-obs-server-side-data-encryption**: Encryption algorithm for server-side encryption ● **x-obs-server-side-encryption-kms-key-id**: KMS master key. If this header is not provided, the default master key will be used. ● **x-obs-sse-kms-key-project-id**: ID of the project that the KMS master key belongs to
	Configuring Bucket Encryption	The encryption configuration is in the body. For details, see **Configuring Bucket Encryption**.
Object	**Uploading Objects - PUT**	● **x-obs-server-side-data-encryption**: Encryption algorithm for server-side encryption ● **x-obs-server-side-encryption-kms-key-id**: KMS master key. If this header is not provided, the default master key will be used. ● **x-obs-sse-kms-key-project-id**: ID of the project that the KMS master key belongs to
	Uploading Objects - POST	
	Initiating a Multipart Upload	
	Copying Objects	

You can configure a bucket policy to limit the server-side encryption for a specific bucket to a specified method. For example, if you want people to use PUT to upload objects to bucket **ExampleBucket** and encrypt them with SSE-KMS, you can use the following bucket policy to deny any upload requests that do not contain the **x-obs-server-side-encryption:"kms"** header:

```
{
   "Statement": [
      {
         "Sid": "DenyUnEncryptedObjectUploads",
         "Effect": "Deny",
         "Principal": "*",
            "Action": "PutObject",
            "Resource": "ExampleBucket/*",
            "Condition": {
                "StringNotEquals": {
                    "x-obs-server-side-encryption": "kms"
                }
            }
        }
    ]
}
```

SSE-OBS

SSE-OBS uses a data key derived from the OBS root key to encrypt and decrypt data. APIs related to SSE-OBS encryption

Table 6-4 APIs related to SSE-OBS encryption

Resource	API	Header
Bucket	**Creating a Bucket**	**x-obs-server-side-encryption**: Encryption method of a bucket
	Configuring Bucket Encryption	The encryption configuration is in the body. For details, see **Configuring Bucket Encryption**.
Object	**Uploading Objects - PUT**	**x-obs-server-side-encryption**: Encryption method of a bucket
	Uploading Objects - POST	
	Initiating a Multipart Upload	
	Copying Objects	

You can configure a bucket policy to restrict the request headers for a specified bucket. For example, if you require that object upload requests do not contain header **x-obs-server-side-encryption:"AES256"**, you can use the following bucket policy:

```
{
    "Statement": [
        {
            "Sid": "DenyUnEncryptedObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "PutObject",
            "Resource": "YourBucket/*",
            "Condition": {
                "StringNotEquals": {
                    "x-obs-server-side-encryption": "AES256"
                }
            }
        }
    ]
}
```

SSE-C

With SSE-C, OBS uses the keys and MD5 values provided by customers for server-side encryption.

Table 6-5 APIs that support SSE-C encryption

Resource	API	Header
Object level	**Uploading Objects - PUT**	- **x-obs-server-side-encryption-customer-algorithm**: Algorithm used to encrypt an object - **x-obs-server-side-encryption-customer-key**: Key used to encrypt an object. The header value is a 256-bit, base64-encoded key. - **x-obs-server-side-encryption-customer-key-MD5**: MD5 value of the key used for encrypting an object
	Uploading Objects - POST	
	Initiating a Multipart Upload	
	Multipart Upload	
	Querying Object Metadata	
	Downloading Objects	

Resource	API	Header
	Uploading a Part of an Object - Copy Copying Objects	- If the source part or object is not encrypted but you want to encrypt their copy, use the following three headers: - **x-obs-server-side-encryption-customer-algorithm**: Algorithm used to encrypt an object - **x-obs-server-side-encryption-customer-key**: Key used to encrypt an object. The header value is a 256-bit, base64-encoded key. - **x-obs-server-side-encryption-customer-key-MD5**: MD5 value of the key used for encrypting an object If the source part or object has been encrypted and their copy also requires encryption, use the following headers to decrypt the source: - **x-obs-copy-source-server-side-encryption-customer-algorithm**: Algorithm used to decrypt the source object - **x-obs-copy-source-server-side-encryption-customer-key**: Key used to decrypt the source object. The header value is a 256-bit, base64-encoded key. - **x-obs-copy-source-server-side-encryption-customer-key-MD5**: MD5 value of the key used for decrypting the source object

Enabling and Configuring Server-Side Encryption for a Bucket

You can use OBS Console, APIs, or SDKs to configure Server-Side Encryption for a bucket.

Using OBS Console

The following describes how to enable server-side encryption for an existing bucket. For details about how to enable server-side encryption when creating a bucket, see **Creating a Bucket**.

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Overview**.

Step 4 In the **Basic Configurations** area, click **Server-Side Encryption**. The **Server-Side Encryption** dialog box is displayed.

Step 5 Choose **SSE-KMS** or **SSE-OBS**.

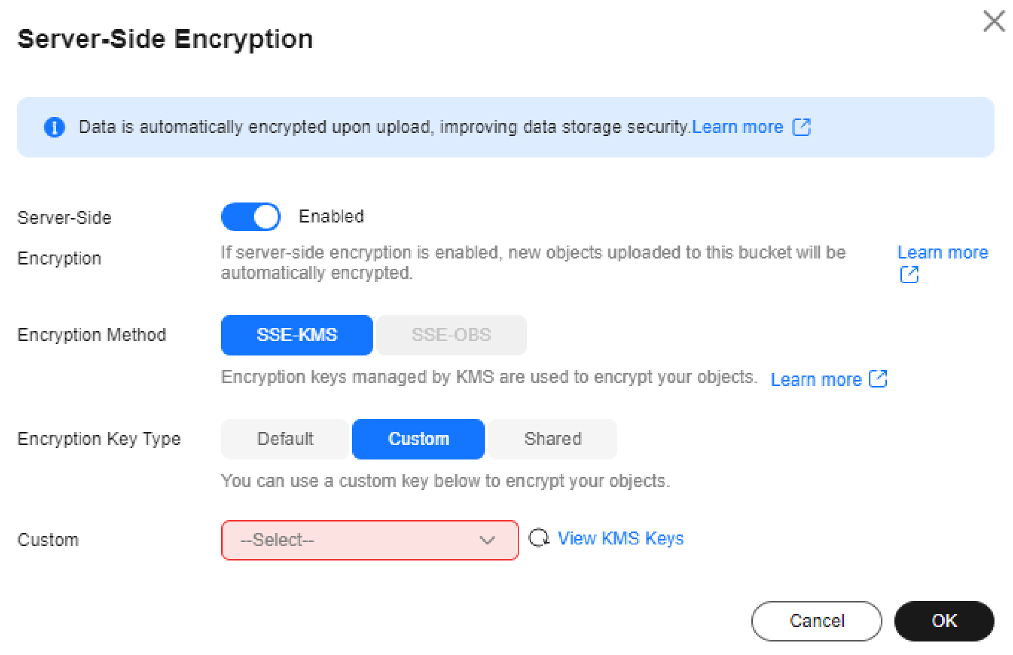
Choose **SSE-KMS**.

- You can choose **Default** to use the default key in the current region to encrypt the objects you upload to the bucket. If you do not have a default key, OBS automatically creates one the first time you upload an object.
- You can also choose **Custom** to use a custom key for encryption. If there is no custom key available, click **View KMS Keys** to create one.
- You can also select **Shared Key** to enter a shared key ID. The key shared by other users will be used to encrypt your objects. For details about how to obtain a shared key ID, see **Viewing a CMK**.

NOTE

A shared key from **a project or a subproject** can be configured here. However, if a shared key from a subproject is specified, the owner of the shared key cannot access objects encrypted with that key, but the bucket owner can.

Figure 6-3 Choosing SSE-KMS for a bucket



Server-Side Encryption

Data is automatically encrypted upon upload, improving data storage security.Learn more

Server-Side Encryption: Enabled
If server-side encryption is enabled, new objects uploaded to this bucket will be automatically encrypted. Learn more

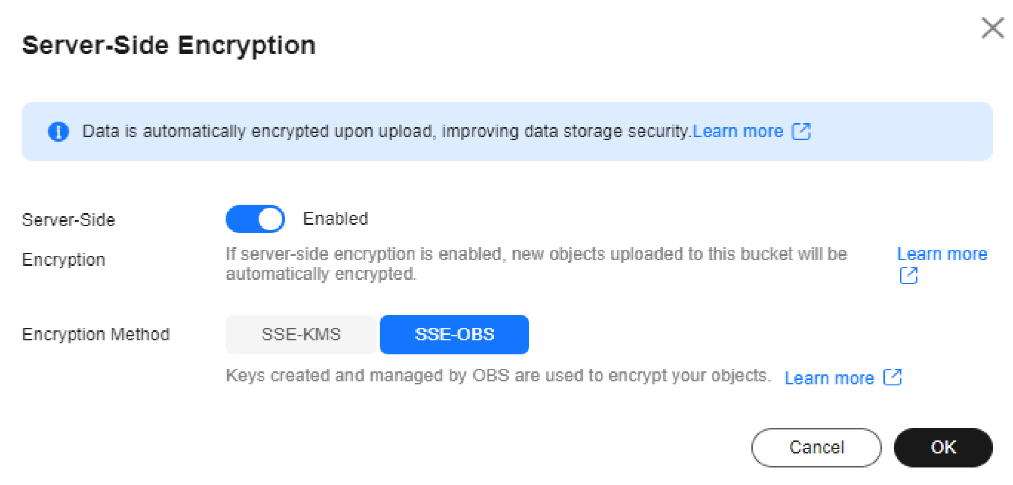
Encryption Method: SSE-KMS | SSE-OBS
Encryption keys managed by KMS are used to encrypt your objects. Learn more

Encryption Key Type: Default | Custom | Shared
You can use a custom key below to encrypt your objects.

Custom: --Select-- View KMS Keys

When **SSE-OBS** is chosen, the keys created and managed by OBS are used for encryption.

Figure 6-4 Choosing SSE-OBS for a bucket



Server-Side Encryption

Data is automatically encrypted upon upload, improving data storage security.Learn more

Server-Side Encryption: Enabled

If server-side encryption is enabled, new objects uploaded to this bucket will be automatically encrypted. Learn more

Encryption Method: SSE-KMS [✓ SSE-OBS]

Keys created and managed by OBS are used to encrypt your objects. Learn more

Step 6 Click **OK**.

----End

Using APIs

SSE-KMS

SSE-C

Configuring Bucket Encryption

Using SDKs

Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

Configuring Object-Level Server-Side Encryption

You can use OBS Console, APIs, or SDKs to encrypt objects to be uploaded.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

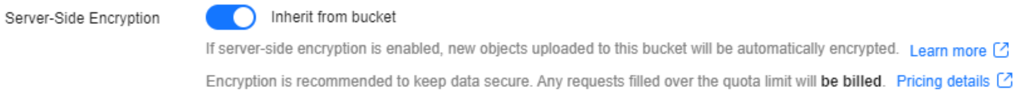
Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 Click **Upload Object**. The **Upload Object** dialog box is displayed.

Step 4 Add the files to be uploaded.

Step 5 Enable **Inherit from bucket** if your bucket has server-side encryption enabled and the objects in it need to inherit the encryption configuration of the bucket.

Figure 6-5 Inherit from bucket



Server-Side Encryption: Inherit from bucket

If server-side encryption is enabled, new objects uploaded to this bucket will be automatically encrypted. Learn more

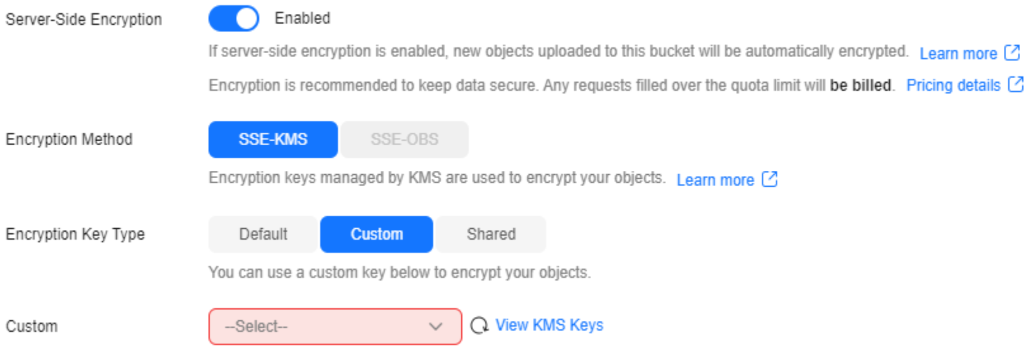
Encryption is recommended to keep data secure. Any requests filled over the quota limit will **be billed**. Pricing details

If your bucket has server-side encryption disabled or it has server-side encryption enabled but you need to a different encryption configuration for its objects, you can select **SSE-KMS** or **SSE-OBS** based on service requirements. If SSE-KMS is selected, you need to select an encryption key type.

Table 6-6 Parameters related to server-side encryption

Parameter	Value	
Encryption Key Type	Default	The default key of the current region will be used to encrypt your objects. If there is no such a default key, OBS creates one the first time you upload an object.
	Custom	You can click **View KMS Keys** to switch to the Data Encryption Workshop (DEW) console to create a key. Then go back here to select the key from the drop-down list.
	Shared	Enter the shared key ID. Your uploaded object will be encrypted using the key shared by other users. For details about how to obtain a shared key ID, see **Viewing a CMK**.

Figure 6-6 Configuring encryption for an object in a bucket with server-side encryption disabled



Server-Side Encryption: Enabled

If server-side encryption is enabled, new objects uploaded to this bucket will be automatically encrypted. Learn more

Encryption is recommended to keep data secure. Any requests filled over the quota limit will **be billed**. Pricing details

Encryption Method: SSE-KMS

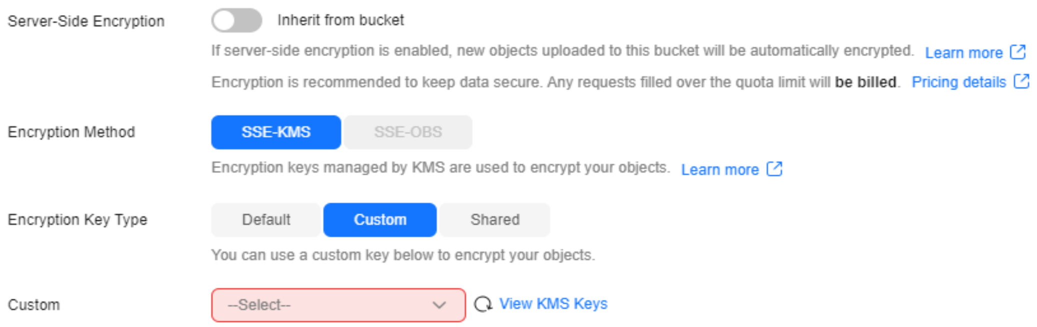
Encryption keys managed by KMS are used to encrypt your objects. Learn more

Encryption Key Type: Custom

You can use a custom key below to encrypt your objects.

Custom: --Select--

Figure 6-7 Configuring encryption for an object in a bucket with server-side encryption enabled



Server-Side Encryption: Inherit from bucket

If server-side encryption is enabled, new objects uploaded to this bucket will be automatically encrypted. Learn more

Encryption is recommended to keep data secure. Any requests filled over the quota limit will **be billed**. Pricing details

Encryption Method: SSE-KMS | SSE-OBS

Encryption keys managed by KMS are used to encrypt your objects. Learn more

Encryption Key Type: Default | Custom | Shared

You can use a custom key below to encrypt your objects.

Custom: --Select-- View KMS Keys

Step 6 Click **Upload**.

After the object is uploaded, you can view its encryption status on its details page.

----End

Using APIs

SSE-KMS

SSE-C

Using SDKs

Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

6.2 WORM

You can determine whether to enable WORM when creating a bucket. For details, see **Creating a Bucket**. When creating a bucket, if you enable WORM, you can continue to configure WORM after the bucket is created; if you do not enable it, you are not allowed to enable or configure it for that bucket later.

The following describes how to configure WORM retention after you create a bucket with WORM enabled.

OBS provides write-once-read-many (WORM) to protect objects from being deleted or tampered with within a specified period. WORM works at both the bucket and object levels in compliance mode.

Scenarios

In compliance mode, a WORM-protected object version cannot be overwritten or deleted by anyone, including the root user in your account.

When WORM is configured for a bucket, the protection applies to all objects in the bucket. When WORM is configured for an object version, the protection applies to the current object version only. No matter which type of WORM protection you want to use, you must enable WORM for the bucket first. A bucket-level WORM retention policy is applied only to objects uploaded after the policy was configured. If an object is protected by a bucket-level WORM policy and an object-level WORM policy at the same time, the object-level WORM policy takes precedence.

Precautions

- When you enable WORM for a bucket, OBS automatically enables versioning and versioning cannot be suspended later for that bucket. WORM protects objects based on the object version IDs. Only object versions with any WORM retention policy configured can be protected. Assume that object **test.txt 001** is protected by WORM. If another file with the same name is uploaded, a new object version **test.txt 002** with no WORM policy configured will be generated. In such case, **test.txt 002** is not protected and can be deleted. When you download an object without specifying a version ID, the current object version (**test.txt 002**) will be downloaded.
- A lifecycle rule cannot delete WORM-protected objects, but can transition their storage class. After an object is no longer protected, it will be deleted when meeting the expiration rule in a lifecycle configuration.
- If you do not enable WORM when creating a bucket, you cannot enable or configure it for that bucket later. If you cannot configure WORM for a bucket, it may be because you did not enable WORM when you created the bucket or your bucket was created before this feature was released. In such case, to use WORM, you need to create a new bucket and enable WORM for it.
- Once you enable WORM for a bucket, you cannot disable it or suspend versioning for the bucket, but you can disable the default WORM policy for the bucket.
- Buckets with WORM enabled do not support cross-region replication.
- If you have deregistered your account or your account has been frozen, the WORM-protected objects will be permanently deleted.
- WORM-based protection is not available for migration.
- The metadata of a WORM-protected object can still be modified.

Prerequisites

You have enabled WORM for the bucket when you create it.

Configuring WORM for a Bucket

You can use OBS Console or APIs to configure WORM for a bucket.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Overview**.

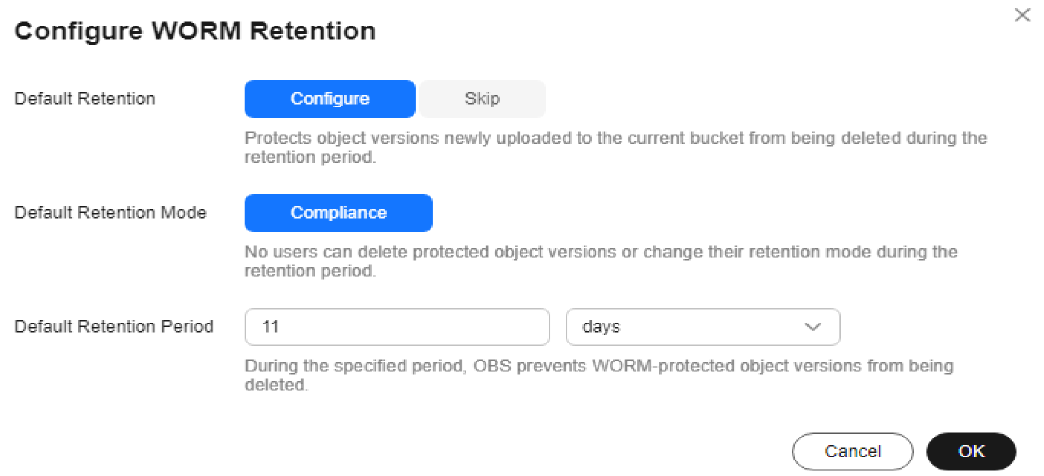
Step 4 In the **Basic Configurations** area, click **WORM Retention**. The **Configure WORM Retention** dialog box is displayed.

Step 5 Choose **Configure** and specify a default retention period. The default retention mode is **Compliance**.

NOTE

- Only the compliance retention mode is currently supported. In this mode, no users can delete protected object versions or change their retention mode during the specified retention period.
- During the specified default retention period, OBS prevents WORM-protected object versions from being deleted. You can configure a retention period in either days (from **1** to **36500**) or years (from **1** to **100**). The upper limit is 100 years.
- When you upload an object to a WORM-protected bucket, the object inherits the WORM retention from the bucket by default. You can also configure a different WORM retention for the object under advanced settings. If both a bucket-level and object-level WORM retention policy are applied to an object, the object-level retention policy will be used.

Figure 6-8 Configuring a WORM retention policy



Configure WORM Retention

Default Retention: Configure | Skip

Protects object versions newly uploaded to the current bucket from being deleted during the retention period.

Default Retention Mode: Compliance

No users can delete protected object versions or change their retention mode during the retention period.

Default Retention Period: 11 days

During the specified period, OBS prevents WORM-protected object versions from being deleted.

Step 6 Click **OK**.

----End

Using the API

Creating a Bucket

Configuring a Default WORM Policy for a Bucket

Obtaining the Default WORM Policy of a Bucket

Configuring WORM Retention for an Object

Querying Object Metadata that helps obtain the object-level WORM retention configuration

Skipping the WORM Retention Configuration

Step 1 In the navigation pane of OBS Console, choose **Object Storage**.

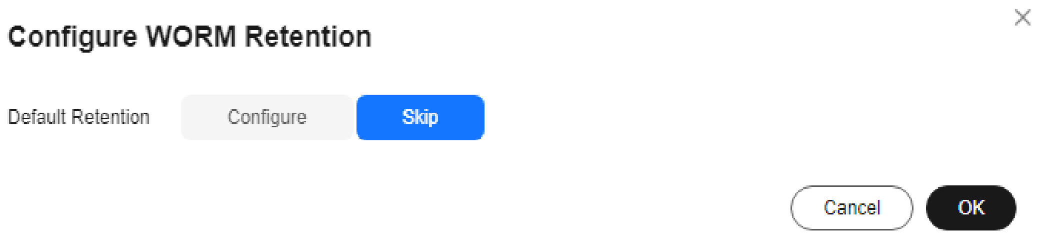
Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Overview**.

Step 4 In the **Basic Configurations** area, click **WORM Retention**. The **Configure WORM Retention** dialog box is displayed.

Step 5 Select **Skip** and click **OK**.

Figure 6-9 Skipping the WORM retention configuration



Configure WORM Retention

Default Retention: Configure | Skip

----End

Extending the Retention Period

After WORM is configured for an object, you can go to the object details page and extend the retention period of an object version on the **Versions** page. Before the specified date, OBS prevents protected object versions from being deleted.

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

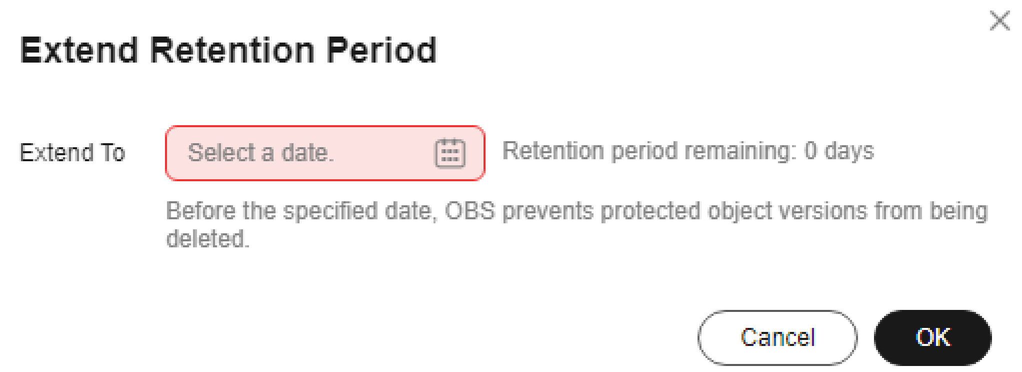
Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the object list, click the object you want to go to the object details page.

Step 4 On the **Versions** tab page, view all versions of the object.

Step 5 Locate the object version for which you want to extend the retention period, choose **More** > **Extend Retention Period**, and select a date.

Figure 6-10 Extending the retention period



Extend Retention Period

Extend To: Select a date. Retention period remaining: 0 days

Before the specified date, OBS prevents protected object versions from being deleted.

NOTE

A retention period can only be extended, but not shortened.

Assume that an object version was configured to be protected until March 30, 2023. If you want to extend the retention period on March 1, 2023, you can extend it to March 31, 2023 or a later date. If you extend the retention period on April 1, 2023, you can extend it to the current day (April 1, 2023) or a later date. If the current day is used, the object version will no longer be protected by WORM after 24:00 on that day.

----End

Manually and Permanently Deleting Objects from a WORM-Enabled Bucket

In a WORM-enabled bucket, if an object has no retention policy configured or its retention policy has expired, you can delete a desired version of the object. If an object version is within the retention period, it cannot be deleted.

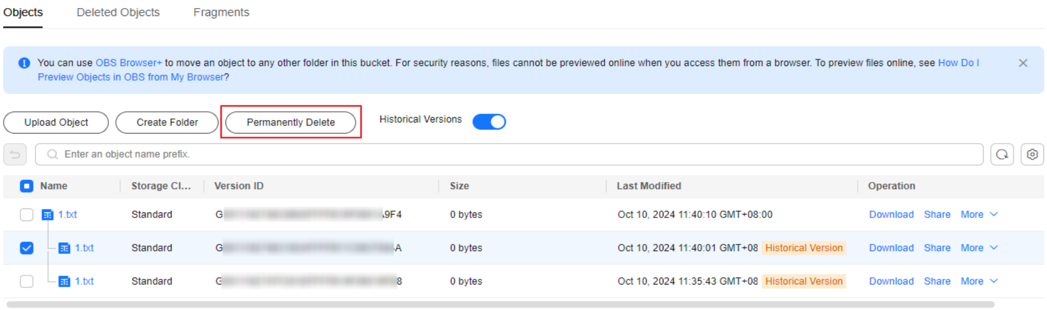
Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 Enable **Historical Versions**.

Step 4 Select the object version to be permanently deleted and click **Permanently Delete** above the search bar.

Figure 6-11 Permanently deleting an object version



You can use OBS Browser+ to move an object to any other folder in this bucket. For security reasons, files cannot be previewed online when you access them from a browser. To preview files online, see How Do I Preview Objects in OBS from My Browser?

☐	Name	Storage Cl...	Version ID	Size	Last Modified		Operation
☐	1.txt	Standard	G[redacted]9F4	0 bytes	Oct 10, 2024 11:40:10 GMT+08:00		Download Share More
☑	1.txt	Standard	G[redacted]A	0 bytes	Oct 10, 2024 11:40:01 GMT+08	Historical Version	Download Share More
☐	1.txt	Standard	G[redacted]8	0 bytes	Oct 10, 2024 11:35:43 GMT+08	Historical Version	Download Share More

Step 5 Click **OK**.

----End

Using a Lifecycle Rule to Delete Objects from a WORM-Enabled Bucket

You can configure a lifecycle rule to let OBS automatically expire and delete objects in a WORM enabled bucket. To realize this, the objects must have no retention policies configured or their retention policies have expired. If the objects are within their retention period, they cannot be deleted.

NOTE

In a WORM-enabled bucket, folders cannot be permanently deleted from the **Deleted Objects** list. To permanently delete a folder, you can only configure a lifecycle rule.

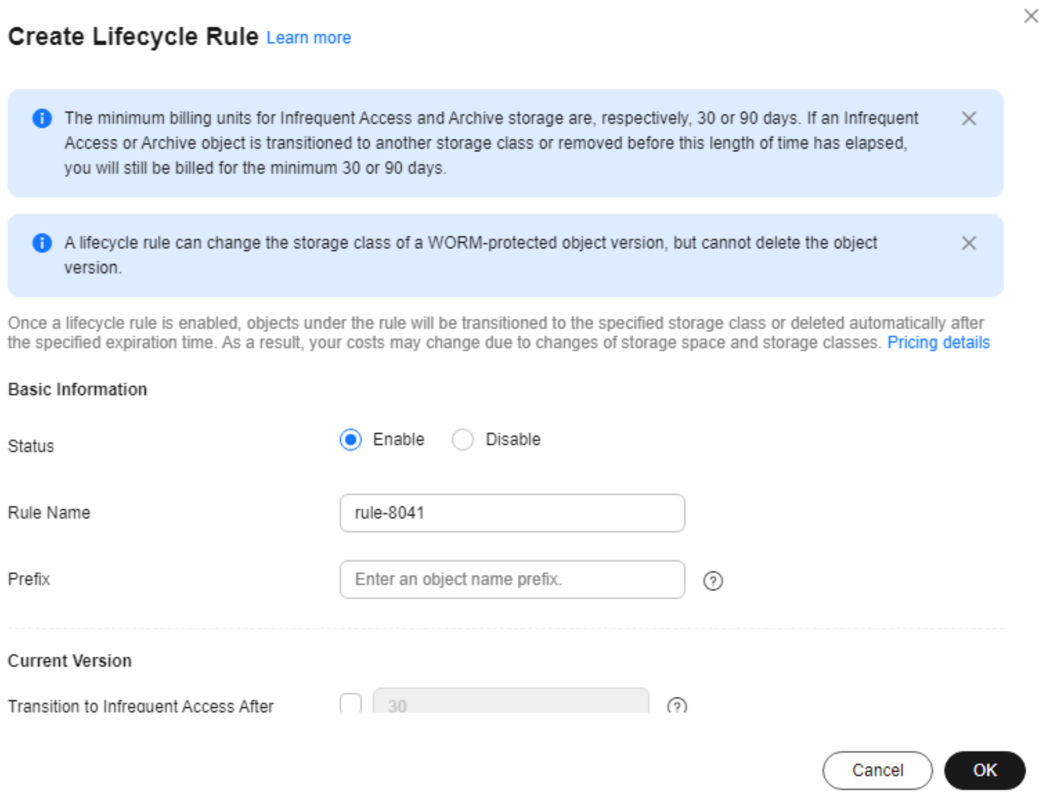
Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Basic Configurations** > **Lifecycle Rules**.

Step 4 Click **Create**.

Figure 6-12 Creating a lifecycle rule



Create Lifecycle Rule Learn more

The minimum billing units for Infrequent Access and Archive storage are, respectively, 30 or 90 days. If an Infrequent Access or Archive object is transitioned to another storage class or removed before this length of time has elapsed, you will still be billed for the minimum 30 or 90 days.

A lifecycle rule can change the storage class of a WORM-protected object version, but cannot delete the object version.

Once a lifecycle rule is enabled, objects under the rule will be transitioned to the specified storage class or deleted automatically after the specified expiration time. As a result, your costs may change due to changes of storage space and storage classes. Pricing details

Basic Information

Status ● Enable ○ Disable

Rule Name rule-8041

Prefix Enter an object name prefix.

Current Version

Transition to Infrequent Access After □ 30

Step 5 Configure a lifecycle rule.

Configure parameters under **Basic Information**:

- **Status**: Select **Enable** to enable this lifecycle rule after the configuration.
- **Rule Name**: It identifies a lifecycle rule. The rule name must be no longer than 255 characters.
- **Prefix**: It is optional.
  - If this field is configured, objects with the specified prefix will be managed by the lifecycle rule. The prefix cannot start with a slash (/) or contain two consecutive slashes (//), and cannot contain the following special characters: \:*?"<>|
  - If this field is not configured, all objects in the bucket will be managed by the lifecycle rule.

Configure parameters under **Current Version** or **Historical Version**:

Delete Objects After (Days): After this number of days since the last update, OBS will expire and then delete the objects meeting the specified conditions. The days set here must be larger than any of the days configured for the transition actions.

Suppose that you last updated the following files in OBS on November 7, 2023:

- **log/notConfigured-1.log** (This file has no WORM retention policy configured.)
- **log/expired-1.log** (The WORM retention policy configured for this file has expired.)
- **doc/withinRetention-1.doc** (The WORM retention policy configured for this file expires on November 30, 2023.)

Then on November 10, 2023, you last updated the following files:

- **log/notConfigured-2.log** (This file has no WORM retention policy configured.)
- **log/expired-2.log** (The WORM retention policy configured for this file has expired.)
- **doc/withinRetention-2.doc** (The WORM retention policy configured for this file expires on November 30, 2023.)

On November 10, 2023, you set the objects prefixed with **log** to expire one day later. You might encounter the following situations:

- Objects **log/notConfigured-1.log** and **log/expired-1.log** last updated on November 7, 2023 might be deleted after the last system scan. The deletion could happen on November 10, 2023 or November 11, 2023, depending on the time of the last system scan. **doc/withinRetention-1.doc** will not be deleted.
- Objects **log/notConfigured-2.log** and **log/expired-2.log** last uploaded on November 10, 2023 might be deleted on November 11, 2023 or November 12, 2023, depending on whether they have been stored for over one day (since their last update) when the system scan happened. **doc/withinRetention-2.doc** will not be deleted.

NOTE

For more information, see **Creating a Lifecycle Rule**.

Step 6 Click **OK**.

----End

Related Operations

When uploading an object, configure a retention policy for the object. For details, see **Streaming Upload (PUT)**.

To normally delete objects from a WORM-enabled bucket, see **Deleting an Object**.

6.3 CORS

This section describes how to use CORS in HTML5 to enable cross-origin access.

You can create CORS rules or replicate existing CORS rules from another bucket.

Context

Cross-origin access refers to access between different domains.

For browser security purposes, cross-origin access is typically restricted by the same-origin policy. Under this policy, resources in a domain cannot be accessed by a different domain using JavaScript.

If two web pages have the same protocol, domain name or IP, and port (if specified), they have the same origin. **Table 6-7** shows you how to determine whether the URLs listed in the table have the same origin as **https://support.huaweicloud.com/dir/test.html**.

Table 6-7 Cross-origin analysis

URL	Access Result	Reason
https://support.huaweicloud.com/dir/other.html	Succeeded	Same protocol, domain name, and port
https://support.huaweicloud.com/dir/inner/other.html	Succeeded	Same protocol, domain name, and port
http://support.huaweicloud.com/dir/test.html	Failed	Same domain name and port, but different protocols
https://support.huaweicloud.com:81/dir/test.html	Failed	Same protocol and domain name, but different ports
https://help.huaweicloud.com/dir/test.html	Failed	Same protocol and port, but different domain names

Scenarios

CORS (cross-origin resource sharing) is a browser-standard mechanism provided by the World Wide Web Consortium (W3C). It defines how a web client in one origin interacts with resources in another origin. For general web page requests, website scripts and contents in one origin cannot interact with those in another origin because of Same Origin Policies (SOPs).

OBS supports CORS. OBS resources can be accessed across origins.

OBS supports **static website hosting**. Static websites stored in OBS can respond to website requests from another origin only when CORS is configured for the bucket where the website files are stored.

CORS:

- Enables you to access OBS resources without using a proxy when using JavaScript and HTML5 to develop web applications.
- Enables you to directly upload files to OBS with the dragging function of HTML5, view the progress, or update contents directly from web applications.
- Enables external web pages, style sheets, or HTML5 applications hosted in different origins to share web fonts or images stored in OBS.

The CORS configuration takes about two minutes to take effect.

NOTICE

OBS is configured to support cross-origin access using the root domain name by default. To prevent potential attacks:

You can create a **crossdomain.xml** file with specific rules in the bucket for your client and add then **Security.loadPolicyFile("https://bucket.obs.ap-southeast-1.myhuaweicloud.com/crossdomain.xml")** in the flash code of the file. **bucket.obs.ap-southeast-1.myhuaweicloud.com** needs to be replaced with the actual access domain name of your bucket.

Prerequisites

Static website hosting has been configured. For details, see **Static Website Hosting**.

Creating a CORS Rule

You can use OBS Console, APIs, or SDKs to create CORS rules.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

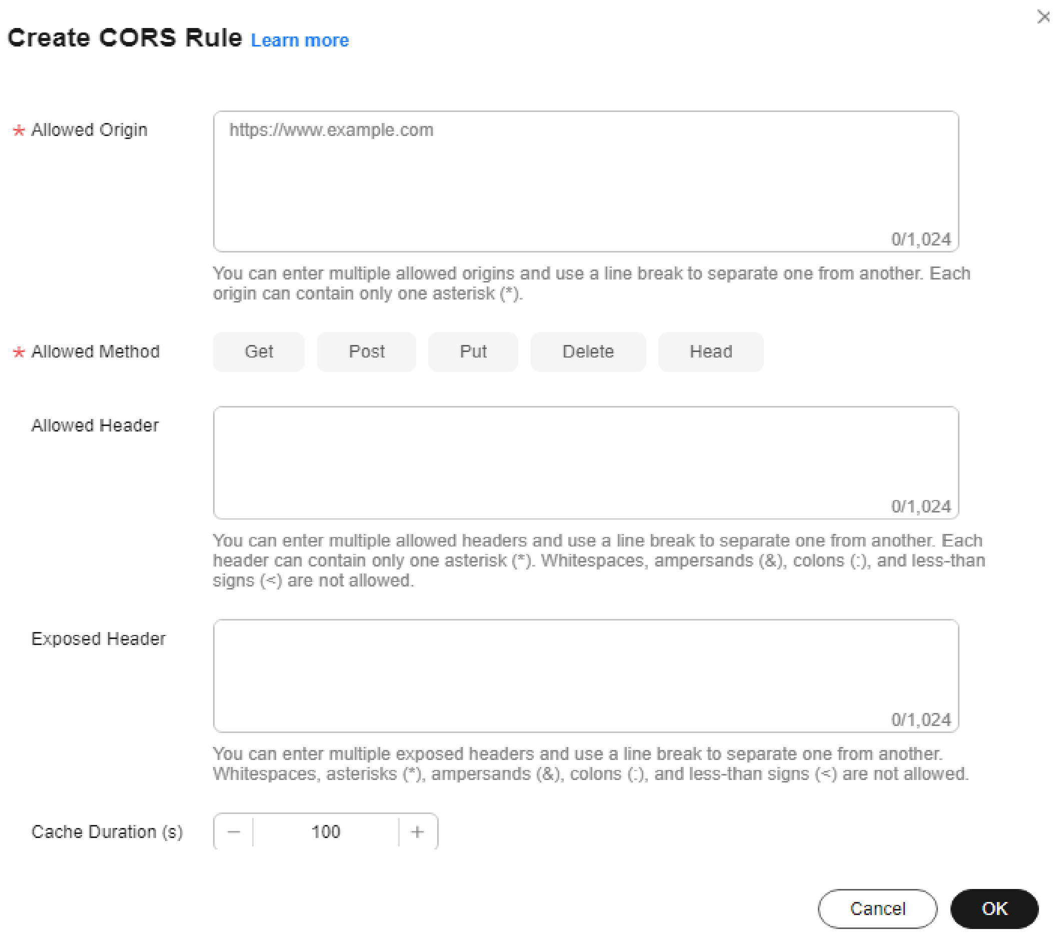
Step 3 In the navigation pane, choose **Permissions** > **CORS Rules**.

Step 4 Click **Create**. The **Create CORS Rule** dialog box is displayed. See **Figure 6-13** for details.

NOTE

A bucket can have a maximum of 100 CORS rules configured.

Figure 6-13 Creating a CORS rule



Create CORS Rule Learn more

* Allowed Origin: https://www.example.com (0/1,024)

You can enter multiple allowed origins and use a line break to separate one from another. Each origin can contain only one asterisk (*).

* Allowed Method: Get | Post | Put | Delete | Head

Allowed Header: (0/1,024)

You can enter multiple allowed headers and use a line break to separate one from another. Each header can contain only one asterisk (*). Whitespaces, ampersands (&), colons (:), and less-than signs (<) are not allowed.

Exposed Header: (0/1,024)

You can enter multiple exposed headers and use a line break to separate one from another. Whitespaces, asterisks (*), ampersands (&), colons (:), and less-than signs (<) are not allowed.

Cache Duration (s): 100

Step 5 In the **CORS Rule** dialog box, configure **Allowed Origin**, **Allowed Method**, **Allowed Header**, **Exposed Header**, and **Cache Duration (s)**.

NOTE

If CDN acceleration is enabled for the bucket, HTTP headers must be configured on CDN. For details, see **HTTP Header Settings**.

Table 6-8 Parameters in CORS rules

Parameter	Description

Allowed Origin	Mandatory
	Specifies the origins from which requests can access the bucket.
	Multiple matching rules are allowed. One rule occupies one line, and allows one wildcard character (*) at most. An example is given as follows:
	`http://rds.example.com`
	`https://*.vbs.example.com`

Parameter	Description
Allowed Method	Mandatory Specifies the allowed request methods for buckets and objects. The methods include Get, Post, Put, Delete, and Head.
Allowed Header	Optional Specifies the allowed headers in cross-origin requests. Only CORS requests matching the allowed headers are valid. You can enter multiple allowed headers (one per line) and each line can contain one wildcard character (*) at most. Spaces and special characters including **&:<** are not allowed.
Exposed Header	Optional Specifies the exposed headers in CORS responses, providing additional information for clients. By default, a browser can access only headers **Content-Length** and **Content-Type**. If the browser wants to access other headers, you need to configure those headers in this parameter. For the configuration of other headers, see **Configuring CORS for a Bucket**. You can enter multiple exposed headers (one per line). Spaces and special characters including ***&:<** are not allowed.
Cache Duration (s)	Mandatory Specifies the duration that your browser can cache CORS responses, expressed in seconds. The default value is **100**.

Step 6 Click **OK**.

Message "The CORS rule created successfully." is displayed. The CORS configuration takes effect within two minutes.

Then, only the addresses specified in **Allowed Origin** can access the OBS bucket using the methods specified in **Allowed Method**. Suppose you are configuring a CORS rule for bucket **testbucket** by setting **Allowed Origin** to **https://www.example.com**, **Allowed Method** to **GET**, **Allowed Header** to *****, **Exposed Header** to **ETag**, and **Cache Duration (s)** to **100**. Then, only GET requests from **https://www.example.com** are allowed to access bucket **testbucket**. In addition, there are no limits on headers in a request, the ETag value can be returned in the response, and the client which the requests are from can cache the CORS response for 100 seconds.

----End

Using the API

Configuring Bucket CORS

Using SDKs

Java	Python	C	Go	BrowserJS: not supported	.NET	Android	iOS	PHP	Node.js

Replicating CORS Rules

You can only use OBS Console to replicate CORS rules.

Step 1 In the navigation pane of OBS Console, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Overview**.

Step 4 In the **Basic Configurations** area, click **CORS Rules**. The **CORS Rules** page is displayed.

Alternatively, you can choose **Permissions** > **CORS Rules** in the navigation pane.

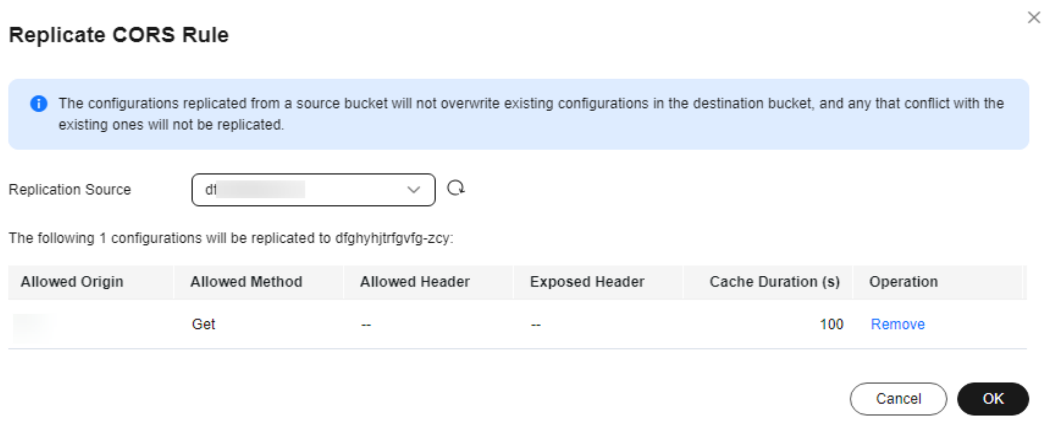
Step 5 Click **Replicate**.

Step 6 Select a replication source, which is the bucket whose CORS rules you want to replicate.

NOTE

- The CORS rules replicated from a source bucket will not overwrite existing rules in the destination bucket, and any that conflict with the existing ones will not be replicated.
- Both source and destination buckets must be of the 3.0 version.
- You can remove the rules that you do not want to replicate.
- There can be 100 CORS rules at most in a bucket. If the number of rules you will replicate plus the number of existing rules in the destination bucket exceeds 100, the replication will fail. Before replicating the rules, delete some if necessary.

Figure 6-14 Replicating CORS rules



Replicate CORS Rule

The configurations replicated from a source bucket will not overwrite existing configurations in the destination bucket, and any that conflict with the existing ones will not be replicated.

Replication Source: df

The following 1 configurations will be replicated to dfghyhjtrfgvfg-zcy:

Allowed Origin	Allowed Method	Allowed Header	Exposed Header	Cache Duration (s)	Operation
	Get	--	--	100	Remove

Step 7 Click **OK** to replicate the CORS rules to the destination bucket.

----End

6.4 URL Validation

OBS blocks access requests from blacklisted URLs and allows those from whitelisted URLs.

Scenarios

Some rogue websites may steal links from other websites to enrich their content without any costs. Link stealing hurts the interests of the original websites and it is also a strain on their servers. URL validation is designed to address this issue.

In HTTP, the **Referer** field allows websites and web servers to identify where people are visiting them from. URL validation of OBS utilizes this **Referer** field. The idea is that once you find that a request to your resource is not originated from an authorized source, you can have the request blocked or redirected to a specific web page. This way, OBS prevents unauthorized access to data stored in buckets.

Referers can be configured using a whitelist or blacklist.

Referer rules are as follows:

- The length of a whitelist or blacklist cannot exceed 1,024 characters.
- Referer format:
  - You can enter multiple referers, each in a line.
  - The referer parameter supports asterisks (*) and question marks (?). An asterisk works as a wildcard that can replace zero or multiple characters, and a question mark (?) can replace a single character.
  - If the referer header field contains **http** or **https** during download, the referer must contain **http** or **https**.
- If **Whitelisted Referers** is left blank but **Blacklisted Referers** is not, all websites except those specified in the blacklist are allowed to access data in the target bucket.

- If both **Whitelisted Referers** and **Blacklisted Referers** have referers, only those specified in the whitelist that are not in the blacklist are allowed to access the target bucket.

NOTE

If **Whitelisted Referers** overlaps with **Blacklisted Referers**, the overlapped referers are forbidden. For example, if both **Whitelisted Referers** and **Blacklisted Referers** contain **https://www.example.com**, requests from this website will be blocked.

- If **Whitelisted Referers** and **Blacklisted Referers** are both left blank, all websites are allowed to access data in the target bucket by default.
- Before determining whether a user has the four types of permissions (read, write, ACL read, and ACL write) for a bucket or objects in the bucket, check whether this user complies with the URL validation principles of the **Referer** field.
- Referer can be a wildcard domain name.

Whitelist and blacklist setting methods:

- Whitelist settings

  By setting a whitelist, you can allow requests from the websites in the whitelist, but deny those from the websites that are not in the whitelist.

  For the requests that are initialized from a browser's address box, you can add the **${null}** field to **Referer** of **Condition** to specify whether to allow the HTTP requests with a blank **referer**.

  To configure a whitelist, refer to the following policy:

```
"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}"]}
     }
   }
]
```

  If you set a whitelist in this way, only the requests whose **referer** is set to **www.example01.com** or left blank can work on resources in buckets.

- Blacklist settings

  To configure a blacklist, refer to the following policy:

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

```
  }
]
```

If you set a blacklist in this way, requests whose **referer** is set to **www.example01.com** or **www.example02.com** cannot work on resources in buckets.

Prerequisites

Static website hosting has been enabled.

Ways to Configure URL Validation

You can use OBS Console or APIs to configure URL validation.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Permissions** > **URL Validation**.

Step 4 Click ✎ next to the text box of **Whitelisted Referers** or **Blacklisted Referers** and enter the referers.

Principles for setting **Referers**:

- The length of a whitelist or blacklist cannot exceed 1024 characters.
- Referer format:
  - You can enter multiple referers, each in a line.
  - The referer parameter supports asterisks (*) and question marks (?). An asterisk works as a wildcard that can replace zero or multiple characters, and a question mark (?) can replace a single character.
  - If the referer header field contains **http** or **https** during download, the referer must contain **http** or **https**.
- If **Whitelisted Referers** is left blank but **Blacklisted Referers** is not, all websites except those specified in the blacklist are allowed to access data in the target bucket.
- If **Whitelisted Referers** is not left blank, only the websites specified in the whitelist are allowed to access the target bucket no matter whether **Blacklisted Referers** is left blank or not.

NOTE

If **Whitelisted Referers** is configured the same as **Blacklisted Referers**, the blacklist takes effect. For example, if both **Whitelisted Referers** and **Blacklisted Referers** are set to **https://www.example.com**, access requests from this address will be blocked.

- If **Whitelisted Referers** and **Blacklisted Referers** are both left blank, all websites are allowed to access data in the target bucket by default.
- Before determining whether a user has the four types of permissions (read, write, ACL read, and ACL write) for a bucket or objects in the bucket, check

whether this user complies with the URL validation principles of the **Referer** field.

Step 5 Click ✓ to save the settings.

----End

Using the API

Configure a URL validation whitelist

7 Disaster Recovery and Backup

7.1 Versioning

Scenarios

OBS can store multiple versions of an object. You can quickly search for and restore different versions or restore data in the event of accidental deletions or application faults.

By default, versioning is disabled for a new OBS bucket. New objects will overwrite existing objects in the bucket in case they have the same names.

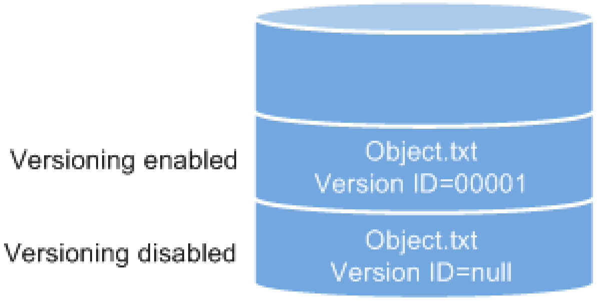
Constraints

When you enable **WORM** for a bucket, OBS automatically enables versioning and versioning cannot be suspended later for that bucket.

Enabling Versioning

- Enabling versioning does not change the version ID (**null**) and content of any existing object in a bucket. After versioning is enabled for a bucket, if you upload a new object with the same name as an existing object, a version ID will be assigned to this new object, as shown in **Figure 7-1**.

Figure 7-1 Versioning (with existing objects)



- In a versioning-enabled bucket, OBS automatically assigns a unique version ID to each newly uploaded object. Objects with the same name are stored in OBS with different version IDs, as shown in **Figure 7-2**.

Figure 7-2 Versioning (for new objects)

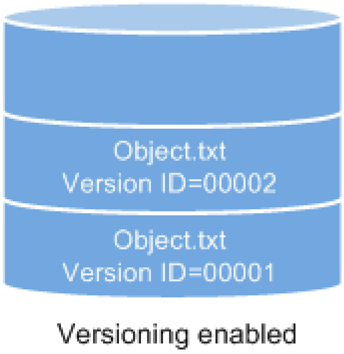
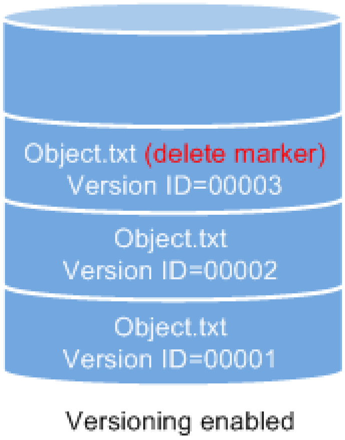


Table 7-1 Version description

Version	Description
Latest version	After versioning is enabled, each operation on an object will result in saving of the object with a new version ID. The version ID generated upon the latest operation is called the latest version.
Historical version	After versioning is enabled, each operation on an object will result in saving of the object with a new version ID. Version IDs generated upon operations other than the latest operation are called historical versions.

- The latest objects in a bucket are returned by default after a GET Object request.
- Objects can be downloaded by version IDs. By default, the latest object is downloaded if the version ID is not specified. For details, see **Related Operations** in **Versioning**.
- You can select an object and click **Delete** on the right to delete the object. After the object is deleted, OBS inserts a delete marker with a unique version ID for that object, and the deleted object is displayed in the **Deleted Objects** list. For details, see **Deleting an Object**. If attempts are then made to access this deleted object, error 404 will be returned.

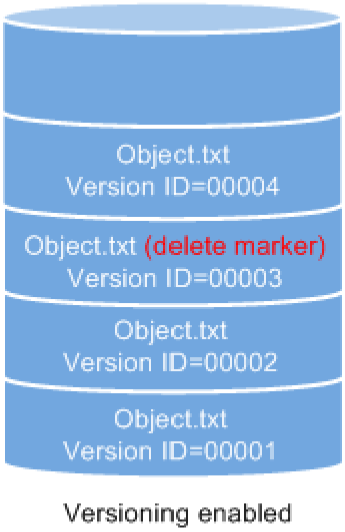
Figure 7-3 Object with a delete marker



- You can recover a deleted object by deleting the delete marker. For details, see **Related Operations**.
- After an object is deleted, you can specify the version number in **Deleted Objects** to permanently delete the object of the specified version. For details, see **Deleting Objects from a Bucket with Versioning Disabled, Suspended, or Enabled**.
- An object appears in either the object list or the list of deleted objects. It will never appear in both lists at the same time.

  For example, after object **A** is deleted, it will appear in the **Deleted Objects** list. If you later upload another object with the same name **A**, the new object **A** will appear in the **Objects** list, but the previously deleted object **A** will disappear from the **Deleted Objects** list. For details, see **Figure 7-4**.

Figure 7-4 Uploading a namesake object after the original one is deleted

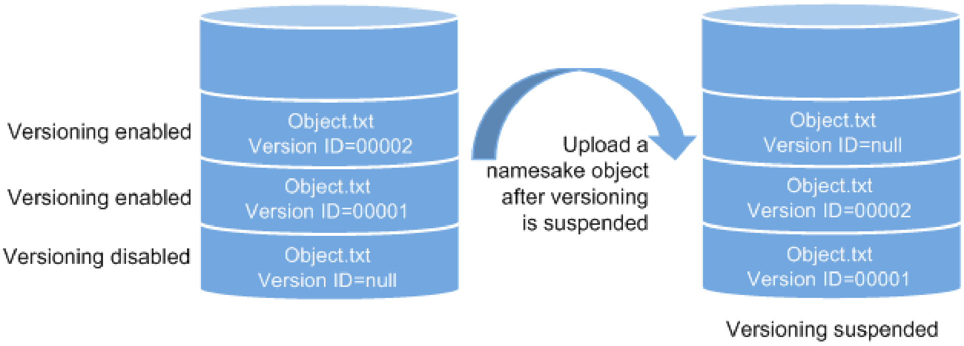


- All object versions except those with **Delete Marker** stored in OBS are charged.

Suspending Versioning

Once versioning is enabled for a bucket, it cannot be disabled, but it can be suspended. When versioning is suspended, a null, not a specific version ID, will be allocated to a newly uploaded object. If the newly uploaded object has the same name as an existing object with a null version ID, the new object will overwrite the existing object.

Figure 7-5 Object versions in the scenario when versioning is suspended



If versioning is no longer needed, you can suspend it. After versioning is suspended:

- Existing object versions are still retained in OBS. If you no longer desire these versions, manually delete them.
- Objects can be downloaded by version IDs. By default, the latest object is downloaded if the version ID is not specified.
- All historical object versions except those with **Delete Marker** stored in OBS are charged.

Differences Between Scenarios When Versioning Is Suspended and Disabled

If you delete an object from a versioning-suspended bucket, a delete marker with a **null** version ID will be generated, no matter whether the object has historical versions. But, if versioning is disabled, the same operation will not generate a delete marker.

NOTE

In a versioning-enabled bucket, each historical version of an object is stored and you will be billed for the storage of all object versions. Therefore, enable versioning when necessary.

Ways to Configure Versioning

You can use OBS Console, APIs, or SDKs to configure versioning.

Using OBS Console

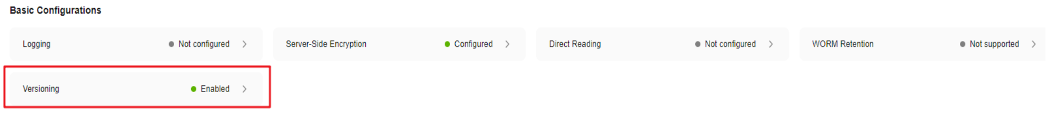
Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Overview**.

Step 4 In the **Basic Configurations** area, click **Versioning**.

Figure 7-6 Editing versioning status

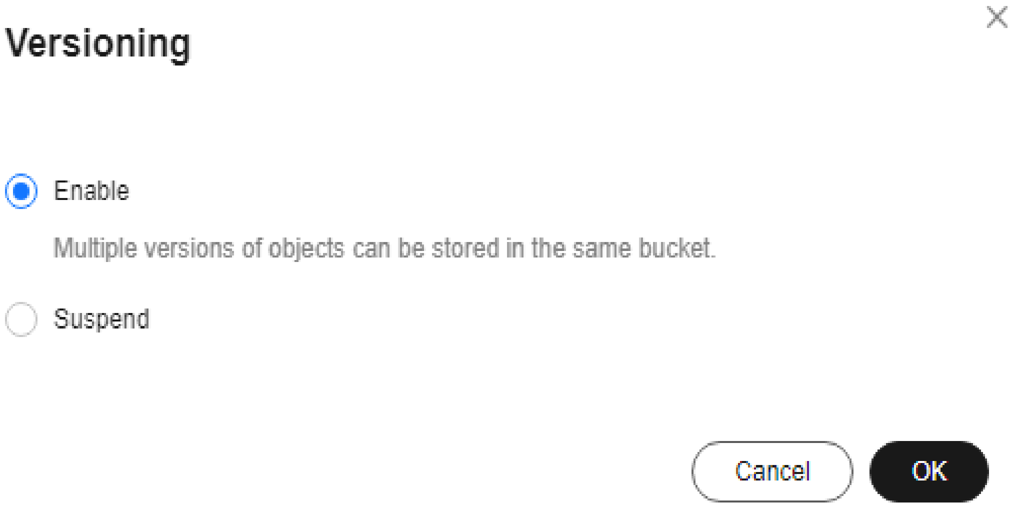


Basic Configurations

Logging	● Not configured
Server-Side Encryption	● Configured
Direct Reading	● Not configured
WORM Retention	● Not supported
Versioning	● Enabled

Step 5 Select **Enable**.

Figure 7-7 Configuring versioning



Versioning

● Enable

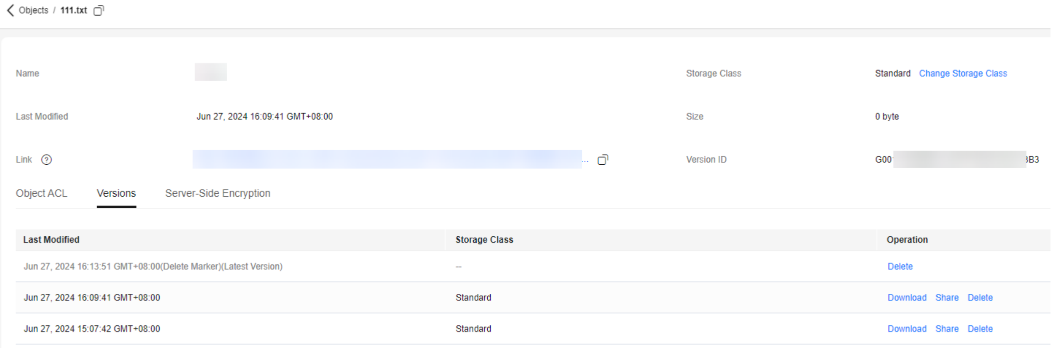
Multiple versions of objects can be stored in the same bucket.

○ Suspend

Step 6 Click **OK** to enable versioning for the bucket.

Step 7 Click an object to go to the object details page. On the **Versions** tab page, view all versions of the object.

Figure 7-8 Viewing object versions



Objects / 111.txt

Name	[redacted]	Storage Class	Standard Change Storage Class
Last Modified	Jun 27, 2024 16:09:41 GMT+08:00	Size	0 byte
Link	[redacted]	Version ID	G00[redacted]B3

Last Modified	Storage Class	Operation
Jun 27, 2024 16:13:51 GMT+08:00(Delete Marker)(Latest Version)	--	Delete
Jun 27, 2024 16:09:41 GMT+08:00	Standard	Download Share Delete
Jun 27, 2024 15:07:42 GMT+08:00	Standard	Download Share Delete

----End

Using the API

Configuring Versioning for a Bucket

Using SDKs

Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

Related Operations

After versioning is configured for a bucket, you can go to the object details page, click the **Versions** tab, and then delete, share, and download object versions, and extend the retention period of an object version.

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the object list, click the object you want to go to the object details page.

Step 4 On the **Versions** tab page, view all versions of the object.

Step 5 Perform the following operations on object versions:

1. Download a desired version of the object by clicking **Download** in the **Operation** column.

   NOTE

   If the version you want to download is in the Archive or Deep Archive storage class, restore it first.

2. Share a version of the object by clicking **Share** in the **Operation** column.
3. Delete a version of the object by choosing **More** > **Delete** in the **Operation** column. If you delete the latest version, the most recent version will become the latest version.

   NOTE

   In a WORM-enabled bucket, if an object has no retention policy configured or its retention policy has expired, you can delete a desired object version on the object's **Versions** tab page. If an object version is within the retention period, it cannot be deleted.

4. Locate the object version for which you want to extend the retention period, choose **More** > **Extend Retention Period**, and select a date. A retention period can only be extended, but not shortened.

----End

7.2 Cross-Region Replication

To replicate objects from a source bucket to a destination bucket in a different region, you can configure a single cross-region replication rule that is applied to all objects in the bucket, or you can configure multiple rules that are applied to a set of objects based on the prefix.

NOTE

A cross-region replication rule may not take effect immediately upon its configuration. Accordingly, the objects that this rule is applied to may not be replicated immediately after the rule is configured.

Buckets with WORM enabled do not support cross-region replication.

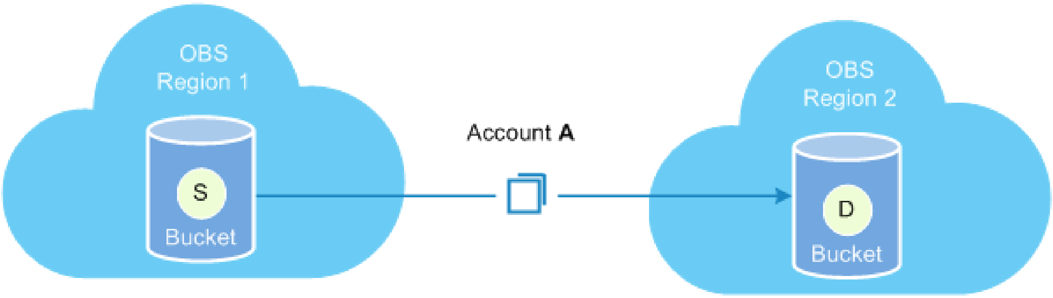
Scenarios

Cross-region replication provides disaster recovery across regions, catering to your needs for remote backup.

Cross-region replication refers to the process of automatically and asynchronously replicating data from a source bucket in one region to a destination bucket in another region based on the created replication rule. Source and destination buckets must belong to the same account. Replication across accounts is currently not supported.

You can configure a rule to replicate only objects with a specified prefix or replicate all objects in a bucket. Objects replicated to the destination bucket are precise copies of objects in the source bucket. These objects have the same name and metadata, including content, size, last modification time, creator, version ID, user-defined metadata, and ACL. By default, the storage class of an object replica is the same as that of the source object. You can also specify a storage class for the object copy.

Figure 7-9 Cross-region replication



- Regulatory compliance

  OBS stores data across AZs that are far apart from each other, but regulatory compliance may require farther distances. Cross-region replication enables you to meet regulatory requirements.
- Low latency

  The same OBS resources may need to be accessed from different locations. To minimize the access latency, you can use cross-region replication to create object copies in the nearest region.
- Data replication

  Cross-region replication allows you to easily migrate your data stored in OBS from one region to another.
- Data backup and disaster recovery

  To ensure data security and availability, you need to create explicit backups for all data written to OBS in the data center of another region, so that secure backups are available if the source data is damaged irrevocably.

- Ease of maintenance

  You have a compute cluster across regions that analyzes the same collection of objects. You need to maintain object replicas in these regions.

NOTICE

OBS helps you replicate your service data stored in OBS to a specified region, but Huawei Cloud has no access to your data. You need to ensure the legal compliance of your use of OBS on your own. If your replication involves cross-border transfer, ensure that your use complies with relevant laws and regulations.

Contents Replicated

After the cross-region replication rule is enabled, objects that meet the following conditions are copied to the destination bucket:

- Newly uploaded objects (excluding objects in the Deep Archive or Archive storage class)
- Updated objects, for example, objects whose content or ACL information is updated
- Historical objects in a bucket with **Synchronize Existing Objects** enabled (excluding objects in the Archive or Deep Archive storage class)

For example, on July 28, you enabled cross-region replication for a source bucket and uploaded objects A and B to the bucket. Objects A and B were synchronously replicated to the destination bucket. On July 29, you uploaded object C to the source bucket. This time, only object C was replicated to the destination bucket. On July 30, you modified object A in the source bucket. Then, only object A was replicated to the destination bucket.

NOTE

Cross-region replication does not replicate objects encrypted using SSE-C.

Constraints

Bucket version:

- Currently, only buckets of version 3.0 and later support cross-region replication. The version of a bucket can be viewed in the **Basic Information** area of the bucket's **Overview** page on OBS Console.

Status of the source bucket and destination bucket:

- The source and destination buckets must have the same versioning status.

Functions:

- If objects in the source bucket are in the Archive, or Deep Archive storage class, they cannot be replicated to the destination bucket.
- If the region where the destination bucket resides does not support a certain storage class, object replicas will be stored in the standard storage class.

- OBS currently only supports the replication between one source bucket and one destination bucket. Replication from one source bucket to multiple destination buckets is not supported. The destination bucket can be modified. However, modifying the destination bucket will change the destination bucket of all existing rules.
- If cross-region replication is enabled, data cannot be added to the end of objects in the source bucket.
- For a source bucket, you can create only one cross-region replication rule that applies to all objects in the bucket, or you can create a maximum of 100 cross-region replication rules based on object prefixes to replicate only some of the objects.

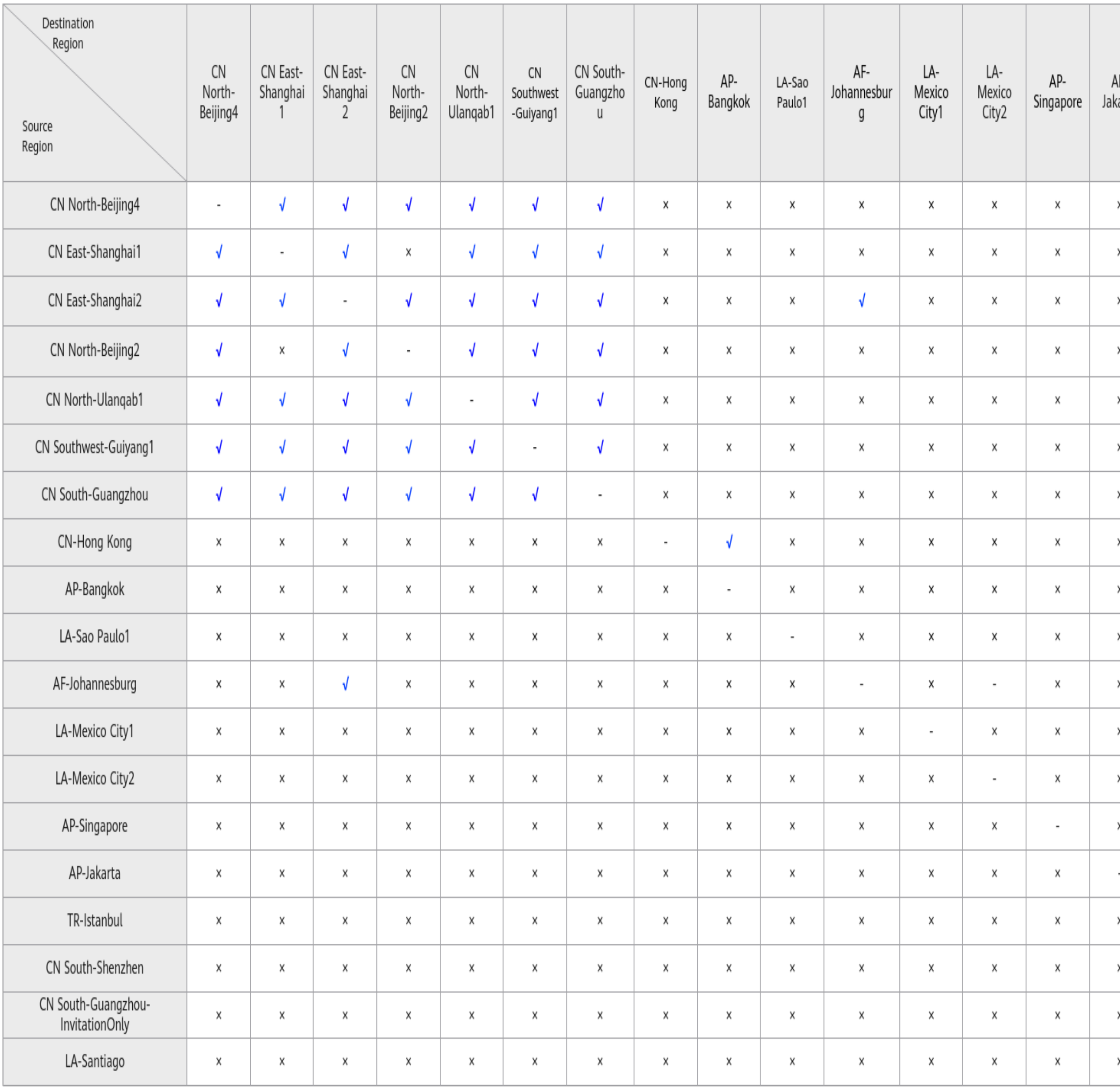
Time:

- A cross-region replication rule may not take effect immediately upon its configuration. Accordingly, the objects that this rule is applied to may not be replicated immediately after the rule is configured.

Area:

- The source and destination buckets must be in two different regions. Data cannot be replicated between buckets in the same region.
- Before replicating data, ensure that the source and destination regions can have their data replicated from each other. **Figure 7-10** lists the supported regions. "√" indicates that data can be replicated between regions. "x" indicates that data cannot be replicated between regions.

Figure 7-10 Replication between regions



Destination Region / Source Region	CN North-Beijing4	CN East-Shanghai1	CN East-Shanghai2	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	LA-Mexico City2	AP-Singapore
CN North-Beijing4	-	√	√	√	√	√	√	x	x	x	x	x	x	x
CN East-Shanghai1	√	-	√	x	√	√	√	x	x	x	x	x	x	x
CN East-Shanghai2	√	√	-	√	√	√	√	x	x	x	√	x	x	x
CN North-Beijing2	√	x	√	-	√	√	√	x	x	x	x	x	x	x
CN North-Ulanqab1	√	√	√	√	-	√	√	x	x	x	x	x	x	x
CN Southwest-Guiyang1	√	√	√	√	√	-	√	x	x	x	x	x	x	x
CN South-Guangzhou	√	√	√	√	√	√	-	x	x	x	x	x	x	x
CN-Hong Kong	x	x	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	-	x
LA-Mexico City1	x	x	x	x	x	x	x	x	x	x	x	-	x	x
LA-Mexico City2	x	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	-
AP-Jakarta	x	x	x	x	x	x	x	x	x	x	x	x	x	x
TR-Istanbul	x	x	x	x	x	x	x	x	x	x	x	x	x	x
CN South-Shenzhen	x	x	x	x	x	x	x	x	x	x	x	x	x	x
CN South-Guangzhou-InvitationOnly	x	x	x	x	x	x	x	x	x	x	x	x	x	x
LA-Santiago	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Synchronization of historical objects:

- By default, objects uploaded before cross-region replication is enabled are not replicated to the destination bucket unless the function for synchronizing existing objects is enabled.
- If the function for synchronizing existing objects is enabled, modifying the cross-region replication configuration may cause synchronization failures. Therefore, do not modify the cross-region replication configuration before the synchronization completes.

Versioning:

- If versioning is enabled or suspended for both the source and destination buckets and cross-region replication is also enabled for both buckets, deleting an object without specifying its version in the source bucket will also delete the object in the destination bucket.
- For an enabled cross-region replication rule, if you change the versioning status of the destination bucket, the replication of objects will fail. If you want to change the versioning status of the source bucket, delete the replication configuration first, and then make the change.

Access control:

- The owners of the source and destination buckets must have the read and write permissions to their respective bucket. Otherwise, data cannot be synchronized. If the system does not have the permissions to read the source bucket or write to the destination bucket due to read or write permission errors, objects cannot be replicated successfully, and such replication will not be resumed even if the permission error is rectified.
- Do not delete or overwrite object replicas in the destination bucket, or modify their ACLs, which may cause inconsistency of latest object versions or permission control settings between the destination bucket and the source bucket.
- After a replication with **Synchronize Existing Objects** enabled is complete, if the replication policy keeps unchanged, any ACL changes of source objects will be synchronized to object copies. However, ACL changes of source historical objects will not be synchronized to the copies of historical objects.

Others:

- Objects in a source bucket can be replicated to only one destination bucket, and cannot be replicated again from the destination bucket to another bucket. For example, bucket A and bucket B are in two different regions. You can replicate data from bucket A to bucket B or the other way round. However, data replicas in either bucket A or bucket B cannot be replicated anymore.
- If you delete the OBS agency for an enabled cross-region replication rule, the object replication will be in the **FAILED** status.

Prerequisites

The source bucket version is 3.0 or later, and cross-region replication is available in the region of the source bucket. For details about the support for cross-region replication in each region, search for "cross-region replication" on the **Function Overview** page.

Ways to Configure Cross-Region Replication

You can use OBS Console, APIs, or obsutil to configure cross-region replication.

Using OBS Console

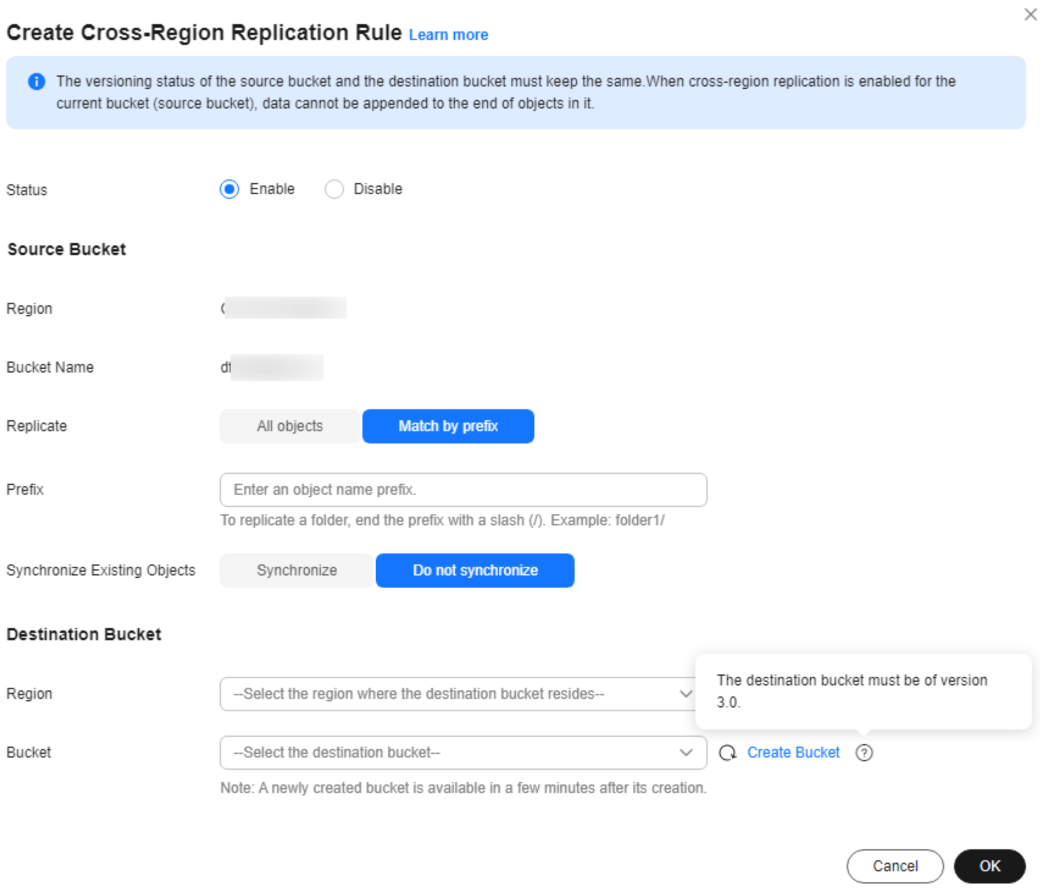
Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, click **Cross-Region Replication**.

Step 4 Click **Create Rule**. The **Create Cross-Region Replication Rule** dialog box is displayed. See **Figure 7-11**.

Figure 7-11 Creating a cross-region replication rule



Create Cross-Region Replication Rule Learn more

The versioning status of the source bucket and the destination bucket must keep the same.When cross-region replication is enabled for the current bucket (source bucket), data cannot be appended to the end of objects in it.

Status ● Enable ○ Disable

Source Bucket

Region (

Bucket Name dt

Replicate: All objects | **Match by prefix**

Prefix: Enter an object name prefix.

To replicate a folder, end the prefix with a slash (/). Example: folder1/

Synchronize Existing Objects: Synchronize | **Do not synchronize**

Destination Bucket

Region: --Select the region where the destination bucket resides--

The destination bucket must be of version 3.0.

Bucket: --Select the destination bucket--

Create Bucket

Note: A newly created bucket is available in a few minutes after its creation.

Cancel OK

NOTE

- The versioning status of the source and destination buckets must keep the same.
- A bucket can have only one destination bucket and one IAM agency configured for cross-region replication. The destination bucket and IAM agency specified in a later replication rule will overwrite those in the previous replication rule of the bucket.

Step 5 Configure a cross-region replication rule according to your service needs. For details about the parameters, see **Table 7-2**.

Table 7-2 Cross-region replication parameters

Parameter		Description
Status		Indicates whether the rule is enabled or disabled after being created. The versioning status of the source and destination buckets must keep the same.
Source Bucket	Replicate	Indicates the objects the rule will apply to. ● **All objects**: The rule applies to all objects in the bucket. ● **Match by prefix**: The rule applies only to objects with the specified prefix.
	Prefix	● To apply the rule to objects with the specified prefix, you must set **Prefix** to a value no longer than 1,024 characters. ● If the specified prefix overlaps with the prefix of an existing rule, OBS regards these two rules as one and forbids you to configure the one you are configuring. For example, if there is already a rule with prefix **abc** in OBS, you cannot configure another rule whose prefix starts with **abc**. ● To copy a folder, end the prefix with a slash (/), for example, **imgs/**.
	Synchronize Existing Objects	Indicates whether to synchronize the objects that were already in the bucket before the rule configuration to the destination bucket. By default, these objects are not synchronized.

Parameter		Description
	Replicate KMS encrypted objects	OBS will try to copy KMS encrypted objects no matter whether this option is selected or not. ● If this option is selected, only the IAM agencies that have the **KMS Administrator** permission for both source and destination ends are displayed in the drop-down list of **IAM Agency** in the **Create Cross-Region Replication Rule** dialog box. ● If this option is not selected, only the IAM agencies that do not have the **KMS Administrator** permission for either the source or destination end are displayed in the drop-down list of **IAM Agency** in the **Create Cross-Region Replication Rule** dialog box. If KMS is not available in the destination region or the agency does not have the **KMS Administrator** permission in the source and destination regions, KMS encrypted objects will fail to be replicated to the destination bucket, and the object replication status will be failed. After a KMS-encrypted object is replicated to the destination bucket, the key for encrypting the object copy changes to the default key **obs/default** of the destination region.
Destination Bucket	Region	Indicates the region of the destination bucket. The destination and source buckets must be in different regions.
	Bucket	Indicates the destination bucket.
	Change storage class for replicated objects	By default, this option is not selected, indicating that the storage class of object copies is the same as that of the source objects. If you need to change the storage class of objects copies, select this parameter, then you can specify a storage class.

Parameter		Description
Permissions	IAM Agency	Delegates OBS to operate your resources, so that OBS can use this agency to implement cross-region replication. If there is no IAM agency available, click **View IAM agencies** to create one. If you have already created IAM agencies, select one from the drop-down list. **NOTE** The IAM agency selected here must be of OBS. The OBS project must have the **Tenant Administrator** permission. If **Replicate KMS encrypted objects** is selected, you also need the **KMS Administrator** permission in the regions where the source and destination buckets are located.

Step 6 (Optional) Create an IAM Agency. For details, see **Creating an Agency for Cross-Region Replication**.

Step 7 Click **OK**. The cross-region replication rule is created.

----End

Using the API

Configuring Cross-Region Replication for a Bucket

Using obsutil

Command Line Structure

- In Windows
  - Copying a single object
    ```
    obsutil cp obs://srcbucket/key obs://dstbucket/[dest] [-dryRun][-u] [-crr] [-vlength] [-vmd5] [-p=1] [-threshold=52428800] [-versionId=xxx] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Copying objects in batches
    ```
    obsutil cp obs://srcbucket[/key] obs://dstbucket[/dest] -r [-dryRun][-f] [-flat] [-u] [-crr] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
- In Linux or macOS
  - Copying a single object
    ```
    ./obsutil cp obs://srcbucket/key obs://dstbucket/[dest] [-dryRun] [-u] [-crr] [-vlength] [-vmd5] [-p=1] [-threshold=52428800] [-versionId=xxx] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```
  - Copying objects in batches
    ```
    ./obsutil cp obs://srcbucket[/key] obs://dstbucket[/dest] -r [-dryRun] [-f] [-flat] [-u] [-crr] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] [-e=xxx] [-i=xxx] [-k=xxx] [-t=xxx]
    ```

NOTE

- The source path and destination path cannot be the same.
- The source path and destination path cannot be partly overlapped either. If the source path overlaps with the prefix of the destination path, recursive replication applies. If the destination path overlaps with the prefix of the source path, the replication may overwrite objects in the source path.

Examples

- Take the Windows OS as an example. Run the **obsutil cp obs://bucket-test/key obs://bucket-test2** command to copy a single object.

```
obsutil cp obs://bucket-test/key obs://bucket-test2

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx

[========================================================] 100.00% 6/s 0s
Copy successfully, 19B, obs://bucket-test/key --> obs://bucket-test2/key
ext.txt
```

- Take the Windows OS as an example. Run the **obsutil cp obs://bucket-test/temp/ obs://bucket-test2 -f -r** command to copy objects in batches.

```
obsutil cp obs://bucket-test/temp/ obs://bucket-test2 -r -f

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx
OutputDir: xxxx


[================================================================] 100.00% 10/s 0s
Succeed count is:   5         Failed count is:    0
Metrics [max cost:298 ms, min cost:192 ms, average cost:238.00 ms, average tps:9.71]
Task id is: 0476929d-9d23-4dc5-b2f8-0a0493f027c5
```

- For more examples, see **Copy**.

Parameter Description

Parameter	Optional or Mandatory	Description
srcbucket	Mandatory	Source bucket name
dstbucket	Mandatory	Destination bucket name
dest	Optional	Indicates the destination object name when copying an object, or the name prefix of destination objects when copying objects in batches.

Parameter	Optional or Mandatory	Description
key	Mandatory for copying an object. Optional for copying objects in batches.	Indicates the source object name when copying an object, or the name prefix of source objects when copying objects in batches. The rules are as follows: ● This parameter cannot be left blank when copying an object. If **dest** is left blank, the source object is copied to the root directory of the destination bucket. If the value of **dest** ends with a slash (/), the destination object name is the value of **dest** plus the source object name. Otherwise, the destination object name is the value of **dest**. ● If this parameter is left blank when copying objects in batches, all objects in the source bucket are copied. If not, objects whose name prefix is the set value in the source bucket are copied. The rules for confirming the name of the destination object are as follows: – If the value of **dest** ends with a slash (/), the destination object name is the value of **dest** plus the source object name. – If the value of **dest** does not end with a slash (/), the destination object name is *the value of **dest**/source object name*. **NOTE** ● If this parameter is configured but the **flat** parameter is not when copying objects in batches, the name of the source object contains the name prefix of the parent object. If **flat** is configured, then the name of the source object does not contain the name prefix of the parent object. ● For details about how to use this parameter, see **Copy**.
fr	Optional for copying an object (additional parameter)	Generates an operation result list when copying an object.
flat	Optional for copying objects in batches (additional parameter)	The name prefix of the parent object is excluded when copying objects in batches.

Parameter	Optional or Mandatory	Description
dryRun	Optional (additional parameter)	Conducts a dry run.
crr	Optional (additional parameter)	Enables the client-side cross-region replication function. In this mode, data is directly copied to the destination bucket from the source bucket through data stream. The buckets can by any two OBS buckets. **NOTE** ● If this parameter is configured, the configuration of client-side cross-region replication must be updated in the configuration file. For details, see **Updating a Configuration File**. ● The configurations of the source bucket and destination bucket are respectively **akCrr/skCrr/tokenCrr/endpointCrr** and **ak/sk/token/endpoint** in the configuration file. **NOTICE** When cross-region replication is enabled, the upload/download bandwidth, CPU, and memory resources of the host where commands are executed will be occupied, which may deteriorate the host performance.
vlength	Optional (additional parameter)	Verifies whether the object size in the destination bucket is the same as that in the source bucket after the copy task completes. **NOTE** This parameter must be used together with **crr**.
vmd5	Optional (additional parameter)	Verifies whether the MD5 value of the destination bucket is the same as that of the source bucket after the copy task completes. **NOTE** ● This parameter must be used together with **crr**. ● Objects in the source bucket must contain metadata **x-obs-meta-md5chksum**, or MD5 verification will be skipped. After the MD5 verification is successful, this parameter value is used for metadata **x-obs-meta-md5chksum** of the destination object, for later MD5 verification during download or copy.
u	Optional (additional parameter)	Indicates incremental copy. If this parameter is set, each object can be copied only when it does not exist in the destination bucket, its size is different from the namesake one in the destination bucket, or it has the latest modification time.

Parameter	Optional or Mandatory	Description
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart copy tasks when copying an object. The default value is the value of **defaultParallels** in the configuration file.
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart copy, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the object to be copied is smaller than the threshold, copy the object directly. If not, a multipart copy is required. ● If you copy an object directly, no part record is generated, and resumable transmission is not supported. ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes).
versionId	Optional for copying an object (additional parameter)	Source object version ID that can be specified when copying an object
acl	Optional (additional parameter)	Access control policies for destination objects that can be specified when copying objects. Possible values are: ● private ● public-read ● public-read-write ● bucket-owner-full-control **NOTE** The preceding four values indicate private read and write, public read, public read and write, and bucket owner full control.

Parameter	Optional or Mandatory	Description
sc	Optional (additional parameter)	Storage classes of the destination objects that can be specified when copying objects. Possible values are: ● **standard**: Standard storage class. It features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month) or data that is smaller than 1 MB. ● **warm**: Infrequent Access storage class. It is ideal for storing infrequently accessed (less than 12 times a year) data, but when needed, the access has to be fast. ● **cold**: Archive storage class. It provides secure, durable, and inexpensive storage for rarely-accessed (once a year) data. ● **deep-archive**: Deep Archive storage class (under limited beta testing). It is suitable for storing data that is barely (once every few years) accessed. This storage class costs less than the Archive storage class, but takes longer time (usually several hours) to restore data.
meta	Optional (additional parameter)	Standard or custom metadata that can be specified for destination objects in object replication. This parameter should be configured in the following format: *key1*:*value1*#*key2*:*value2*#*key3*:*value3*. **NOTE** 1. The format example above indicates that the destination objects contain three groups of custom metadata: *key1:value1*, *key2:value2*, and *key3:value3*. 2. Standard metadata headers include Content-Type, Content-Encoding, Cache-Control, Content-Disposition, Content-Language and Expires.
fs	Optional (additional parameter)	Specifies whether the method of listing parallel file systems is applied. If you are listing parallel file systems, you are recommended to add this parameter. **CAUTION** ● With this method, the listing time required varies largely depending on the directory structures. ● After this parameter is enabled, **marker** and **limit** will be ignored. Then, the buckets or parallel file systems (including directories) will be calculated. ● This parameter is only supported by obsutil 5.5.12 and later.

Parameter	Optional or Mandatory	Description
ps	Optional (additional parameter)	Indicates the size of each part in a multipart copy task, in bytes. The value ranges from **100KB** to **5GB**. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This parameter value can contain a unit, for example, **1MB** (indicating 1048576 bytes). ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source object size.
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart copy and saved to the **copy** subfolder. After the copy succeeds, its part record is deleted automatically. If the copy fails or is suspended, the system attempts to resume the task according to its part record when you perform the copy the next time.
r	Mandatory for copying objects in batches (additional parameter)	Copies objects in batches based on a specified name prefix of objects in the source bucket.
f	Optional for copying objects in batches (additional parameter)	Runs in force mode.
j	Optional for copying objects in batches (additional parameter)	Indicates the maximum number of concurrent tasks for copying objects in batches. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.

Parameter	Optional or Mandatory	Description
exclude	Optional for copying objects in batches (additional parameter)	Indicates the matching patterns of source objects that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use **\*** to represent ***** and **\\?** to represent **?**. ● If the name of the object to be copied matches the value of this parameter, the object is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/). ● Multiple **exclude** parameters can be specified, for example, **-exclude=*.xxx -exclude=*.xxx**.

Parameter	Optional or Mandatory	Description
include	Optional for copying objects in batches (additional parameter)	Indicates the matching patterns of source objects that are included, for example: ***.jpg**. **NOTE** • The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. • You can use **\*** to represent ***** and **\\?** to represent **?**. • Only after identifying that the name of the file to be copied does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is copied. If not, the file is skipped. **NOTICE** • You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. • The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. • This matching pattern applies only to objects whose names do not end with a slash (/). • Multiple **include** parameters can be specified, for example, **-include=*.xxx -include=*.xxx**.

Parameter	Optional or Mandatory	Description
timeRange	Optional for copying objects in batches (additional parameter)	Indicates the time range matching pattern when copying objects. Only objects whose latest modification time is within the configured time range are copied. This pattern has a lower priority than the object matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured object matching patterns. **NOTE** - The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. - Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. - If this parameter is set to ***-***time2*, all files whose latest modification time is earlier than *time2* are matched. If it is set to *time1***-***, all files whose latest modification time is later than *time1* are matched. **NOTICE** - Time in the matching pattern is the UTC time. - This matching pattern applies only to objects whose names do not end with a slash (/).
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on objects whose names end with a slash (/).
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** - The naming rule for result lists is as follows: **cp_{succeed | failed | warning}_report_***time***_TaskId.txt** - By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file. - If there are multiple folders and files and you need to confirm the detailed error information about a failed task, refer to the failure list **cp_failed_report_***time***_TaskId.txt** in the result list folder and the **log files** in the log path.

Parameter	Optional or Mandatory	Description
config	Optional (additional parameter)	User-defined configuration file for executing the current command. To learn the parameters that can be configured in this file, see **Configuration Parameters**.
e	Optional (additional parameter)	Specifies the endpoint.
i	Optional (additional parameter)	Specifies the user's AK.
k	Optional (additional parameter)	Specifies the user's SK.
t	Optional (additional parameter)	Specifies the user's security token.

Response

Field	Description
Parallel	Parameter **-p** in the request
Jobs	Parameter **-j** in the request
Threshold	Parameter **-threshold** in the request
PartSize	Parameter **-ps** in the request
Exclude	Parameter **-exclude** in the request
Include	Parameter **-include** in the request
TimeRange	Parameter **-timeRange** in the request
VerifyLength	Parameter **-vlength** in the request
VerifyMd5	Parameter **-vmd5** in the request
CheckpointDir	Parameter **-cpd** in the request
OutputDir	Parameter **-o** in the request
ArcDir	Parameter **-arcDir** in the request
Succeed count	Number of successful tasks
Failed count	Number of failed tasks

Field	Description
Skip count	Number of tasks that are skipped during incremental upload, download, or copy, and synchronous upload, download, or copy. **NOTE** Skipped tasks are recorded into successful tasks.
Warning count	Number of tasks that are executed successfully but contain warnings. **NOTE** ● The task for which a warning is generated may be a failure or a success, which needs to be further determined according to the corresponding result list. ● The number of tasks that generate warnings is independent of the number of successful or failed tasks. The total number of tasks is the number of successful tasks plus the number of failed tasks.
Succeed bytes	Number of bytes that are successfully uploaded or downloaded.
max cost	Maximum duration of all tasks, in ms
min cost	Minimum duration of all tasks, in ms
average cost	Average duration of all tasks, in ms
average tps	The average number of tasks completed per second
Task id	Unique ID of an operation, which is used to search for the result list generated in a batch task

Related Operations

Creating an Agency for Cross-Region Replication

Step 1 In the **Create Cross-Region Replication Rule** dialog box on OBS Console, click **View IAM agencies** to jump to the **Agencies** page on the IAM console.

Step 2 Click **Create Agency**.

Step 3 Enter an agency name.

Step 4 Select **Cloud service** for the **Agency Type**.

Step 5 Select **Object Storage Service (OBS)** for **Cloud Service**.

Step 6 Set a validity period.

Step 7 Click **Next**.

NOTE

The console for creating an agency has the new and old editions. The following steps use the new edition.

Step 8 On the **Select Policy/Role** page, search for and select **OBS Administrator** and click **Next**.

Step 9 On the **Select Scope** page, select **Global services** for **Scope** and click **OK**.

Step 10 (Optional) If **Replicate KMS encrypted objects** is selected, the IAM agency also needs the **KMS Administrator** permission in the regions where the source and destination buckets are located.

1. Go to the **Agencies** page of the IAM console and click the name of the agency created in the previous step.
2. Choose the **Permissions** tab and click **Authorize**.
3. On the **Select Policy/Role** page, search for and select **KMS Administrator**. Then, click **Next**.
4. On the **Select Scope** page, select **Region-specific projects** for **Scope**. Then, select the projects in the regions where the source and destination buckets are located. Click **OK**.

----End

8 Domain Name Management

8.1 Overview of Bucket Domain Names

OBS bucket-related domain names include **user-defined domain names** and **default domain names**.

When you use the default domain name to access objects in a bucket, the objects are forcibly downloaded and cannot be previewed. If you do not want to expose a bucket's default domain name or want to preview objects in a browser, you can bind a user-defined domain name to the bucket and use this bound domain name to access objects in the bucket..

User-defined Domain Names

You can bind a user-defined domain name to a bucket, and then use this domain name to access the objects hosted in the bucket. For more information, see **Accessing a Bucket Using a User-Defined Domain Name**.

Default Domain Names

OBS assigns each newly created bucket an access domain name (also the default domain name) based on the bucket name and region information.

examplebucket.obs.ap-southeast-1.myhuaweicloud.com is an example default domain name. In this example, **examplebucket** is the bucket name and **obs.ap-southeast-1.myhuaweicloud.com** is the endpoint of the region where the bucket is located. For more information, see **OBS Domain Names**.

For security and compliance reasons, Huawei Cloud OBS does not allow you to use the default domain name of a bucket to preview objects in the bucket. After you upload a file to a bucket, OBS creates an access address for the uploaded file based on the bucket's default domain name. When you use this address to access the file in a browser, the file content is not displayed. Instead, the file is downloaded as an attachment.

The effective time and supported regions are as follows:

January 1, 2022: CN North-Beijing1, CN North-Beijing4, CN East-Shanghai1, CN East-Shanghai2, and CN South-Guangzhou

March 25, 2022: CN-Hong Kong, AP-Bangkok, AP-Singapore, AF-Johannesburg, LA-Mexico City1, LA-Mexico City2, LA-Sao Paulo1, AP-Jakarta, and LA-Santiago

8.2 Accessing a Bucket Using a User-Defined Domain Name

After you upload a file to a bucket, OBS generates an access address for the file that includes the bucket's default domain name. Accessing this address using a browser will automatically download the file. To avoid exposing the default domain name of a bucket or to preview files online through a browser, you can bind a user-defined domain name to the bucket and use it to access the files in the bucket.

Scenarios

- File preview: To avoid automatic file downloads, you can bind a user-defined domain name to a bucket for online file preview.
- Preventing domain names from being blocked: Some application platforms may block the default domain name of a bucket. To access files in a bucket, you can bind a user-defined domain name to the bucket.
- Keeping the access address unchanged: To keep website or file access addresses unchanged after a migration or storage location change, you can upload the files to a bucket and bind a user-defined domain name to the bucket. This will establish a mapping between the website domain name and the default bucket domain name. In this way, you can still access the files using the original address.
- Improved brand image: Having a fixed and personalized domain name for accessing an enterprise website can make the enterprise appear more professional and reliable. This enhances enterprise image and improves customer experience.
- Easy access: You can use a user-defined domain name that is easy to remember to access a bucket and files in the bucket.

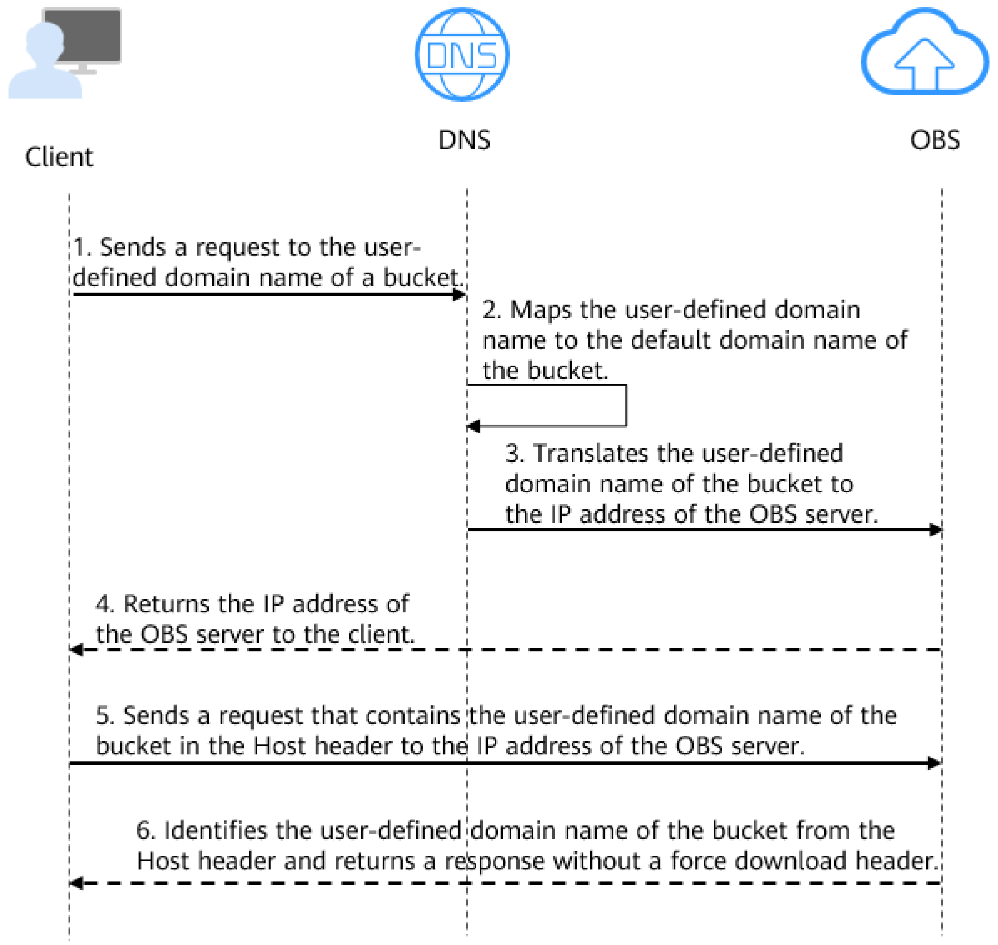
Principles

The following describes the process of accessing objects in a bucket using a user-defined domain name and the process of accessing objects in a bucket using the default bucket domain name.

Using a User-Defined Domain Name to Access Objects in a Bucket

Figure 8-1 shows the process of accessing objects in a bucket using a user-defined domain name.

Figure 8-1 Using a user-defined domain name to access objects in a bucket

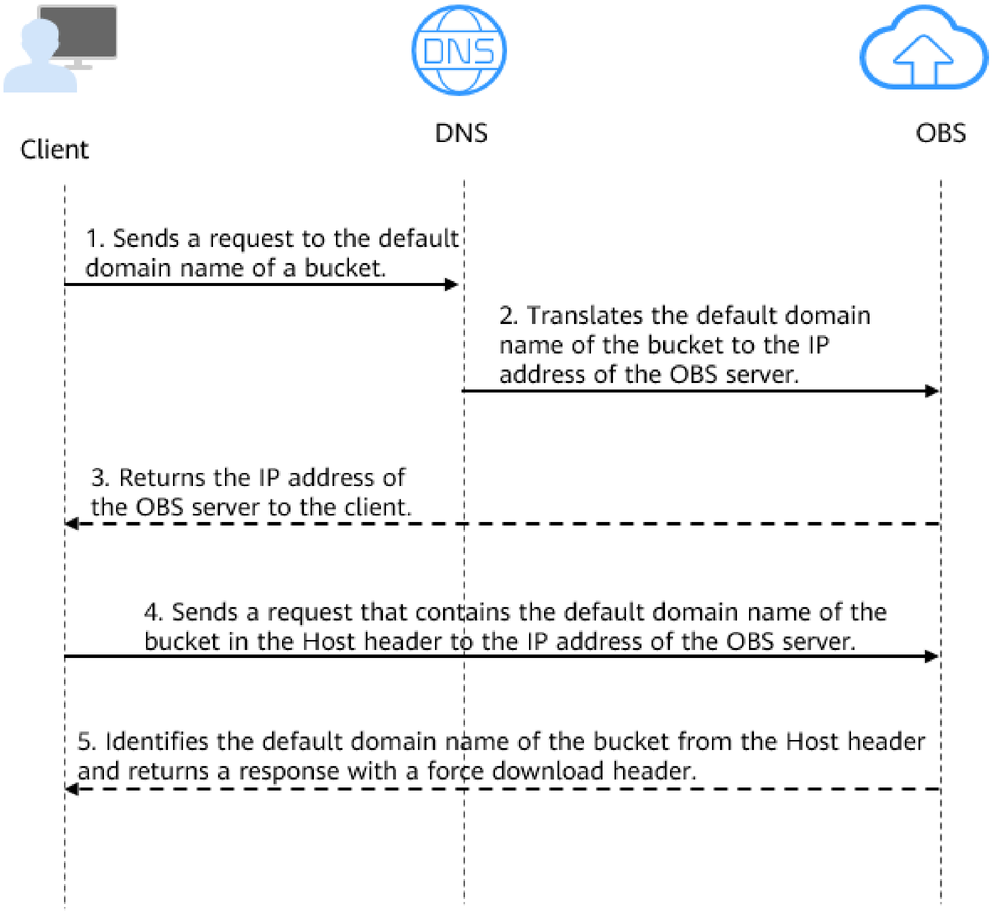

Client
DNS
OBS
1. Sends a request to the user-defined domain name of a bucket.
2. Maps the user-defined domain name to the default domain name of the bucket.
3. Translates the user-defined domain name of the bucket to the IP address of the OBS server.
4. Returns the IP address of the OBS server to the client.
5. Sends a request that contains the user-defined domain name of the bucket in the Host header to the IP address of the OBS server.
6. Identifies the user-defined domain name of the bucket from the Host header and returns a response without a force download header.

1. A user enters a file access address (that includes a user-defined domain name) in the browser to initiate an access request.
2. DNS maps the user-defined domain name to the default domain name of the bucket.
3. DNS translates the default domain name of the bucket to the IP address of the OBS server.
4. OBS returns the IP address of the server to the client.
5. The client sends a request with the user-defined domain name specified in the Host header to the IP address of the OBS server.
6. OBS identifies the user-defined domain name in the Host header and returns a response with **Content-Disposition** set to **inline** to the client, which indicates that the file is expected to be displayed in the browser.

Using the Default Domain Name of a Bucket to Access Objects in the Bucket

Figure 8-2 shows the process of accessing objects in a bucket using its default domain name (*bucketName*.obs.*endpoint*.myhuaweicloud.com).

Figure 8-2 Using the default domain name of a bucket to access objects in the bucket


Client
DNS
OBS
1. Sends a request to the default domain name of a bucket.
2. Translates the default domain name of the bucket to the IP address of the OBS server.
3. Returns the IP address of the OBS server to the client.
4. Sends a request that contains the default domain name of the bucket in the Host header to the IP address of the OBS server.
5. Identifies the default domain name of the bucket from the Host header and returns a response with a force download header.

1. A user enters a file access address (that includes the default domain name of the bucket) in the browser to initiate an access request.
2. DNS translates the default domain name of the bucket to the IP address of the OBS server.
3. OBS returns the IP address of the server to the client.
4. The client sends a request with the default domain name specified in the Host header to the IP address of the OBS server.
5. OBS identifies the default domain name in the Host header and returns a response with **Content-Disposition** set to **attachment** to the client, which indicates that the file is expected to be downloaded and saved locally.

Constraints

Table 8-1 Constraints

Item	Description
Bucket version	Only buckets whose version is 3.0 or later support user-defined domain names. The bucket version can be viewed in the **Basic Information** area of the bucket's **Overview** page on OBS Console.
Number of domain names	By default, a bucket can have up to 20 user-defined domain names bound.
Functions	● User-defined domain names currently only allow requests over HTTP. If you want to use a user-defined domain name to access OBS over HTTPS, you need to configure CDN and manage HTTPS certificates on the CDN console. For details, see **Configuring an HTTPS Certificate**. ● A user-defined domain name can be bound to only one bucket. ● Chinese domain names are not supported. ● The suffix of a user-defined domain name can contain 2 to 6 uppercase or lowercase letters. ● As required by the MIIT, if the bucket which a user-defined domain name is bound to is in a Chinese mainland region, you must complete the ICP filing. **NOTE** If an acceleration domain name is also required, to prevent objects in OBS buckets from being directly downloaded upon access, you need to perform other required operations after the user-defined domain name and the acceleration domain name have been configured. For details, see **With CDN Acceleration Enabled, Why Are the Objects in My OBS Bucket Directly Downloaded When I Access Them?**

Prerequisites

- There is a bucket with objects. For details, see **Creating a Bucket**.
- There is a second-level domain.
- As required by the MIIT, if the bucket which a user-defined domain name is bound to is in a Chinese mainland region, you must complete the ICP filing.

Configuring a User-Defined Domain Name

You can use OBS Console, APIs, or SDKs to configure user-defined domain names.

Using OBS Console

To configure a user-defined domain name on OBS Console, perform the following steps:

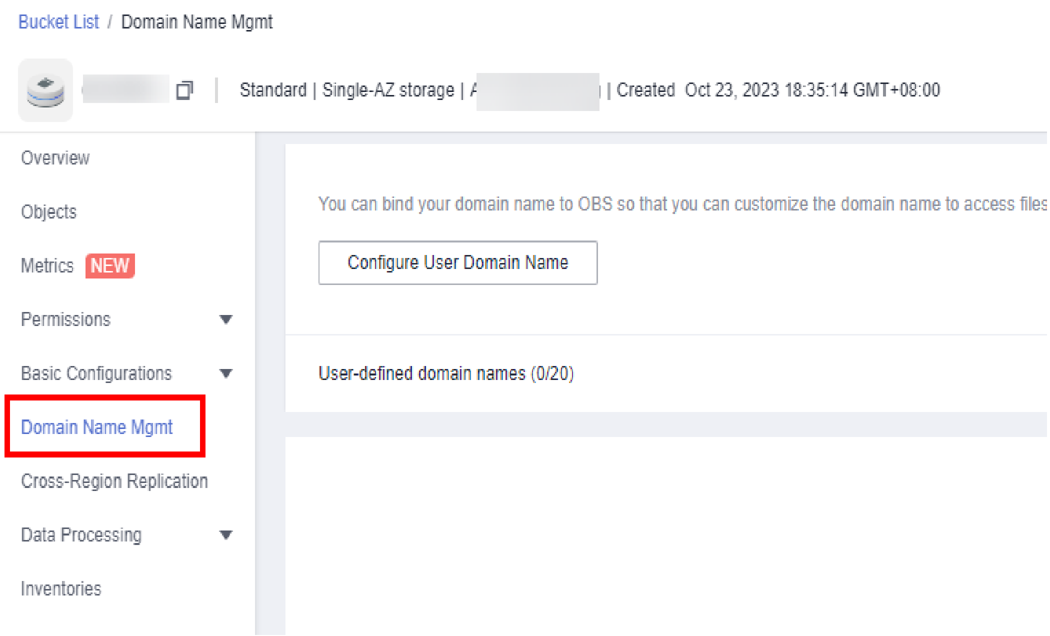
1: Configure a User-Defined Domain Name

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Domain Name Mgmt**.

Figure 8-3 Domain name management page



Bucket List / Domain Name Mgmt

[redacted] | Standard | Single-AZ storage | [redacted] | Created Oct 23, 2023 18:35:14 GMT+08:00

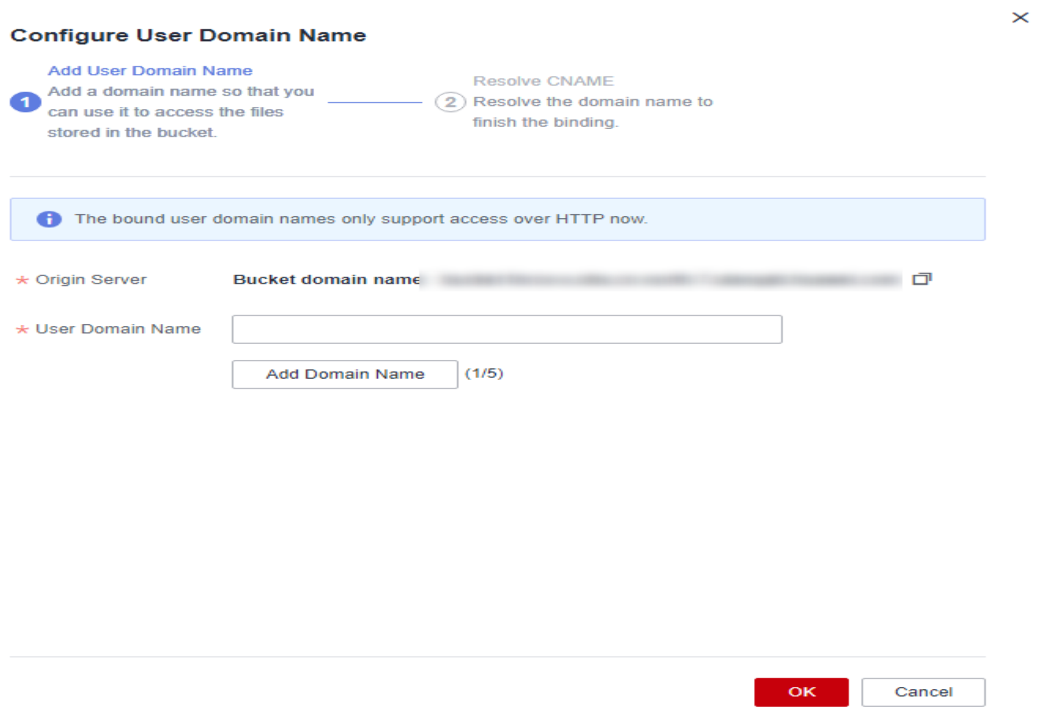
You can bind your domain name to OBS so that you can customize the domain name to access files

User-defined domain names (0/20)

Step 4 Click **Configure User Domain Name** in the upper part of the page or click **Configure User Domain Name** in the lower card area of the page. In the displayed dialog box, enter the domain name and click **OK**.

The domain name suffix can contain 2 to 6 uppercase or lowercase letters, for example, .com and .cn.

Figure 8-4 Configuring a user domain name



Configure User Domain Name

1 Add User Domain Name
Add a domain name so that you can use it to access the files stored in the bucket.

2 Resolve CNAME
Resolve the domain name to finish the binding.

The bound user domain names only support access over HTTP now.

* Origin Server: Bucket domain name

* User Domain Name:

----End

2: Configure CNAME Record

Configure a CNAME record to map the user-defined domain name to the default domain name of the bucket.

Step 1 If you add a domain name registered with Huawei Cloud, click **Resolve**. A CNAME record will be added by Huawei Cloud DNS.

If it is a domain name not registered with Huawei Cloud, skip this step and go to **Step 2**.

Step 2 Manually configure the CNAME record.

The CNAME record configuration varies depending on DNS service providers. If your DNS service provider is not Huawei Cloud, refer to **Configuring a CNAME Record**.

If you use Huawei Cloud DNS, perform the following steps:

1. Log in to the Huawei Cloud console. On the homepage, choose **Networking** > **Domain Name Service**. The DNS console is displayed.
2. In the navigation pane, choose **Public Zones**.
3. Locate the row that contains the domain name to which you want to add a record set and click the domain name.
4. On the **Record Sets** tab, click **Add Record Set**.
5. Configure the parameters. Retain the default values for those not listed in the table below. For details, see **Adding a CNAME Record Set**.

Table 8-2 Parameters

Parameter	Description	Example Value
Name	Prefix of the domain name to be resolved. For example, if the domain name is **example.com**, the prefix can be as follows: – **www**: The domain name is www.example.com, which is usually used for a website. – Left blank: The domain name is example.com. To use an at sign (@) as the domain name prefix, just leave this parameter blank. – **abc**: The domain name is abc.example.com, a subdomain of example.com. – **mail**: The domain name is mail.example.com, which is usually used for email servers. – *****: The domain name is *.example.com, which is a wildcard domain name, indicating all subdomains of example.com.	Leave it blank.
Type	Type of the record set, which is CNAME here. A message may be displayed indicating that the record set you are trying to add conflicts with an existing record set. For details, see **Why Is a Message Indicating Conflict with an Existing Record Set Displayed When I Add a Record Set?**	CNAME – Map one domain to another

Parameter	Description	Example Value
Line	Resolution line. The DNS server will return the IP address of the specific line, depending on where the visitors come from. This parameter is only configurable for public zone record sets. – **Default**: returns the default resolution result irrespective of where the visitors come from. – **ISP**: returns the resolution result based on visitors' carrier networks. For details, see **Configuring ISP Lines**. – **Region**: returns the resolution result based on visitors' geographical locations. For details, see **Configuring Region Lines**. – **Custom line**: returns a specific IP address based on the IP address range of visitors. For details, see **Configuring Custom Lines**.	Default
TTL (s)	Cache duration of the record set on a local DNS server, in seconds. The value ranges from **1** to **2147483647**, and the default value is **300**. If your service address changes frequently, set TTL to a smaller value. For details, see **What Is TTL?**	300
Value	Domain name alias. You can enter only one domain name. **NOTE** – If CDN acceleration is not used, set this parameter to the bucket domain name. – If CDN acceleration is used, set this parameter to the domain name specified in the CNAME record allocated by CDN.	webserver01.example.com

6. Click **OK**.
7. Check whether the CNAME record takes effect.

   Open the Windows command line interface and run the following command:

   ```
   nslookup -qt=cname User-defined domain name bound to the bucket
   ```

   - Without CDN acceleration: If the output is the bucket domain name, the CNAME record has taken effect.
   - With CDN acceleration: If the output is the CNAME domain name allocated by CDN, the CNAME record has taken effect.

----End

Using APIs

Configuring a Custom Domain Name for a Bucket

Using SDKs

Java: not supported	Python: not supported	**C**	**Go**	BrowserJS: not supported	.NET: not supported	Android: not supported	iOS: not supported	PHP: not supported	Node.js: not supported

Follow-Up Operations

Accessing a User-Defined Domain Name Over HTTPS

User-defined domain names currently only allow requests over HTTP. To use HTTPS, you can **submit a service ticket**.

You can also manage HTTPS certificates on the CDN console to allow access over HTTPS. For details, see **Configuring an HTTPS Certificate**.

Preventing Data Theft

Some rogue websites may steal links from other websites to enrich their content without any costs. Link stealing hurts the interests of the original websites and it is also a strain on their servers. URL validation is designed to address this issue.

To prevent data stored in buckets from being theft, you can specify Referer (a whitelisted, blacklisted, or empty Referer) in the HTTP header to allow access only from specific sources. For details, see **URL Validation**.

Configuring Static Website Hosting

If you want to host your static website on OBS and use a user-defined domain name to access the website, you need to configure static website hosting for the storage space. For details, see **Static Website Hosting**.

FAQ

- **How Do I Preview Objects in OBS in a Browser Online?**
- **Why Is the Message "NoSuchBucket" Displayed When I Use a User-Defined Domain Name to Access a Bucket That Can Be Accessed by the OBS Domain Name?**
- **Why Is the CNAME Resolution Status Unknown?**
- **Why Can Only the Domain Names I Bought on Huawei Cloud Be Automatically Resolved?**

9 Data Management

9.1 Lifecycle Management

9.1.1 Overview of Lifecycle Management

Scenarios

The last modification time of an object determines when the lifecycle rule applied to the object becomes effective to periodically transition or delete the object. The transition action is to automatically change objects that are no longer frequently accessed to a storage class with a lower cost without copying the objects themselves.

Figure 9-1 OBS object lifecycle management

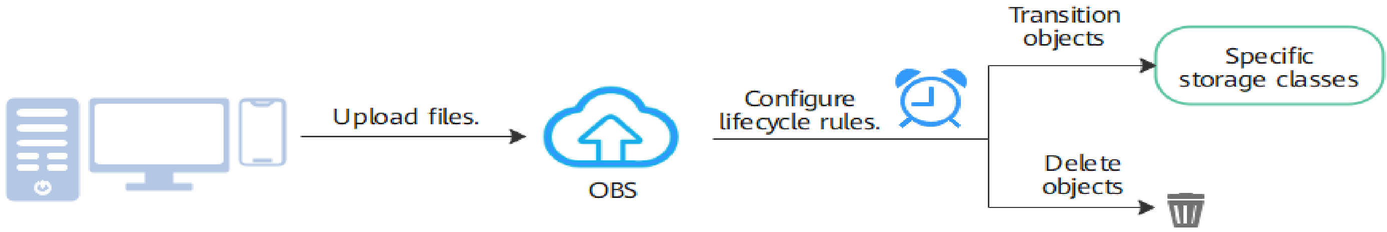


Table 9-1 Key actions in OBS object lifecycle management

Action	Scenarios	Objects Managed	Operation Guide
Transitions between storage classes	Periodically transition data that is frequently accessed over a period of time but may not be accessed after that period to the Archive or Deep Archive storage class to reduce storage costs. Such data includes digital media, financial and medical records, long-term database backups, and data retained for regulatory compliance.	Objects in a bucket (including the latest and historical versions of objects when versioning is enabled for the bucket)	**Transitioning Objects Using Lifecycle Rules**

Action	Scenarios	Objects Managed	Operation Guide
Periodical deletion of objects	Periodically delete data, such as log files, that needs to be retained for a period of time and can be deleted after that period.	● Objects in a bucket (including the latest and historical versions of objects when versioning is enabled for the bucket) ● Fragments **NOTE** In a multipart upload, a file is divided into multiple parts and then uploaded. After all parts are uploaded, you can make an API call to assemble the parts into a complete object. The parts that failed to be uploaded or assembled are called fragments. You can continue to execute the interrupted or failed multipart upload to remove fragments, or directly delete fragments to reduce storage costs. For more information about fragments, see **Managing Fragments**. ● Expired delete markers **NOTE** An expired delete marker is the only single delete marker that remains after all historical versions of an object are deleted. Removing expired delete markers helps improve performance.	**Deleting Objects Using Lifecycle Rules**

- Descriptions about the elements and syntax of lifecycle rules in **Lifecycle Configuration Elements**
- Lifecycle rule configuration procedure in **Creating a Lifecycle Rule**
- Lifecycle rule examples in **Lifecycle Configuration Examples**

Points in Time for Lifecycle Management

Below describes the key points in time involved in lifecycle management.

Last Modification Time of an Object

- An object uploaded to a bucket cannot be modified, so the object's last modification time is when the object was uploaded. Operations, such as changing the object storage class, ACL, metadata, and encryption method and appending data to an object, only change the metadata of an object.

- For parallel file systems, modifying and truncating a file will change the last modification time of the file.
- Uploading or copying an object in a bucket or a file in a parallel file system will change their last modification time if there is such an object or a file with the same name. If versioning is not enabled, the last modification time of the object or file is the time when the object or file was last uploaded. If versioning is enabled, a newly uploaded object or file becomes the current version, and the original one becomes a historical version. The last modification time of both the current version and the historical version is the time when the object or file was last uploaded.

Start Time of an Object's Lifecycle

After an object is last updated, OBS starts to calculate its lifecycle at the next 00:00 (UTC time). Assume an object was uploaded at 09:00 on June 1, 2024 (UTC). OBS would calculate its lifecycle from 00:00 on June 2, 2024 (UTC). If the object was configured to be deleted one day later, it would be deleted at 00:00 on June 3, 2024 (UTC).

Completion Time of Lifecycle Rule Actions

- It usually takes 24 hours at most for the actions in a lifecycle rule to complete. Considering the start time of the rule, there may be a delay in transitioning storage classes or deleting expired objects, but the total completion time will not exceed 48 hours.

  Assume an object was uploaded at 09:00 on June 1, 2024 (UTC). OBS would calculate its lifecycle from 00:00 on June 2, 2024 (UTC). If the object was configured to be deleted one day later, it would be deleted at 00:00 on June 3, 2024 (UTC). The execution latency does not exceed 24 hours and the deletion would be completed at 00:00 on June 4, 2024 (UTC).
- If you make changes to an existing lifecycle rule, the lifecycle task on the current day will be terminated and the lifecycle execution time may be prolonged. Therefore, do not frequently change a lifecycle rule. For example, if an object was uploaded at 20:20 on June 1, 2024 (UTC) and has a rule configured to delete it one day later, the rule would be deleted at 00:00 on June 4, 2024 (UTC). However, if the lifecycle rule was modified, for example, the rule name was changed, the lifecycle rule needs to be reloaded and executed again, the deletion may be complete later than 00:00 on June 4, 2024 (UTC).

9.1.2 Rules for Auto-deleting and Transitioning Objects Through Lifecycle Rules

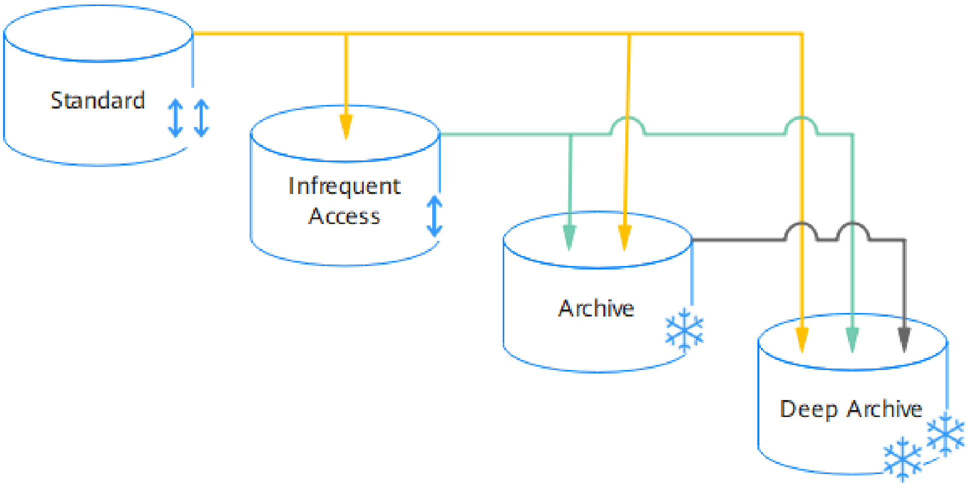
This section describes the rules for auto-deleting and transitioning objects through lifecycle rules.

Constraints

Because multi-AZ redundancy is not supported for Archive and Deep Archive storage, you cannot transition objects in a multi-AZ bucket to Archive or Deep Archive through lifecycle rules.

Transitioning Objects Using Lifecycle Rules

Figure 9-2 Transitioning objects using lifecycle rules



As shown in **Figure 9-2**, OBS allows you to use lifecycle rules to transition objects as follows:

- From Standard to Infrequent Access or Archive or Deep Archive
- From Infrequent Access to Archive or Deep Archive
- From Archive to Deep Archive

When a lifecycle rule is applied, OBS transitions objects based on the versioning state of your bucket.

Unversioned Bucket

Each object in an unversioned bucket has only one version. You can use lifecycle rules to transition such objects.

Table 9-2 Transitioning objects in an unversioned bucket

Action Type	Effect of Action	Timer Start Time
Transitioning between storage classes	The object is transitioned to the target storage class.	Time when an object is uploaded

NOTE

If versioning is not enabled for a bucket, there are no noncurrent versions in the bucket. Therefore, the NoncurrentVersionTransition field is invalid.

Versioning-enabled Bucket

You can use lifecycle rules to transition the current and noncurrent versions of objects in a versioning-enabled bucket.

Table 9-3 Transitioning objects in a versioning-enabled bucket

Action Type	Effect of Action	Timer Start Time
Transitioning the current version	● If the current version is not a delete marker version, it is transitioned to the target storage class. ● If the current version is a delete marker version, no action is taken.	Time when the current version is uploaded
Transitioning a noncurrent version	● If the noncurrent version is not a delete marker version, it is transitioned to the target storage class. ● If the noncurrent version is a delete marker version, no action is taken.	Time when the current version becomes a noncurrent version

Versioning-suspended Bucket

You can use lifecycle rules to transition the current and noncurrent versions of objects in a versioning-suspended bucket.

Table 9-4 Transitioning the current and noncurrent versions of objects in a versioning-suspended bucket

Action Type	Effect of Action	Timer Start Time
Transitioning the current version	● If the current version is not a delete marker version, it is transitioned to the target storage class. ● If the current version is a delete marker version, no action is taken.	Time when the current version is uploaded
Transitioning a noncurrent version	● If the noncurrent version is not a delete marker version, it is transitioned to the target storage class. ● If the noncurrent version is a delete marker version, no action is taken.	Time when the current version becomes a noncurrent version

Deleting Objects Using Lifecycle Rules

OBS uses lifecycle rules to delete objects based on the versioning state of your bucket.

Unversioned Bucket

If versioning is not enabled for a bucket, each object has only one version. You can use lifecycle rules to delete objects and fragments from this bucket.

Table 9-5 Deleting objects and fragments from an unversioned bucket

Action Type	Effect of Action
Deleting an object	The object is permanently deleted and cannot be restored.
Deleting fragments	The fragments are permanently deleted and cannot be restored.

NOTE

If versioning is not enabled for a bucket, the bucket does not have noncurrent versions or delete markers. Therefore, the NoncurrentVersionExpiration and ExpiredObjectDeleteMarker fields are invalid.

Versioning-enabled Bucket

You can use lifecycle rules to delete the current version, noncurrent versions, expired delete markers, and fragments of objects from a versioning-enabled bucket.

Table 9-6 Deleting objects and fragments from a versioning-enabled bucket

Action Type	Effect of Action
Deleting the current version of an object	● If the current version is not a delete marker version, a delete marker with a unique version ID is created. The current version becomes a noncurrent version, and the delete marker version becomes the current version. ● If the current version is a delete marker version, no action is taken.
Deleting a noncurrent version of an object	The noncurrent version is deleted and cannot be restored.
Deleting an expired delete marker	The expired delete marker is deleted. After all noncurrent versions of an object are deleted, the current delete marker version will be deleted to improve the List performance.
Deleting fragments	The fragments are permanently deleted and cannot be restored.

Versioning-suspended Bucket

You can use lifecycle rules to delete the current version, noncurrent versions, expired delete markers, and fragments of objects from a versioning-suspended bucket.

Table 9-7 Deleting objects and fragments from a versioning-suspended bucket

Action Type	Effect of Action
Deleting the current version of an object	● If the current version is not a delete marker version and the version ID is not null, the current version becomes a noncurrent version, and the delete marker version with a null ID becomes the current version. **NOTE** After versioning is suspended, a newly uploaded object will have the version ID of null. If the newly uploaded object has the same name as an existing object with a null version ID, this new object will overwrite the existing object. For details, see **Versioning**. ● If the current version is not a delete marker version and the current version ID is null, the delete marker version with the null ID overwrites the current version. The delete marker then becomes the current version, and the overwritten version cannot be recovered. ● If the current version is a delete marker version, no action is taken.
Deleting a noncurrent version of an object	The noncurrent version is deleted and cannot be restored.
Deleting an expired delete marker	The expired delete marker of the object is deleted. After all noncurrent versions of an object are deleted, the current delete marker version will be deleted to improve the List performance.
Deleting fragments	The fragments are permanently deleted and cannot be restored.

Overlapping Lifecycle Rules

Multiple lifecycle rules may overlap. For example:

- If a rule has no prefix specified, this rule overlaps with any other rules.
- If two rules have the same prefix, they overlap.
- If the prefix of a rule is the substring of another rule's prefix, the rules overlap.

If a rule to be created overlaps with an existing one, the rule must meet the following requirements or it will fail to be created:

- The rule uses either **Days** or **Date**.
- The actions on an object take place in the following order: Transition to Infrequent Access > Transition to Archive > Transition to Deep Archive > Delete the object upon expiration.
- The overlapping rules do not contain the same action, for example, deleting objects, transitioning to Infrequent Access, Archive, or Deep Archive storage.
- If there are two overlapping rules, one can be configured to delete the current version of an object and the other to delete the expired delete markers.

Precautions

- Encrypted objects remain encrypted when their storage class is transitioned.
- Minimum storage duration

  The minimum storage duration is the minimum billable storage duration. This means that objects will be billed for the minimum storage duration even if they are not stored for that long. For example, if an object is transitioned to Archive after being stored in Infrequent Access for 20 days, it will be billed for the storage of 30 days (the minimum storage duration for Infrequent Access).

Item	Standard	Infrequent Access	Archive	Deep Archive (in OBT)
Minimum storage duration	N/A	30 days	90 days	180 days

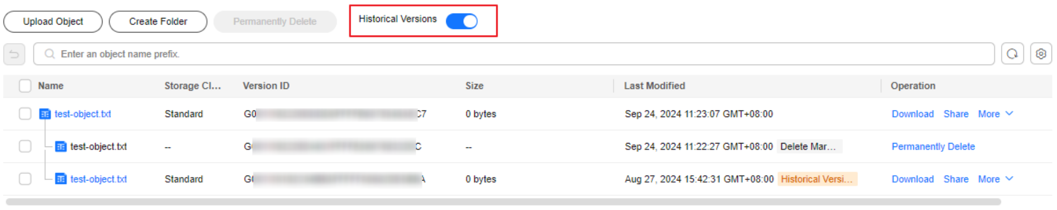
Related Operations

How Do I Check Whether an Object Version Has a Delete Marker?

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 Enable **Historical Version** to check whether the versions of the object have a delete marker.



Upload Object | Create Folder | Permanently Delete | Historical Versions

Enter an object name prefix.

Name	Storage Cl...	Version ID	Size	Last Modified	Operation
test-object.txt	Standard	G0[redacted]C7	0 bytes	Sep 24, 2024 11:23:07 GMT+08:00	Download Share More
test-object.txt	--	G[redacted]C	--	Sep 24, 2024 11:22:27 GMT+08:00 Delete Mar...	Permanently Delete
test-object.txt	Standard	G[redacted]A	0 bytes	Aug 27, 2024 15:42:31 GMT+08:00 Historical Versi...	Download Share More

----End

You can also use the API for **listing objects in a bucket** or the following SDKs to check whether an object version has a delete marker:

Java	**Python**	**C**	**Go**	**BrowserJS**	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**

9.1.3 Lifecycle Configuration Elements

The following uses an example to describe the elements of a lifecycle rule.

Elements of a lifecycle rule

A complete lifecycle rule consists of a range of elements that fall into metadata, filters, and actions (execution time included).

Metadata

Rule metadata describes the rule ID and rule status.

Table 9-8 Metadata elements

Element		Description
Metadata	ID	**Definition**: Rule ID, which uniquely identifies a rule. **Value range**: The value is a string ranging from 0 (exclusive) to 255 characters. It can contain only uppercase or lowercase letters, digits, periods (.), underscores (_), and hyphens (-). **Default value**: If the **ID** element is not used or is left blank, OBS automatically assigns a unique ID to the rule.
	Status	**Definition**: It indicates whether the rule is enabled or disabled. If the rule is disabled, OBS does not perform any actions defined in the rule. **Value range**: ● **Enabled**: The rule is enabled. ● **Disabled**: The rule is disabled. **Default value**: N/A

Filters

The object filter criteria of a rule describe the objects to which a lifecycle rule is applied.

Table 9-9 Filter description

Element		Description
Filter	Prefix	**Definition**: Object name prefix, which identifies the objects to which the lifecycle rule applies If you leave the **Prefix** element blank, the rule applies to all objects in the bucket. **Value range**: The value is a string ranging from 0 (exclusive) to 1024 characters. **Default value**: If this element is not specified, it is left blank by default.

Actions

The following table describes actions that will be performed on objects and when the actions will be performed.

Table 9-10 Elements to describe lifecycle actions

Element	Description	Sub-elements	Description
Expiration	Used to delete objects **NOTE** The **Expiration** action does not delete unassembled multipart uploads.	Days (Use only one of the **Days**, **Date**, and **ExpiredObjectDeleteMarker** elements.)	**Definition**: The number of days since the last object update, after which eligible objects will be expired and then deleted **Constraints**: ● **Days** applies only to the current version of an object. ● For the same object, the time specified for **Expiration** must be later than that for **Transition**. **Value range**: A positive integer, in days **Default value**: N/A

Element	Description	Sub-elements	Description
		Date (Use only one of the **Days**, **Date**, and **ExpiredObjectDeleteMarker** elements.)	**Definition**: The date, before which objects were last modified will be expired and then deleted **Constraints**: - **Date** applies only to the current version of an object. - The date must be compliant with ISO 8601 and its time must be 00:00:00 in UTC. If **Date** is set to **2018-01-01T00:00:00.000Z**, objects whose last modification time is earlier than **2018-01-01T00:00:00.000Z** will be deleted. - For the same object, the time specified for **Expiration** must be later than that for **Transition**. - OBS Console and OBS Browser+ do not support the deletion of objects on a specified date. **Default value**: N/A
	Used to remove expired delete markers	ExpiredObjectDeleteMarker (Use only one of the **Days**, **Date**, and **ExpiredObjectDeleteMarker** elements.)	**Definition**: Whether to remove expired delete markers **Value range**: - **true**: Expired delete markers are removed. - **false**: Expired delete markers are not removed. **Default value**: N/A

Element	Description	Sub-elements	Description
Transition	Used to transition objects between storage classes	Days (Use either **Days** or **Date**.)	**Definition**: The number of days since the last object update, after which eligible objects will be transitioned to another storage class **Constraints**: **Days** applies only to the current version of an object. **Value range**: Unit: day ● If only one transition is configured, the time should be at least one day after the update. ● If multiple transitions are configured, the transition to the Deep Archive storage class must be later than the transition to Archive, and the transition to Archive must be later than the transition to Infrequent Access. **Default value**: N/A

Element	Description	Sub-elements	Description
		Date (Use either **Days** or **Date**.)	**Definition**: The date, before which objects were last modified will be transitioned to another storage class **Constraints**: ● **Date** applies only to the current version of an object. ● The date must be compliant with ISO 8601 and its time must be 00:00:00 in UTC. If **Date** is set to **2018-01-01T00:00:00.000Z**, objects whose last modification time is earlier than **2018-01-01T00:00:00.000Z** will be transitioned to another storage class. ● If multiple transitions are configured, the transition to the Deep Archive storage class must be later than the transition to Archive, and the transition to Archive must be later than the transition to Infrequent Access. ● OBS Console and OBS Browser+ do not support the transition of objects on a specified date. **Default value**: N/A
		StorageClass	**Definition**: New storage class eligible objects will be transitioned to **Value range**: ● **WARM**: the Infrequent Access storage class ● **COLD**: the Archive storage class ● **DEEP_ARCHIVE**: the Deep Archive storage class **Default value**: N/A

Element	Description	Sub-elements	Description
AbortIncompleteMultipartUpload	Used to delete fragments	DaysAfterInitiation	**Definition**: The number of days since fragment generation, after which fragments will be expired and then deleted **Value range**: A positive integer, in days **Default value**: N/A
NoncurrentVersionExpiration	Used to delete historical object versions	NoncurrentDays	**Definition**: The number of days since the objects became noncurrent, after which noncurrent object versions will be deleted **Constraints**: ● **NoncurrentDays** only applies to historical object versions. ● **NoncurrentDays** can only be used for buckets that have versioning enabled or suspended. ● For the same historical version of an object, the time specified for **Expiration** must be later than that for **Transition**. **Value range**: A positive integer, in days **Default value**: N/A

Element	Description	Sub-elements	Description
NoncurrentVersionTransition	Used to transition noncurrent versions	NoncurrentDays	**Definition**: The number of days since the objects became noncurrent, after which noncurrent object versions will be transitioned to another storage class **Constraints**: ● **NoncurrentDays** only applies to historical object versions. ● **NoncurrentDays** can only be used for buckets that have versioning enabled or suspended. **Value range**: Unit: day ● If only one transition is involved, the time should be at least one day after the update. ● If multiple transitions are involved, the latter transition should be at least one day later than the former transition. ● If multiple transitions are configured, the transition to the Deep Archive storage class must be later than the transition to Archive, and the transition to Archive must be later than the transition to Infrequent Access. **Default value**: N/A
		StorageClass	**Definition**: New storage class eligible historical object versions will be transitioned to **Value range**: ● **WARM**: the Infrequent Access storage class ● **COLD**: the Archive storage class ● **DEEP_ARCHIVE**: the Deep Archive storage class **Default value**: N/A

NOTICE

In the same lifecycle rule, **Days** or **Date** must be consistently used. If you use **Days** for **Expiration**, you must also use **Days** (rather than **Date**) for **Transition**, and vice versa.

9.1.4 Creating a Lifecycle Rule

Application Scenarios

This section describes how to create a lifecycle rule.

Constraints

- A lifecycle rule can use prefixes but cannot use wildcards, suffixes, or regular expressions.
- A bucket can have an unlimited number of lifecycle rules, but the XML file of all its rules cannot exceed 20 KB.
- A maximum of 20 lifecycle rules can be configured for a parallel file system.
- If you want to modify the lifecycle rules of a bucket, you need to add rules. For example, if a bucket has already had Rule 1, you can add Rule 2 by performing the following operations:
  a. Call GetBucketLifecycle to obtain Rule 1.
  b. Add Rule 2.
  c. Call PutBucketLifecycle to update the lifecycle rule to Rule 1 and Rule 2.

  NOTE

  If you use OBS Console to update an existing lifecycle rule, you do not need to obtain the rule first. You just need to add a new rule and the new rule then automatically overwrites the existing one. For details, see **Configuring a Lifecycle Rule on OBS Console**.

Lifecycle Rule Configuration

You can configure lifecycle rules by using OBS Console, APIs, or SDKs.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 Click the bucket you want to configure a lifecycle rule.

Step 3 In the navigation pane, choose **Overview**.

Step 4 In the **Basic Configurations** area, click **Lifecycle Rules**.

Alternatively, you can choose **Basic Configurations** > **Lifecycle Rules** in the navigation pane.

Step 5 Click **Create**.

Figure 9-3 Creating a lifecycle rule

Step 6 Configure a lifecycle rule.

Basic Information:

- **Status**: Select **Enable** to enable this lifecycle rule.
- **Rule Name**: Enter a rule name that is no longer than 255 characters.
- **Prefix** (optional):
  - If this field is specified, objects with this prefix will be managed by the lifecycle rule. The prefix cannot start with a slash (/) or contain two consecutive slashes (//), and cannot contain special characters (\:*?"<>|).
  - If this field is not specified, all objects in the bucket will be managed by the lifecycle rule.

NOTE

Current Version or **Historical Version**:

NOTE

- **Current Version** and **Historical Version** are two concepts related to **Versioning**. If versioning is enabled for a bucket, uploading objects with the same name to the bucket creates different object versions. The last uploaded object is called the current version, while those previously uploaded are called historical versions. For more information, see **Versioning**.
- You can configure either **Current Version** or **Historical Version** or both of them.

- **Transition to Infrequent Access After (Days)**: After this number of days since the last update, objects meeting specified conditions will be transitioned to Infrequent Access.
- **Transition to Archive After (Days)**: After this number of days since the last update, objects meeting specified conditions will be transitioned to Archive. If you want to transition objects first to Infrequent Access and then to Archive, ensure that **Transition to Archive After (Days)** is larger than **Transition to Infrequent Access After (Days)**. If you want to only transition objects to Archive, there is no such a restriction.
- **Transition to Deep Archive After (Days)**: After this number of days since the last update, objects meeting specified conditions will be transitioned to Deep Archive. If you want to transition objects first to Infrequent Access and then to Deep Archive, ensure that **Transition to Deep Archive After (Days)** is larger than **Transition to Infrequent Access After (Days)**. If you want to transition objects first to Archive and then to Deep Archive, ensure that **Transition to Deep Archive After (Days)** is larger than **Transition to Archive After (Days)**. If you want to only transition objects to Deep Archive, there is no such a restriction.
- **Delete Objects After (Days)**: After this number of days since the last update, objects meeting specified conditions will be automatically deleted. This number must be an integer larger than that specified for any transition.
- **Delete Fragments After (Days)**: After this number of days since fragment generation, OBS will automatically delete fragments in the bucket.

Assume that you stored the following files in OBS on January 7, 2015:

- log/test1.log
- log/test2.log

- doc/example.doc
- doc/good.txt

Then, you stored the following files in OBS on January 10, 2015:

- log/clientlog.log
- log/serverlog.log
- doc/work.doc
- doc/travel.txt

On January 10, 2015, you set the objects prefixed with **log** to expire one day later. You might encounter the following situations:

- Objects **log/test1.log** and **log/test2.log** uploaded on January 7, 2015 would be deleted after the latest system scan on January 10, 2015 or January 11, 2015.
- Objects **log/clientlog.log** and **log/serverlog.log** uploaded on January 10, 2015 would be deleted on January 11, 2015 or January 12, 2015, depending on whether the objects have been stored for a whole day after the previous system scan. If the object has been stored for a whole day after the previous scan, it would be deleted during this scan. If the object has not been stored for a whole day after the previous scan, it would be deleted during the next scan.

Suppose you configured OBS to transition objects with the **log** prefix to Infrequent Access and Archive 30 days and 60 days after they are uploaded and delete the objects 100 days after they are uploaded. OBS would perform those actions on **log/clientlog.log**, **log/serverlog.log**, **log/test1.log**, and **log/test2.log** as you defined.

NOTE

The object storage time starts from the next UTC 00:00. A lifecycle rule becomes effective at most 24 hours after it is configured. There may be a delay in transitioning objects or deleting expired objects, but it does not exceed 48 hours. If you make changes to a lifecycle rule, the action time will be re-calculated.

Step 7 Click **OK** to complete the lifecycle rule configuration.

----End

Using APIs

Configuring Bucket Lifecycle Rules

Using SDKs

Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

Configuring a Lifecycle Rule for a Single Object

You can set the time for deleting an object when uploading it. If versioning is enabled, you can set the deletion time for each version of object when uploading it. An object lifecycle rule only affects object versions and not delete markers or fragments.

Unlike setting a lifecycle rule for a bucket, an object lifecycle rule only applies to that specific object. You can only schedule the deletion of the object, and not any storage class transitions. If a bucket lifecycle rule conflicts with an object lifecycle rule, the object lifecycle rule will apply.

OBS allows you to use only APIs and SDKs to set the lifecycle rule for an object.

Using APIs

Specify the **x-obs-expires** header if you upload an object using **PUT** (streaming upload) or **POST** (browser-based upload).

Using SDKs

The following OBS SDKs provides APIs for you to upload objects and configure the **expires** parameter for object deletion upon expiration:

- **Java**
- **Python**
- **C**
- **Go**
- **BrowserJS**
- **.NET**
- **Android**
- **iOS**
- **PHP**
- **Node.js**

9.1.5 Lifecycle Configuration Examples

If you use APIs or SDKs, refer to the XML example. If you use the OBS Console or OBS Browser+, refer to the screenshot.

Specifying a Rule with a Single Filter Condition

Example 1: Applying a lifecycle rule to all objects in a bucket

To apply the rule to all objects in the bucket, leave the prefix blank. Objects will be transitioned to the Archive storage class 100 days after creation.

XML:

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Prefix></Prefix>
        <Status>Enabled</Status>
        <Transition>
            <Days>100</Days>
            <StorageClass>COLD</StorageClass>
        </Transition>
    </Rule>
</LifecycleConfiguration>
```

Example 2: Specifying a filter by object name prefix

The following lifecycle rule specifies prefix **texta/** as the filter. It applies to objects with the **texta/** prefix, such as **texta/file1.txt** and **texta/file2.txt**.

This rule specifies two actions: transitioning objects to the Infrequent Access storage class 90 days after creation and deleting objects 120 days after creation.

XML:

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Status>Enabled</Status>
        <Prefix>texta/</Prefix>
        <Transition>
            <Days>90</Days>
            <StorageClass>WARM</StorageClass>
        </Transition>
        <Expiration>
            <Days>120</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

Example 3: Specifying a filter by object name prefix and deleting the specified objects that was last modified before the specified date

The following lifecycle rule specifies prefix **texta/** as the filter. It applies to objects with the **texta/** prefix, such as **texta/file1.txt** and **texta/file2.txt**.

This rule specifies one action: deleting objects that were last modified before May 30, 2024.

XML:

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Status>Enabled</Status>
        <Prefix>texta/</Prefix>
        <Expiration>
            <Date>2024-05-30T00:00:00.000Z</Date>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

Example 4: Specifying a filter by object name prefix and transitioning the specified objects that was last modified before the specified date

The following lifecycle rule specifies prefix **texta/** as the filter. It applies to objects with the **texta/** prefix, such as **texta/file1.txt** and **texta/file2.txt**.

This rule specifies one action: transitioning objects that were last modified before May 30, 2024 to the Archive storage class.

XML:

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Status>Enabled</Status>
        <Prefix>texta/</Prefix>
        <Transition>
            <Date>2024-05-30T00:00:00.000Z</Date>
            <StorageClass>COLD</StorageClass>
        </Transition>
    </Rule>
</LifecycleConfiguration>
```

Specifying Multiple Rules

If you want objects to have different lifecycle actions, you can specify multiple rules. The following lifecycle configuration has two rules:

1. Rule 1 applies to objects with the **texta/** prefix. It directs OBS to transition objects to the Archive storage class 120 days after creation and to delete them 360 days after creation.
2. Rule 2 applies to objects with the **textb/** prefix. It directs OBS to transition objects to the Infrequent Access storage class 90 days after creation and to delete them 120 days after creation.

XML:

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule1</ID>
        <Prefix>texta/</Prefix>
        <Status>Enabled</Status>
        <Transition>
            <Days>120</Days>
            <StorageClass>COLD</StorageClass>
        </Transition>
        <Expiration>
            <Days>360</Days>
        </Expiration>
    </Rule>
    <Rule>
        <ID>sample-rule2</ID>
        <Prefix>textb/</Prefix>
        <Status>Enabled</Status>
        <Transition>
            <Days>90</Days>
            <StorageClass>WARM</StorageClass>
        </Transition>
        <Expiration>
            <Days>120</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

Specifying Multiple Rules with Overlapping Filter Conditions

NOTE

For details about overlapping prefixes and conflicting actions, see **Overlapping Lifecycle Rules**.

Example 1: Overlapping prefixes (no conflict)

- **Rules**: Rule 1 specifies an empty filter (indicating all objects in the bucket). It lets OBS delete all objects 120 days after creation. Rule 2 specifies the **typea/** prefix (indicating objects with the **typea/** prefix). It lets OBS transition objects to the Archive storage class 90 days after creation.

- **Result**: There are no conflicting lifecycle actions. Objects with the **typea/** prefix are transitioned to the Archive storage class 90 days after creation and then deleted 120 days after creation.

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule1</ID>
        <Prefix></Prefix>
        <Status>Enabled</Status>
        <Expiration>
            <Days>120</Days>
        </Expiration>
    </Rule>
    <Rule>
        <ID>sample-rule2</ID>
        <Prefix>typea/</Prefix>
        <Status>Enabled</Status>
        <Transition>
            <Days>90</Days>
            <StorageClass>COLD</StorageClass>
        </Transition>
    </Rule>
</LifecycleConfiguration>
```

Example 2: Overlapping prefixes resulting in conflicting lifecycle actions

- **Rules**: Rule 1 specifies an empty filter (indicating all objects in the bucket). It lets OBS delete all objects 90 days after creation. Rule 2 specifies the **typea/** prefix (indicating objects with the **typea/** prefix). It lets OBS transition objects to the Archive storage class 120 days after creation.
- **Result**: The lifecycle actions conflict. Such configuration is not allowed.

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule1</ID>
        <Prefix></Prefix>
        <Status>Enabled</Status>
        <Expiration>
            <Days>90</Days>
        </Expiration>
    </Rule>
    <Rule>
        <ID>sample-rule2</ID>
        <Prefix>typea/</Prefix>
        <Status>Enabled</Status>
        <Transition>
            <Days>120</Days>
            <StorageClass>COLD</StorageClass>
        </Transition>
    </Rule>
</LifecycleConfiguration>
```

Deleting Fragments

The following lifecycle rule specifies prefix **texta/** as the filter. It applies to objects with the **texta/** prefix, such as **texta/file1.txt** and **texta/file2.txt**. This rule lets OBS delete object fragments 10 days after generation.

```
</LifecycleConfiguration>
   <Rule>
        <ID>sample-rule</ID>
        <Prefix>texta/</Prefix>
        <Status>Enabled</Status>
        <AbortIncompleteMultipartUpload>
            <DaysAfterInitiation>10</DaysAfterInitiation>
        </AbortIncompleteMultipartUpload>
   </Rule>
</LifecycleConfiguration>
```

Specifying a Lifecycle Rule for a Versioning-Enabled/Suspended Bucket

Example 1: Periodically transitioning and deleting historical object versions

The following lifecycle rule specifies prefix **prefix1/** as the filter. It applies to objects with the **prefix1/** prefix, such as **prefix1/file1.txt** and **prefix1/file2.txt**.

This rule specifies two lifecycle actions: transitioning historical object versions to the Archive storage class 20 days after generation and deleting them 30 days after generation.

Example 2: Removing expired delete markers

The following lifecycle rule specifies prefix **prefix1/** as the filter. It applies to objects with the **prefix1/** prefix, such as **prefix1/file1.txt** and **prefix1/file2.txt**.

If all versions of an object with the **prefix1/** prefix have been deleted and only one expired delete marker remains, this rule lets OBS remove this expired delete marker.

```
</LifecycleConfiguration>
  <Rule>
      <ID>sample-rule</ID>
      <Prefix>prefix1/</Prefix>
      <Status>Enabled</Status>
      <Expiration>
          <ExpiredObjectDeleteMarker>true</ExpiredObjectDeleteMarker>
      </Expiration>
   </Rule>
</LifecycleConfiguration>
```

Disabling a Lifecycle Rule

The following configuration disables a lifecycle rule. This rule applies to objects with the **texta/** prefix and transitions objects to the Archive storage class 100 days after creation.

XML:

```
<LifecycleConfiguration>
   <Rule>
      <ID>sample-rule</ID>
      <Status>Disabled</Status>
      <Prefix>texta/</Prefix>
      <Transition>
          <Days>100</Days>
          <StorageClass>COLD</StorageClass>
      </Transition>
   </Rule>
</LifecycleConfiguration>
```

9.2 Bucket Inventories

Scenarios

A bucket inventory can list objects in a bucket, save the related object information in CSV files, and deliver the CSV files to the bucket specified for storing bucket inventory files. In this manner, you can easily manage objects in a bucket. A source bucket can also be the destination bucket.

- A bucket inventory file can contain the following object related information: versions, sizes, storage classes, tags, encryption statuses, and last modification time.
- You can encrypt bucket inventory files in the SSE-KMS mode.
- You can set the frequency (daily or weekly) for generating bucket inventory files.
- You can also specify a bucket to store the generated bucket inventory files.

Constraints

Bucket versions

- Inventories can be generated only for OBS 3.0 buckets, but they can be stored in either OBS 3.0 or OBS 2.0 buckets.

Number of bucket inventories

- A bucket can have a maximum of 10 inventories.

Source and destination buckets

- The source bucket (for which a bucket inventory rule is configured) and the destination bucket (where the generated inventory files are stored) must belong to the same account.
- The source and destination buckets must be in the same region.
- The destination bucket cannot have server-side encryption enabled.

Functions

- Inventory files must be in the CSV format.
- Inventories can apply to all objects in a bucket or a set of objects with the same name prefix.
- Inventory rules in the same bucket cannot overlap.
  - If there is already an inventory rule for all objects in the bucket, any other inventory rule with an object name prefix specified cannot be created. To create a rule for only a set of objects, first delete the inventory rule configured for all objects.
  - As long as there is an inventory rule for a set of objects, a rule for all objects in the bucket cannot be created. To create a rule for all objects, first delete all inventory rules that match objects by prefix.
  - If a bucket already has an inventory rule that filters objects by the object name prefix **ab**, the filter of a new inventory rule cannot start with **a** or **abc**. To create such a rule, you need to first delete the existing inventory rule that conflicts with the rule you will create.
- Only SSE-KMS can be used to encrypt bucket inventories.

Permissions

- Inventory files are uploaded to the destination bucket by an OBS system user, so you must grant this user the write permission for the bucket. That is, the destination bucket must contain a policy with the value of **{"Service": "obs"}** for **Principal**. For details, see 1.

Others

- The bucket inventory function is offered for free, but inventory files are billed for the storage space they occupy.
- The bucket inventory function is not available for federated users.

Configuring a Bucket Inventory

Before the configuration, you need to briefly understand what a source bucket or a destination bucket is.

- Source bucket: A source bucket is the bucket for which an inventory is configured. The inventory lists objects stored in the source bucket.
- Destination bucket: A destination bucket is where generated inventory files are stored. A source bucket can also be the destination bucket. You can specify a name prefix for an inventory. Then generated inventory files will be named with the prefix and saved in the directory with the prefix. If you do not specify any name prefix for the inventory, the generated inventory files are stored in the root directory of the bucket.
  - Restrictions on the destination bucket
    - The destination bucket and source bucket must belong to the same tenant.
    - The destination bucket and source bucket must be in the same region.
    - A bucket policy must be configured to grant OBS the permission to write objects to the destination bucket. For details, see **Add a bucket policy for the destination bucket**.
  - The destination bucket contains the following files:
    - A list of inventory files
    - The **Manifest** file, which contains the list of all inventory files under a certain inventory configuration. For details about the **Manifest** file, see **Manifest File**.

Configuring a Bucket Inventory

You use OBS Console or call the API to configure a bucket inventory. If you configure a bucket inventory on OBS Console, a bucket policy with the required permission configuration is automatically generated for the destination bucket. If you call the API to configure the bucket inventory, you need to manually configure the bucket policy for the destination bucket.

1. Add a bucket policy for the destination bucket.

   A bucket policy must be configured for the destination bucket, to grant the OBS system users the permission to write objects to the destination bucket. The format of the bucket policy is as follows. Replace **destbucket** with the actual name of the destination bucket.

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

2. Configure a bucket inventory.

   We provide multiple tools to configure a bucket inventory. For details, see **Bucket Inventories**.

Content in an Inventory File

The content in an inventory file can be configured when creating the inventory. For details about all possible fields, see **Table 9-11**.

Table 9-11 Object metadata listed in an inventory file

Metadata	Description
Bucket	Name of the source bucket
Key	The name of an object. Each object in a bucket has a unique key. (Object names in the inventory file are URL-encoded using UTF-8 character set and can be used only after being decoded.)
VersionId	Version ID of an object. If the value of **IncludedObjectVersions** in the inventory configuration is **Current**, this field is not included in the inventory file.
IsLatest	If the object version is the latest, this parameter is **True**. (If the value of **IncludedObjectVersions** in the inventory configuration is **Current**, this field is not included in the inventory file.)
IsDeleteMarker	When versioning is enabled for the source bucket, if an object is deleted, a new object metadata is generated for the object, and the **IsDeleteMarker** of the metadata is set to **true**. (If the value of **IncludedObjectVersions** in the inventory configuration is **Current**, this field is not included in the inventory file.)
Size	Object size, in bytes
LastModifiedDate	Object creation date or last modification date
ETag	Hexadecimal digest of the object MD5. ETag is the unique identifier of the object content. It can be used to identify whether the object content is changed. For example, if ETag value is **A** when an object is uploaded and the ETag value has changed to **B** when the object is downloaded, it indicates that the object content is changed.
StorageClass	Storage class of an object

Metadata	Description
IsMultipartUploaded	Indicates whether an object is uploaded in the multipart mode.
ReplicationStatus	Cross-region replication status of an object
EncryptionStatus	Encryption status of an object

Inventory File Name

The name of an inventory file is in the following format:

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

- **destinationPrefix**: The inventory file name prefix configured when creating the inventory rule. Inventory files generated under the rule are named after the prefix, which can facilitate the classification of inventory files. If no prefix is specified, the default prefix is **BucketInventory**.
- **sourceBucketName**: Name of the source bucket for which an inventory is configured. This field can be used to differentiate inventory files of different source buckets, if those inventory files are saved in the same destination bucket.
- **inventoryId**: If a source bucket has multiple inventory rules whose inventory files are saved in the same destination bucket, this field can be used to identify different inventory rules.
- **yyyy-MM-dd'T'HH-mm'Z'**: Start time and date for scanning the destination bucket when an inventory file is generated. Objects uploaded to the source bucket after this time may not be listed in the inventory file.
- **UUID_index.csv**: one of the inventory files

Manifest File

If there are a large number of objects in a bucket, multiple inventory files may be generated for a single inventory configuration. It takes some time to generate these files. For example, if there are 200,000 objects in a bucket, it takes about 1.5 minutes to generate all inventory files. One or two hours after all inventory files are generated, a **manifest.json** file will be generated. The **manifest.json** file contains information about all inventory files generated this time, including:

- **sourceBucket**: name of the source bucket
- **destinationBucket**: name of the destination bucket
- **version**: version of the inventory
- **fileFormat**: format of inventory files
- **fileSchema**: object metadata fields contained in the inventory files
- **files**: list of all inventory files
- **key**: inventory file name
- **size**: size of an inventory file, in bytes
- **inventoriedRecord**: number of records contained in an inventory file

The following is an example of a simple **manifest.json** file.

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

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

The name of a **manifest** file is as follows (for details about each field, see **Inventory File Name**):

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

symlink.txt File

The **symlink.txt** file records the path of an inventory file. It helps quickly find all inventory files in big data scenarios. Apache Hive is compatible with the **symlink.txt** file. Hive can automatically find the **symlink.txt** file and the inventory files recorded in it.

The name of the **symlink.txt** file is as follows (for details about each field, see **Inventory File Name**):

```
destinationPrefix/sourceBucketName/inventoryId/hive/dt=YYYY-MM-DD-00-00/symlink.txt
```

Ways to Configure a Bucket Inventory

You can use OBS Console, APIs, or SDKs to configure a bucket inventory.

Using OBS Console

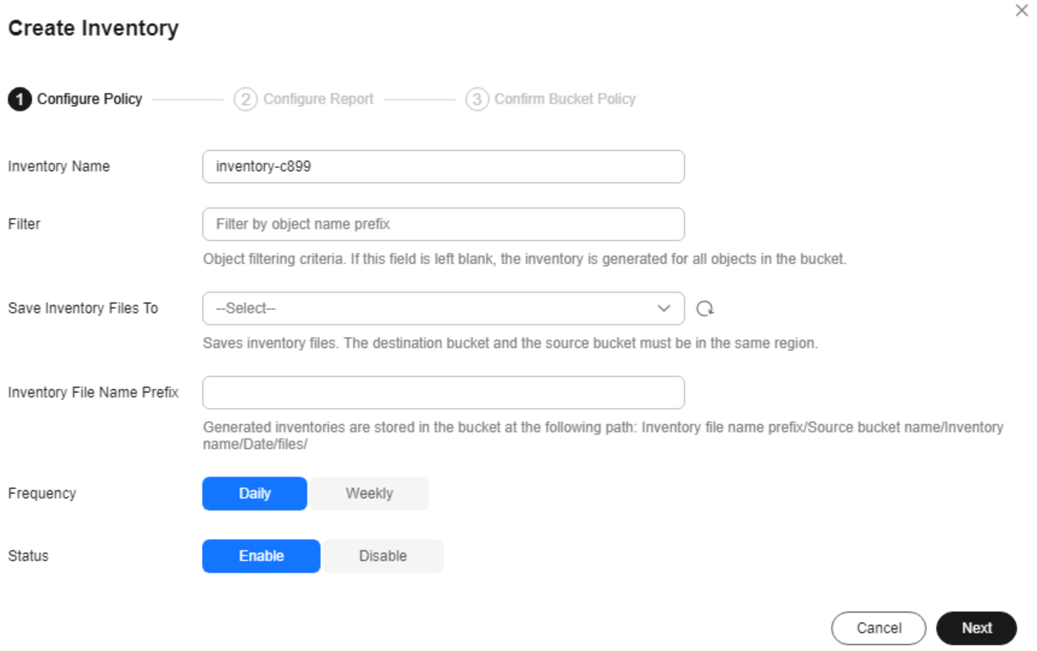
Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, click **Inventories**. The inventory list is displayed.

Step 4 Click **Create**. The **Create Inventory** dialog box is displayed.

Figure 9-4 Inventory settings



Create Inventory

1 Configure Policy — 2 Configure Report — 3 Confirm Bucket Policy

Inventory Name: inventory-c899

Filter: Filter by object name prefix

Object filtering criteria. If this field is left blank, the inventory is generated for all objects in the bucket.

Save Inventory Files To: --Select--

Saves inventory files. The destination bucket and the source bucket must be in the same region.

Inventory File Name Prefix:

Generated inventories are stored in the bucket at the following path: Inventory file name prefix/Source bucket name/Inventory name/Date/files/

Frequency: Daily Weekly

Status: Enable Disable

Step 5 Configure required parameters.

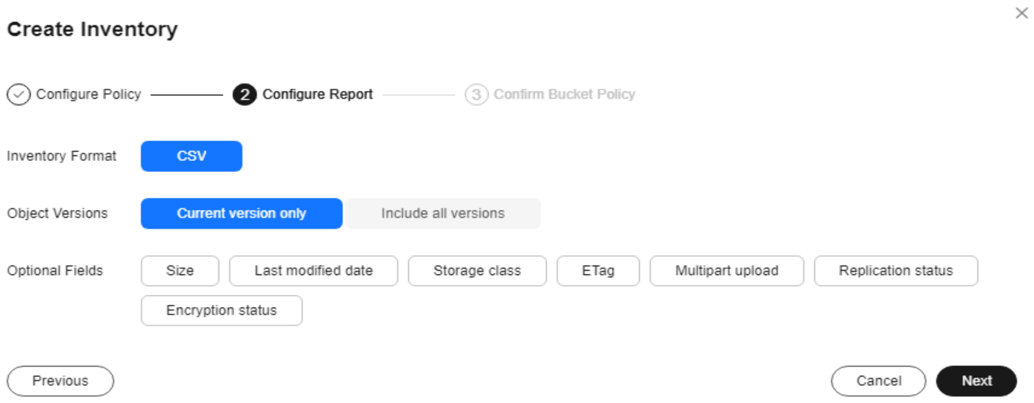
Table 9-12 Parameters for configuring a bucket inventory

Parameter	Description
Inventory Name	Name of a bucket inventory
Filter	Filter of an inventory. You can enter an object name prefix for OBS to create an inventory for objects with the specified prefix. Currently, only a prefix can be used as a filter. If the filter is not specified, the inventory covers all objects in the bucket. If a bucket has multiple inventories, their filters cannot overlap with each other.
Save Inventory Files To	Select a bucket (destination bucket) for saving generated inventory files. This bucket must be in the same region as the source bucket.
Inventory File Name Prefix	Prefix of the inventory file path. An inventory file will be saved in the following path: *Inventory file name prefix/Source bucket name/ Inventory name/Date and time/files/*. If this parameter is not specified, OBS automatically adds **BucketInventory** as the prefix to inventory file's path.
Frequency	How frequently inventory files are generated. It can be set to **Daily** or **Weekly**.

Parameter	Description
Status	Inventory status. You can enable or disable the generation of inventories.

Step 6 Click **Next** to go to the **Configure Report** page.

Figure 9-5 Configuring the report



Create Inventory

Inventory Format: CSV

Object Versions: Current version only (selected) / Include all versions

Optional Fields: Size, Last modified date, Storage class, ETag, Multipart upload, Replication status, Encryption status

Step 7 Configure the report.

Table 9-13 Report related parameters

Parameter	Description
Inventory Format	Inventory files can only be saved in CSV format.
Object Versions	Object versions that you want to list in an inventory file. It can be set to **Current version only** or **Include all versions**.
Optional Fields	Object information fields that can be contained in an inventory file, including **Size**, **Last modified date**, **Storage class**, **ETag**, **Multipart upload**, **Encryption status**, **Replication status**. For details about the fields, see **Metadata in an Inventory File**.

Step 8 Click **Next** to confirm the bucket policy.

OBS then automatically creates a bucket policy on the destination bucket to grant OBS permission to write inventory files to the bucket.

Step 9 Click **OK**.

----End

Using the API

Configuring a Bucket Inventory

Using SDKs

Java	Python: not supported	C: not supported	Go: not supported	BrowserJS: not supported	.NET: not supported	Android: not supported	iOS: not supported	PHP: not supported	Node.js: not supported

9.3 Event Notifications

NOTE

Event notifications have been taken offline. If you have any questions, **submit a service ticket**.

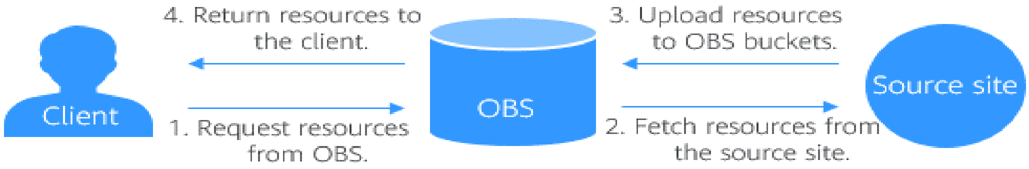
9.4 Back to Source

When a client does not access the requested data in OBS, the 404 error is returned. However, OBS provides the back-to-source function to help you obtain the requested data from its source site if it is not found in OBS.

Back-to-Source by Mirroring

If a mirroring back-to-source rule is configured for an OBS bucket and the requested data is not found in the bucket, the system will retrieve the data, when the back-to-source rule applies to the data, from the origin server, upload it to the bucket, and then return it to the requesting client. This process does not interrupt services. Therefore, you can use this function to seamlessly migrate data from the origin server to OBS, or migrate services to OBS without being sensed by users, at low costs. **Figure 9-6** illustrates the mirroring back-to-source process.

Figure 9-6 Back-to-source by mirroring


4. Return resources to the client.
Client
1. Request resources from OBS.
OBS
3. Upload resources to OBS buckets.
2. Fetch resources from the source site.
Source site

Constraints

Bucket version:

- Only buckets of version 3.0 and later support the mirroring back-to-source function.

Time:

- Any changes to a mirroring back-to-source rule take about five minutes to take effect.

Regions

Back to source is only available in some regions. For details, see **Function Overview**.

Rule count:

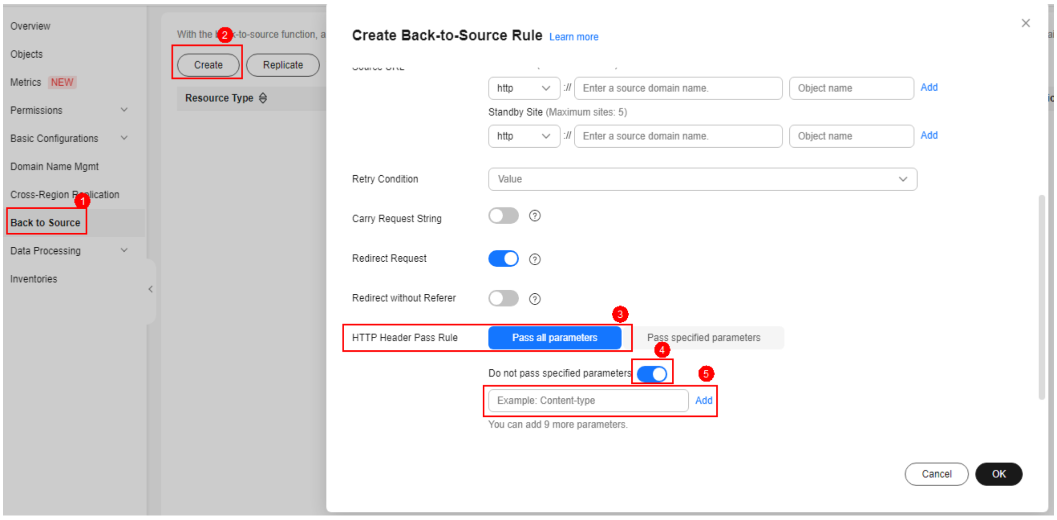
- A maximum of 10 mirroring back-to-source rules can be configured for a bucket.

Function:

- Anonymous users cannot configure mirroring back-to-source rules for a bucket.
- Parallel file systems do not support mirroring back-to-source rules.
- Static website hosting does not support the mirroring back-to-source function.
- A bucket cannot mirror itself.
- Currently, mirroring back to source from private buckets is supported for only some cloud vendors.
- A source cannot transfer data in **Transfer-Encoding: chunked** mode. The response to the request for downloading an object from the source must contain the **Content-Length** header to specify the size of the source object.

  To forbid **Transfer-Encoding: chunked**, in the **Create/Edit Back-to-Source Rule** window, specify **Accept-Encoding** for **Do not pass specified parameters** under the **Pass all parameters** option of **HTTP Header Pass Rule**.

  Figure 9-7 Configuration method

- If Ngnix was deployed as a reverse proxy for your origin server, turn off **chunked_transfer_encoding** of Ngnix.

```
location / {
    chunked_transfer_encoding off;
}
```

Permissions:

- To configure, obtain, or delete mirroring back-to-source rules, you must have the **Tenant Administrator** permission assigned by using IAM.
- Mirroring-based back to source requires that you create an IAM agency to delegate OBS to pull data from its origin server. The permissions the agency should assign to OBS include **obs:object:PutObject**, **obs:object:GetObject**, **obs:bucket:ListBucket**, and **obs:object:AbortMultipartUpload**.
- If SSE-KMS is enabled for a bucket, the **kms:cmk:get**, **kms:cmk:list**, **kms:cmk:create**, **kms:dek:create**, **kms:dek:crypto**, and **kms:dek:crypto** permissions must be configured for the IAM agency for OBS.

Other:

- The mirroring back-to-source function is free.
- An object cannot match two different mirroring back-to-source rules.

Creating a Mirroring Back-to-Source Rule

You can use OBS Console or APIs to create a mirroring back-to-source rule.

Using OBS Console

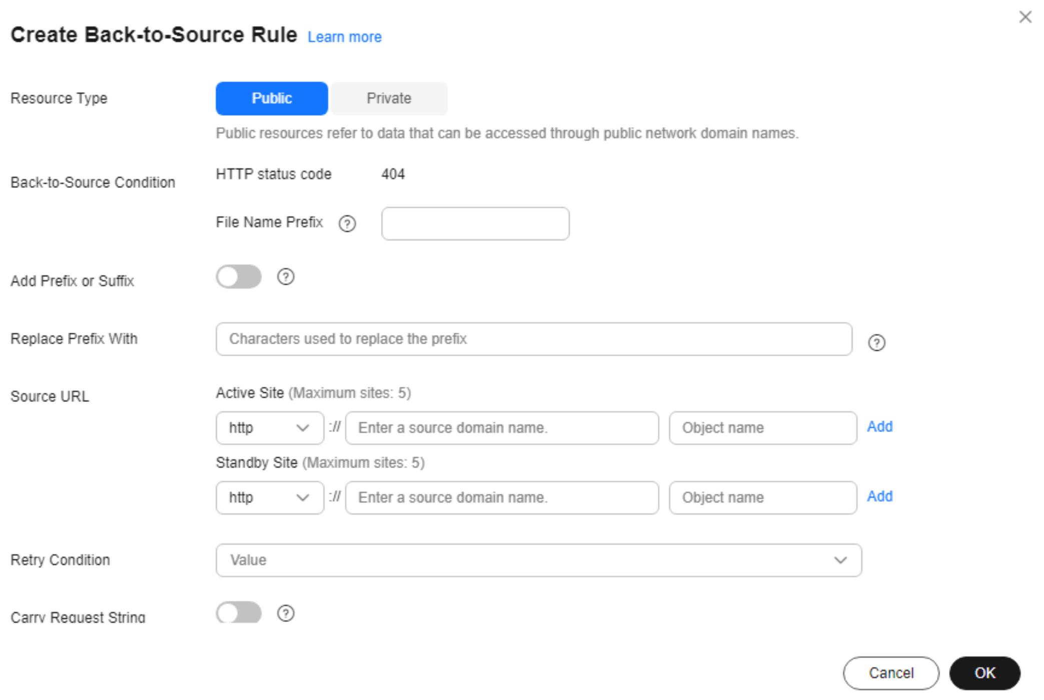
Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Back to Source**. The back-to-source rule list is displayed.

Step 4 Click **Create**.

Figure 9-8 Creating a Mirroring Back-to-Source Rule



Create Back-to-Source Rule Learn more

Resource Type: Public | Private
Public resources refer to data that can be accessed through public network domain names.

Back-to-Source Condition: HTTP status code 404
File Name Prefix

Add Prefix or Suffix

Replace Prefix With: Characters used to replace the prefix

Source URL:
Active Site (Maximum sites: 5)
http :// Enter a source domain name. Object name Add
Standby Site (Maximum sites: 5)
http :// Enter a source domain name. Object name Add

Retry Condition: Value

Carry Request String

Cancel OK

Step 5 Configure a mirroring back-to-source rule by referring to the parameters listed in **Table 9-14**.

Table 9-14 Parameters in a mirroring back-to-source rule

Parameter	Description
Resource Type	Type of the resources at the source site (origin server). ● **Public**: The origin server data comes from public object storage. ● **Private**: The origin server data comes from private object storage of some cloud vendors.
Back-to-Source Condition	Conditions that trigger the back-to-source rule. A mirroring back-to-source rule is triggered when the following conditions are met: The requested object starts with the specified file name prefix, and an HTTP status code 404 is returned because the object is not found in the bucket. The specified file name prefix: ● Cannot exceed 1,023 characters. ● Cannot contain or overlap with any other file name prefix specified for an existing rule. ● Can be left blank, which means that the rule applies to all files that do not meet the conditions of other back-to-source rules configured for the bucket. A bucket can have only one back-to-source rule that does not have a file name prefix specified. For example, if the file name prefix is set to **123/**, the rule is triggered when the **123/456.txt** file is requested but not available in the bucket.
Add Prefix or Suffix	When OBS requests data from the source site, the prefix or suffix is added in front of or after the name of the requested object. However, the object returned to OBS and the client keeps its original name without the added prefix or suffix. Example: A client requests **abc.txt** from OBS, which triggers the back-to-source rule. If the specified prefix is **123**, OBS then retrieves **123abc.txt** from the source site. However, the object is still downloaded as **abc.txt** to OBS and then returned to the client.

Parameter	Description
Replace Prefix With	OBS uses the specified prefix to replace the file name prefix set in the back-to-source condition when it requests data from the source site. However, the object returned to the client keeps the original prefix in its name. Example: The file name prefix is set to **123** as the back-to-source condition and the replacement prefix is set to **abc**. When the client requests **123456.txt**, the back-to-source rule is triggered. Then OBS requests **abc456.txt** from the source site. However, the obtained object is still saved as **123456.txt** in OBS and returned to the client.
Source URL	Source site address. You can set active sites and standby sites. The active site address is preferentially used during the back-to-source process. If multiple active site addresses are configured, all active sites are accessed in polling mode. If two or more active site addresses are configured, when the first request to an active address fails and the retry conditions are met, the request will retry another active site address. Configure at least one active site. Up to five active sites are supported. If you fail to retrieve content from all active sites, the request will try standby sites. Format: *http(https)://source domain name/static path* ● The source domain name is the domain name of the source site. – If the source site is a bucket that can be accessed over HTTP, the address is the bucket domain name. – If the source site is a private bucket provided by other cloud vendors, the address is the region domain name. At present, only private buckets of some cloud vendors are supported. ● The static path indicates the directory that stores the target file. For example, if the static path is **123/**, the target file is in the **123/** directory.
Retry Condition	Condition when a retry is triggered. **4XX** and a specific error code starting with **4** cannot be configured together. This rule works the same for **5XX** and an error code starting with **5**. A maximum of 20 error codes can be configured.
Carry Request String	When this function is enabled, query parameters in the request URL are passed to the source site.

Parameter	Description
Redirect Request	When this function is enabled, the request will follow the 3*xx* redirection response, if redirection is configured for the source site, to fetch the requested resource and save the resource to OBS. A request can follow a maximum of 10 redirections.
Redirect without Referer	With this function enabled, if redirection has been configured for the origin server, the Referer header in the request will be filtered out during redirecting.
HTTP Header Pass Rule	You can specify the HTTP header parameters that can be passed to the source site when a request sent to OBS triggers the mirroring back-to-source rule. **References** provides a configuration example and lists the HTTP headers that are supported and not supported. ● **Pass all parameters/Pass specified parameters**: Set the HTTP header parameters that can be passed. ● **Do not pass specified parameters**: Set the HTTP header parameters that cannot be passed. In this case, OBS does not pass the specified headers to the source site. If a header is specified for both the pass and do-not-pass categories, it is deemed as a do-not-pass parameter. ● **Configure custom parameters**: You can set a custom value for a specified header. If a client request carries this header, OBS changes the header value to the custom value before passing it to the source site.
IAM Agency	Mirroring-based back to source requires an IAM agency to delegate OBS to pull data from its origin server. The agency must grant OBS the **obs:object:PutObject**, **obs:object:GetObject**, **obs:bucket:ListBucket**, and **obs:object:AbortMultipartUpload** permissions. If no such an IAM agency is available, you can create one by referring to **Creating an Agency for Back to Source**.

Step 6 Click **OK**.

----End

Using the API

Creating a Mirroring Back-to-Source Rule

Replicating Mirroring Back-to-Source Rules

You can only use OBS Console to replicate a mirroring back-to-source rule.

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Back to Source**. The back-to-source rule list is displayed.

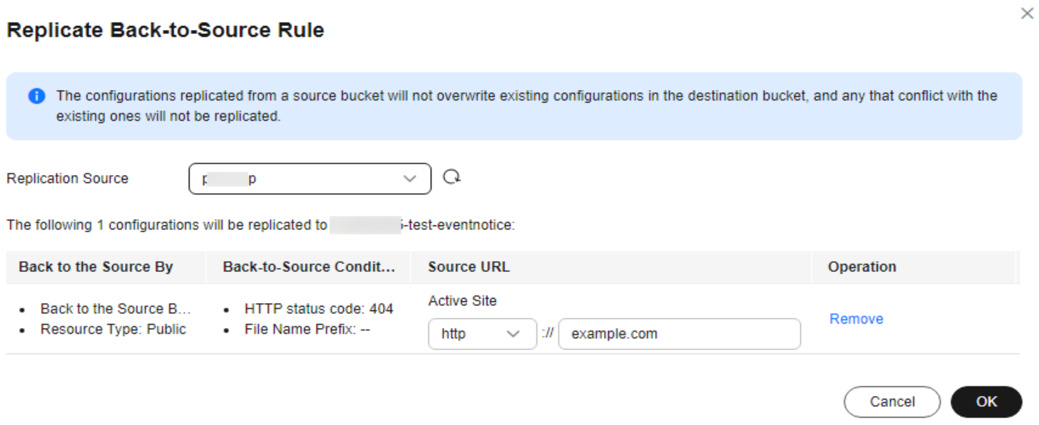
Step 4 Click **Replicate**.

Step 5 Select a replication source, which is bucket whose back-to-source rules you want to replicate.

NOTE

- The back-to-source rules replicated from a source bucket will not overwrite existing rules in the destination bucket, and any that conflict with the existing ones will not be replicated.
- The version of both source and destination buckets must be 3.0.
- Before replication, you can change the source URL. For the source URL configuration, see **Table 9-14**.
- You can remove the rules that you do not want to replicate.
- There can be five back-to-source rules at most in a bucket. If the number of rules you will replicate plus the number of existing rules in the destination bucket exceeds five, the replication will fail. Before replicating the rules, delete some if necessary.

Figure 9-9 Replicating back-to-source rules



Replicate Back-to-Source Rule

The configurations replicated from a source bucket will not overwrite existing configurations in the destination bucket, and any that conflict with the existing ones will not be replicated.

Replication Source: p[redacted]p

The following 1 configurations will be replicated to [redacted]-test-eventnotice:

Back to the Source By	Back-to-Source Condit...	Source URL	Operation
• Back to the Source B... • Resource Type: Public	• HTTP status code: 404 • File Name Prefix: --	Active Site http :// example.com	Remove

Step 6 Click **OK** to replicate the rules to the destination bucket.

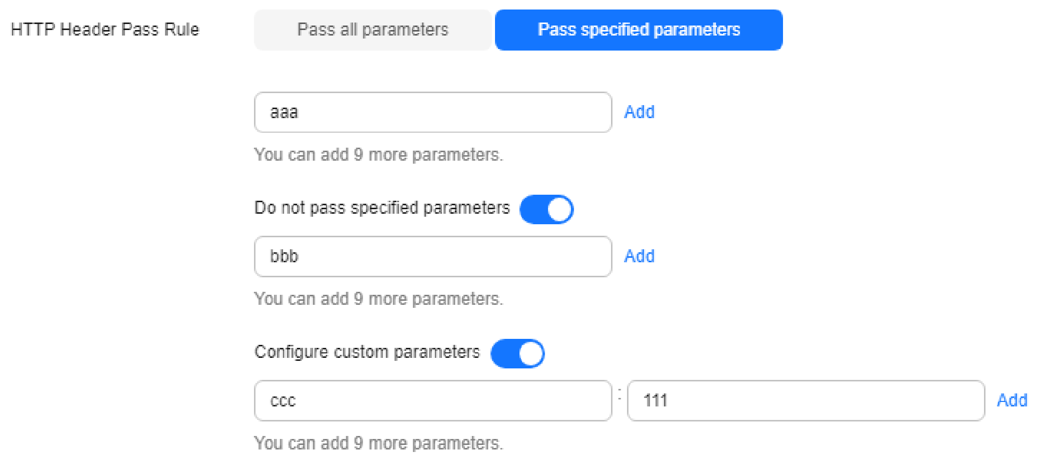
----End

References

Example for configuring an HTTP header pass rule:

Assume that the parameters are set as shown in **Figure 9-10**.

Figure 9-10 Configuring an HTTP header pass rule



HTTP Header Pass Rule: Pass all parameters | **Pass specified parameters**

aaa — Add

You can add 9 more parameters.

Do not pass specified parameters: on

bbb — Add

You can add 9 more parameters.

Configure custom parameters: on

ccc : 111 — Add

You can add 9 more parameters.

Based on the preceding configuration, if the header of the request sent to OBS is as follows:

```
GET /ObjectName HTTP/1.1
Host: bucketname.obs.region.myhuaweicloud.com
aaa:aaa
bbb:bbb
ccc:ccc
```

OBS sends the following request to the source site when the back-to-source rule is triggered:

```
GET /ObjectName HTTP/1.1
Host: source.com
aaa:aaa
ccc:111
```

Notes for passing HTTP headers during back to source

- HTTP headers that can be passed from a source site to a client:
  - Content-Type
  - Content-Language
  - Content-Encoding
  - Content-Disposition
  - Cache-Control
  - Expires
- HTTP headers that cannot be passed from a client to a source site:
  a. HTTP headers starting with the prefix below:
     x-obs-
  b. All standard HTTP headers, including:
     - Content-Length
     - Authorization2
     - Authorization

- Range
- Date

Creating an Agency for Back to Source

Step 1 In the **Create Back-to-Source Rule** dialog box on OBS Console, click **View Agencies** to jump to the **Agencies** page on the IAM console.

Step 2 Click **Create Agency**.

Step 3 Enter an agency name.

Step 4 Select **Cloud service** for the **Agency Type**.

Step 5 Select **Object Storage Service (OBS)** for **Cloud Service**.

Step 6 Set a validity period.

Step 7 Click **Next**.

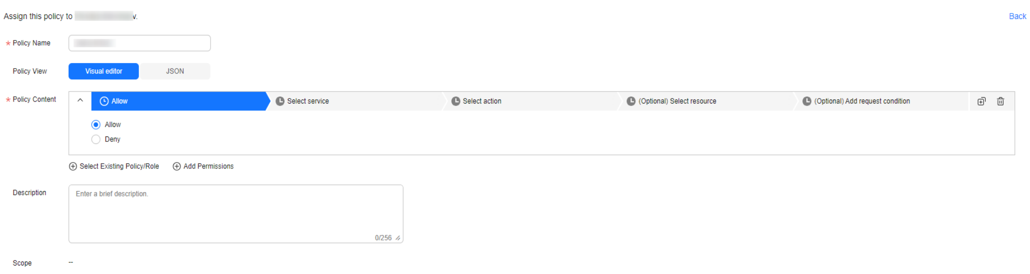
NOTE

The console for creating an agency has the new and old editions. The following steps use the new edition.

Step 8 Click **Done**. In the displayed dialog box, click **Authorize**.

Step 9 If there is already a policy meeting your requirements, go to **Step 11**. Otherwise, click **Create Policy** in the upper right corner. Then, specify a policy name and choose **Visual editor** for **Policy View**.

Figure 9-11 Authorizing an agency



Assign this policy to [redacted]v.

* Policy Name: [redacted]

Policy View: Visual editor | JSON

* Policy Content: Allow

● Allow
○ Deny

Description: Enter a brief description. 0/256

Scope: --

Back
Select service
Select action
(Optional) Select resource
(Optional) Add request condition
Select Existing Policy/Role
Add Permissions

Step 10 Configure **Policy Content** as follows:

1. Select **Allow**.
2. For the service, select **Object Storage Service (OBS)**.
3. For actions, under **ReadOnly**, select **obs:object:GetObject**, under **Read/ Write**, select **obs:object:PutObject** and **obs:object:AbortMultipartUpload**, and under **ListOnly**, select **obs:bucket:ListBucket**.
4. For **Resources**, select **All**.
5. Click **Next**.

Step 11 Select the policy and click **Next**.

Step 12 On the **Select Scope** page, select **All resources** for **Scope** and click **OK**.

NOTE

All resources is the recommended option for **Scope**. In this policy, the **All resources** option means that OBS can use all resources, including those in enterprise projects, region-specific projects, and global services under the account based on assigned permissions.

----End

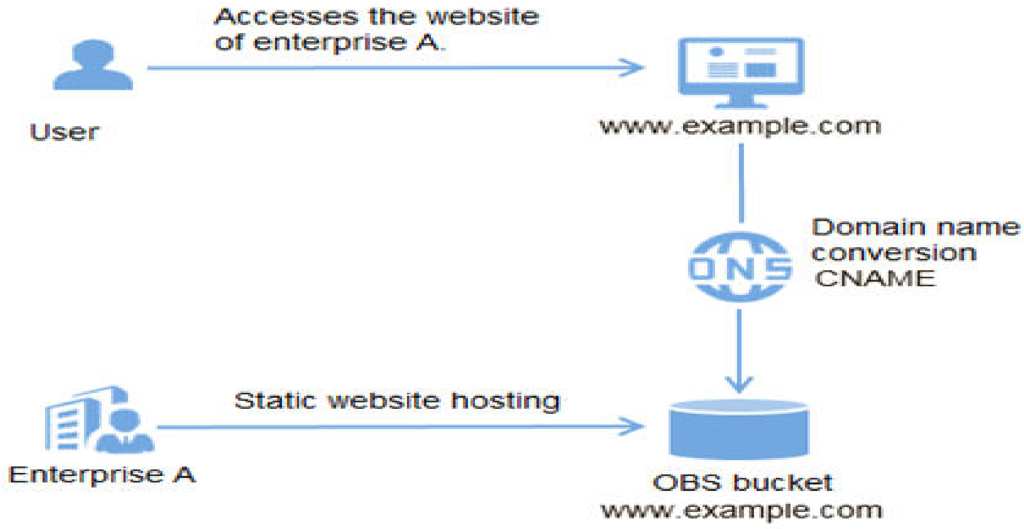
9.5 Static Website Hosting

Scenarios

OBS allows static websites to be hosted by buckets and supports index page, error page, and page redirection. To host a static website in OBS, upload the static website files to a bucket, grant the read permission for these files to anonymous users, and configure static website hosting for the bucket.

Static websites contain static web pages and some scripts that can run on clients, such as JavaScript and Flash. Different from static websites, dynamic websites rely on servers to process scripts, such as PHP, JSP, and ASP.NET. OBS does not support scripts running on servers.

Figure 9-12 Static website hosting



After static website hosting is configured, you can access OBS in either of the following ways:

- Resource management mode: You can access OBS using a common domain name. You can use APIs or SDKs and default OBS domain names with endpoints contained to perform common operations on buckets and objects, such as upload, download, deletion, or listing.
- Static website: You can access objects in a bucket by using a specific domain name. You can configure domain name hosting rules to enable redirection of an index page, error page, or access page.

The configuration of static website hosting takes about two minutes to take effect. After it takes effect, you can access static resources using the following URLs:

```
https://static-website-domain-name/object-name
http://static-website-domain-name/object-name
```

A static website domain name is in the *BucketName***.obs-website.***Endpoint* format. *BucketName* indicates the name of a bucket, and *Endpoint* indicates the OBS domain name of the region where the bucket is located. For details about regions and endpoints, see **Regions and Endpoints**. For example, if you want to access bucket **testbucket** in the CN-Hong Kong region, the corresponding endpoint is **ap-southeast-1.myhuaweicloud.com**, and the static website domain name is **testbucket.obs-website.ap-southeast-1.myhuaweicloud.com**.

NOTE

- Avoid using periods (.) in the target bucket name, or there may be errors when the client verifies the certificate during HTTPS access.
- In static website hosting scenarios, anonymous users must be granted access to hosted static website files. When they access the hosted files, there will be costs for outbound Internet traffic and requests.

To allow your users to access the content on your website, you must assign public read to your content. You can use bucket policies or ACLs on objects to grant permissions.

The following table lists the differences between the resource management mode and static website mode.

Main Difference	Resource Management	Static Website
Access control	Both public content and private content are supported.	Only publicly readable content is supported.
Error handling	An error response in XML format is returned.	An HTML document is returned.
Redirection support	N/A	Both object-level and bucket-level redirection are supported.
Supported requests	All operations on buckets and objects are supported.	Only GET and HEAD requests on objects are supported.
Response to GET and HEAD requests at the root level of a bucket	A list of object keys in a bucket is returned.	An index file specified in the configuration is returned.

Precautions

For security and compliance purposes, Huawei Cloud OBS does not allow you to use static website hosting based on the default domain name (**a bucket domain name or static website domain name**). When you use such a domain name to access web pages in a browser, no content will be displayed. Instead, the content is downloaded as an attachment.

This restriction takes effect in different regions at the following two points in time:

January 1, 2022: CN North-Beijing1, CN North-Beijing4, CN East-Shanghai1, CN East-Shanghai2, and CN South-Guangzhou

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

You can still use static website hosting with a user-defined domain name. This way, the content can still be previewed. For details, see **How Do I Preview OBS Objects in My Web Browser?**

Configuring Static Website Hosting

Description

If you want to use a bucket to host static websites, add the website configuration to the bucket. The configuration includes the following information:

Index document

When you enter a URL such as **http://example.com**, you are not requesting a specific page. In this case, the web server will provide a default page that contains the directory of the requested website. This default page is called an index document, and in most cases it is known as **index.html**. When you configure a bucket for website hosting, you must specify an index document. When a request is sent to the root domain or any subfolder, OBS returns this index document.

Error document

If an error occurs, OBS returns an HTML error document. For 4*XX* errors, you can provide your own custom error document that gives users guidance to address the error.

Redirection of all requests

If the root domain is **example.com** and you need to respond to requests for **http://example.com** and **http://www.example.com**, you can create two buckets named **example.com** and **www.example.com**, respectively. Then you store the website content in only one bucket (such as **example.com**), and configure the other bucket to redirect all requests to the **example.com** bucket.

Redirection based on advanced conditions

You can redirect requests based on the specific object name or prefix in the request, or based on the response code. For example, if you deleted or renamed an object in a bucket, you can add a routing rule to redirect requests for this object to another object.

The syntax format is as follows:

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

```
  <RoutingRules>
     ......RoutingRules......
  </RoutingRules>
</WebsiteConfiguration>
```

Here is the syntax format of **RoutingRules**. The content in square brackets ([]) is optional.

```
<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>
```

For website configuration for redirecting all requests and for setting routing rules, see the "Request Elements" area in **Configuring Static Website Hosting for a Bucket**

Examples:

Example 1: Modifying the object name prefix for redirection

Assume that your bucket contains the following objects:

index.html

docs/article1.html

docs/article2.html

You decide to change the object name from **docs/** to **documents/**. After the modification, the request for an object with prefix **/docs** needs to be redirected to **documents/**. For example, the request for **docs/article1.html** needs to be redirected to **documents/article1.html**.

To meet this need, you can add the following routing rule to the website configuration:

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

```
  </RoutingRule>
</RoutingRules>
```

Example 2: Redirecting requests sent to deleted folders to a specified page

Assume that you have deleted the **images/** folder, that is, you have deleted all objects whose object name prefix is **images/**. You can add a routing rule that redirects the requests of all objects whose prefix is **images/** to a page named **folderdeleted.html**.

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

Example 3: Redirection when HTTP errors occur

You want to redirect requests for an object that is not found to **www.example.com**. You can add a routing rule to redirect users to **www.example.com** if HTTP status code 404 (object not found) is returned. The following example inserts the object name prefix **report-404/** into **Redirect**. Assume you request the page **ExamplePage.html** and receive an HTTP 404 error. In this case, you will be redirected to the page **report-404/ExamplePage.html** on the **www.example.com**. If there is no routing rule and HTTP error 404 occurs, the error document specified in the configuration is returned.

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

Index Document Support

An index document is a web page returned when a request is sent to the root of a website or any subfolder. For example, if the user enters **http://www.example.com** in the browser, the user is not requesting any specific page. In this case, OBS provides an index document, also known as a default page.

When configuring website hosting for your bucket, you must provide the name of an index document, which must be uploaded to OBS and configured with public read.

The trailing slash (/) in the root URL is optional. Both the following URLs can return an index document.

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

In OBS, objects are stored in buckets horizontally, not in a hierarchical structure like that of a file system on a computer. You can create a logical hierarchy based on object names that represent a folder structure. For example, consider a bucket with the following three objects:

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

Although they are not stored in a hierarchical structure, you can infer the following logical relationship from their object name.

- The **sample1.jpg** object is located at the root level of the bucket.
- The **sample2.jpg** object is located in the **photos/2006/Jan** subfolder.
- The **sample3.jpg** object is located in the **photos/2006/Feb** subfolder.

Custom Error Document Support

Table 9-15 lists the HTTP response codes returned by OBS when an error occurs.

Table 9-15 List of OBS error codes

HTTP Error Code	Description
301 Moved Permanently	When a user sends a request to an endpoint of OBS, a **301 Moved Permanently** response is returned.
302 Found	When OBS receives a request for key *x* that does not end with a slash, it searches for the object whose key name is *x*. If the object is not found, OBS determines that the request is sent for subfolder **x**. Then OBS redirects the request by adding a slash to the end of **x** and returns **302 Found**.
304 Not Modified	OBS users request the **If-Modified-Since** and **If-Unmodified-Since** headers to determine whether the requested object is the same as the cached copy stored on the client. If the objects are the same, the website endpoint returns a **304 Not Modified** response.
403 Forbidden	If the requested object is changed to deny public read, a request for this object will receive a **403 Forbidden** response. An object owner must use bucket policies or ACLs to allow public read for the object.

HTTP Error Code	Description
404 Not Found	This response is returned if: ● OBS determines that an invalid object key is referenced in the website URL. ● The OBS determines that the request is for an index document that does not exist. ● The bucket specified in the URL does not exist. ● The bucket specified in the URL exists but is not configured as a website. You can create a custom document that will be returned if **404 Not Found** occurs. Ensure that the document has been uploaded to a bucket configured as a website and the document has been configured as an error document in the website hosting configuration.
500 Service Error	If there is an internal server error, a **500 Service Error** response is returned.
503 Service Unavailable	If your requests are too frequent, a **503 Service Unavailable** response is returned.

Permissions Required for Website Access

When you configure a bucket as a website, you must configure public read for the objects in the bucket. To perform this operation, you need to write a bucket policy that grants the GetObject permission to everyone. If an object requested by a user does not exist on the website, OBS returns the HTTP response code **404 Not Found**. If the object exists but you did not grant read permission for the object, OBS returns HTTP response code **403 Access Denied**. Requesters can use the response code to determine whether the requested object exists. If you do not need this function, do not enable the bucket website support.

NOTE

In static website hosting scenarios, anonymous users must be granted access to hosted static website files. When they access the hosted files, there will be costs for outbound Internet traffic and requests.

The following example bucket policy grants everyone the permission to access objects in a specified folder. For more information about bucket policies, see **Bucket Policy**.

```
{
  "Statement":[{
```

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

NOTICE

A bucket policy applies only to objects owned by a bucket owner. If the bucket contains objects that are not owned by the bucket owner, use object ACLs to grant the public read permission for the objects.

You can use bucket policies or object ACLs to grant the public read permission for your objects. To use ACLs to grant the read permission to everyone, refer to the example below. You can add the authorization element to object ACLs. For details about ACL management, see **ACLs**.

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

Related Functions

Function	Relationship with Static Website Hosting	Reference
Cross-origin resource sharing (CORS)	By default, static websites hosted in an OBS bucket can only respond to access requests from websites in the same domain. You can configure CORS for the bucket so static websites can be accessed from a different origin.	**CORS**
User-defined domain name configuration	OBS allows you to continue to use the original domain name of your static website hosted in OBS by configuring a user-defined domain name, with no code change required.	**Configuring User-defined Domain Names** **Using a User-Defined Domain Name to Host a Static Website**
URL validation	OBS uses URL validation to protect your website from inline linking. OBS verifies URLs based on the referer field in the HTTP header.	**URL Validation**

Prerequisites

Web page files required for static website hosting have been uploaded to the specified bucket.

The static website files hosted in the bucket are accessible to all users.

Static web page files in the Archiveor Deep Archive storage class have been restored. For more information, see **Restoring Archive or Deep Archive Objects**.

Ways to Configure Static Website Hosting

You can use OBS Console, APIs, or SDKs to configure static website hosting.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

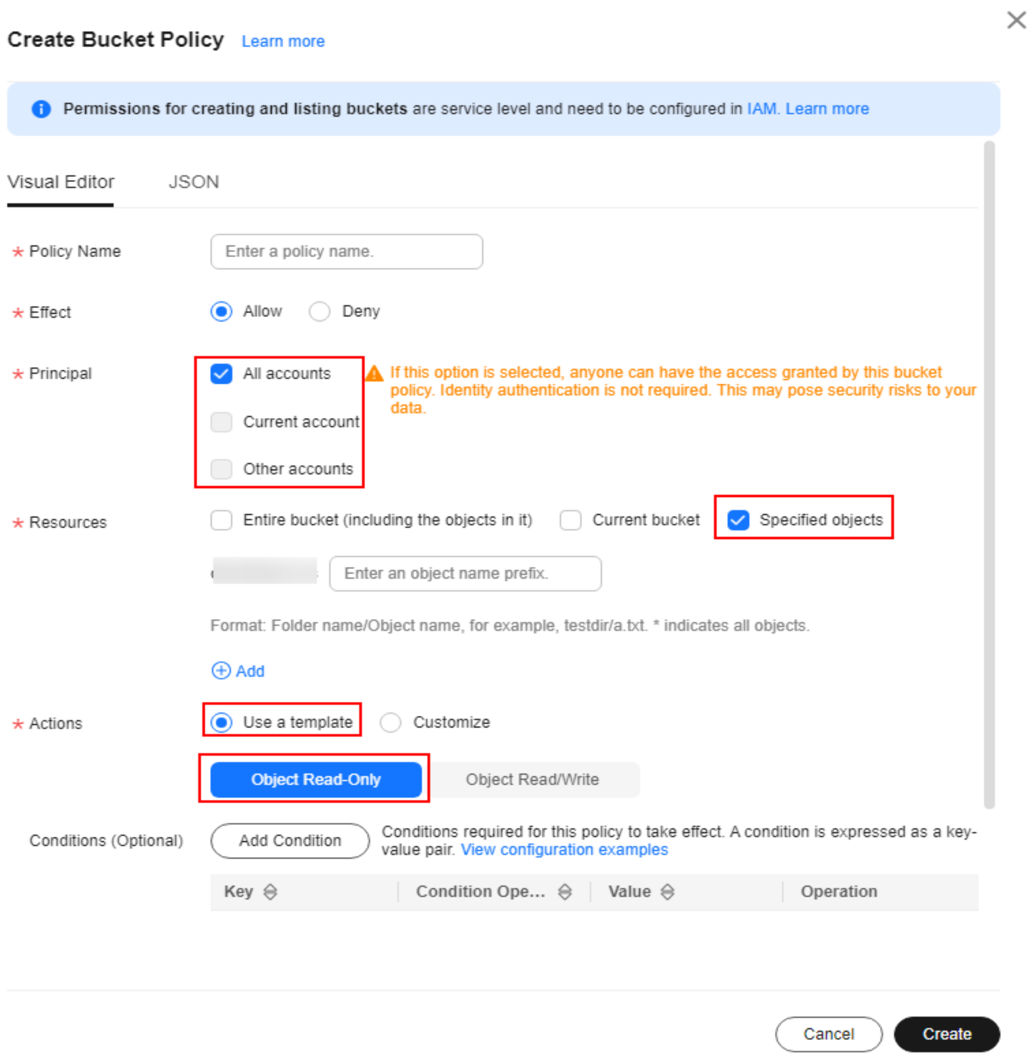
Step 3 (**Optional**) If the static website files in the bucket are not accessible to everyone, perform this step. If they are already accessible to everyone, skip this step.

To grant required permissions, see **Granting All Accounts Read Permission for Specified Objects**.

If the bucket contains only static website files, configure the **Object Read-Only** policy for the bucket, so that all files in it are publicly accessible.

1. Choose **Permissions** > **Bucket Policies**.
2. Click **Create**.
3. Configure bucket policy information.

Figure 9-13 Granting the Object Read-Only permission



Create Bucket Policy Learn more

Permissions for creating and listing buckets are service level and need to be configured in IAM. Learn more

Visual Editor JSON

* Policy Name Enter a policy name.

* Effect ● Allow ○ Deny

* Principal ☑ All accounts □ Current account □ Other accounts

If this option is selected, anyone can have the access granted by this bucket policy. Identity authentication is not required. This may pose security risks to your data.

* Resources □ Entire bucket (including the objects in it) □ Current bucket ☑ Specified objects

[redacted] Enter an object name prefix.

Format: Folder name/Object name, for example, testdir/a.txt. * indicates all objects.

Add

* Actions ● Use a template ○ Customize

Object Read-Only Object Read/Write

Conditions (Optional) Add Condition Conditions required for this policy to take effect. A condition is expressed as a key-value pair. View configuration examples

Key	Condition Ope...	Value	Operation

Cancel Create

Table 9-16 Parameters for configuring a public read policy

Parameter		Description
Configuration method		**Visual Editor** and **JSON** are available. The visual editor is used here. For details about configurations in the JSON view, see **Creating a Custom Bucket Policy (JSON View)**.
Policy Name		Enter a custom policy name.
Policy content	Effect	Select **Allow**.
	Principals	Select **All accounts**.
	Resources	– Select **Specified objects**. – Set the resource path to * (indicating all objects in the bucket).

Parameter		Description
	Actions	– Choose **Use a template**. – Select **Object Read-Only**.

4. Click **Create**. The bucket policy is created.

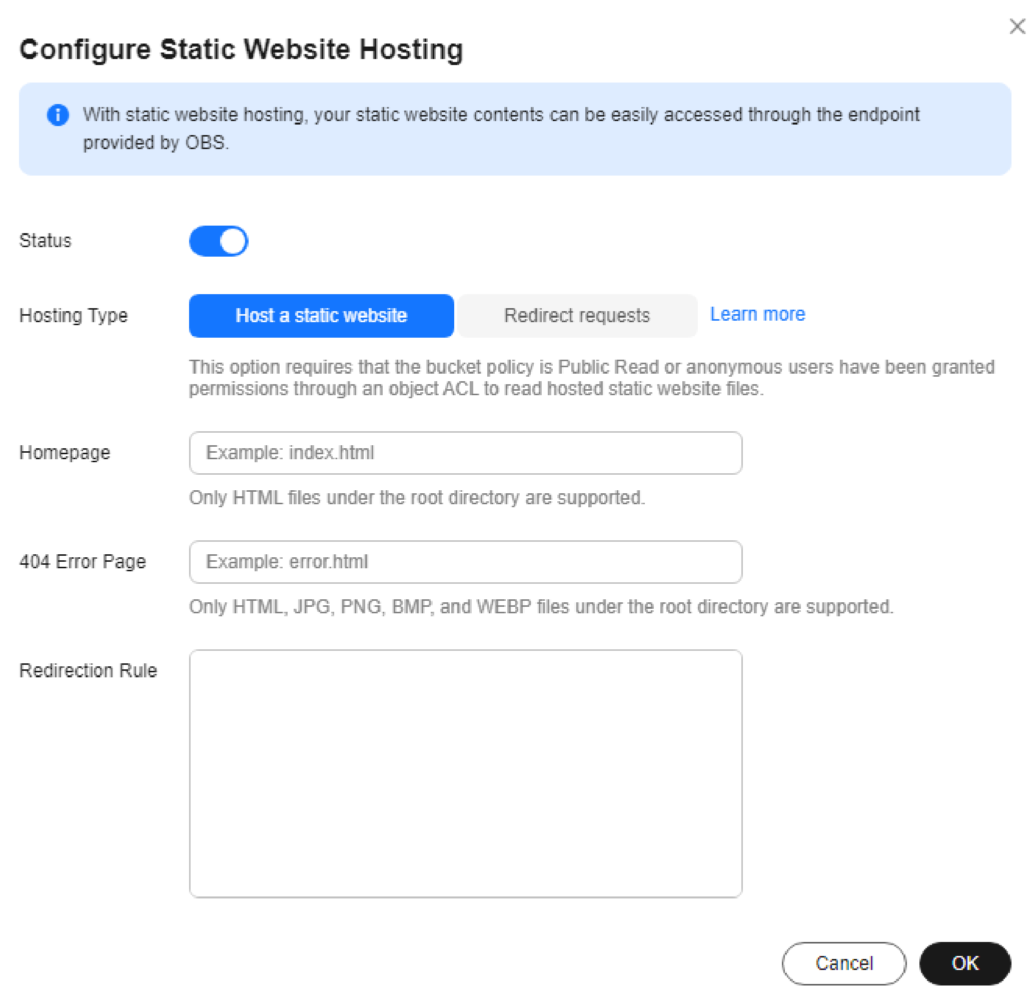
Step 4 In the navigation pane, choose **Basic Configurations** > **Static Website Hosting**. The **Static Website Hosting** page is displayed.

Step 5 Click **Configure Static Website Hosting**. The **Configure Static Website Hosting** dialog box is displayed.

Step 6 Enable **Status**.

Step 7 Set the hosting type to the current bucket. For details, see **Figure 9-14**.

Figure 9-14 Configuring static website hosting



Configure Static Website Hosting

With static website hosting, your static website contents can be easily accessed through the endpoint provided by OBS.

Status

Hosting Type: Host a static website

This option requires that the bucket policy is Public Read or anonymous users have been granted permissions through an object ACL to read hosted static website files.

Homepage: Example: index.html

Only HTML files under the root directory are supported.

404 Error Page: Example: error.html

Only HTML, JPG, PNG, BMP, and WEBP files under the root directory are supported.

Redirection Rule

Step 8 Configure the homepage and 404 error page.

- **Homepage**: specifies the default homepage of the static website. When OBS Console is used to configure static website hosting, only HTML web pages are supported. When APIs are used to configure static website hosting, OBS does not have any restriction but the **Content-Type** of objects must be specified.

OBS only allows files such as **index.html** in the root directory of a bucket to function as the default homepage. Do not set the default homepage with a multi-level directory structure (for example, **/page/index.html**).

- **404 Error Page**: specifies the error page returned when an error occurs during static website access. When OBS Console is used to configure static website hosting, only HTML, JPG, PNG, BMP, and WebP files under the root directory are supported. When APIs are used to configure static website hosting, OBS does not have any restriction but the **Content-Type** of objects must be specified.

Step 9 **Optional**: In **Redirection Rules**, configure redirection rules. Requests that comply with the redirection rules are redirected to the specific host or page.

A redirection rule is in the JSON format. Each rule contains a **Condition** and a **Redirect**. Multiple rules can be configured. The parameters are described in **Table 9-17**.

Table 9-17 Parameter description

Container	Key	Description
Condition	KeyPrefixEquals	Object name prefix on which the redirection rule takes effect. When a request is sent for accessing an object, the redirection rule takes effect if the object name prefix matches the value specified for this parameter. For example, to redirect the request for object **ExamplePage.html**, set the **KeyPrefixEquals** to **ExamplePage.html**.
	HttpErrorCodeReturnedEquals	HTTP error codes upon which the redirection rule takes effect. The specified redirection is applied only when the error code returned equals the value specified for this parameter. For example, if you want to redirect requests to **NotFound.html** when HTTP error code 404 is returned, set **HttpErrorCodeReturnedEquals** to **404** in **Condition**, and set **ReplaceKeyWith** to **NotFound.html** in **Redirect**.
Redirect	Protocol	Protocol used for redirecting requests. The value can be **http** or **https**. If this parameter is not specified, the default value **http** is used.

Container	Key	Description
	HostName	Host name to which the redirection is pointed. If this parameter is not specified, the request is redirected to the host from which the original request is initiated.
	ReplaceKeyPrefix-With	The object name prefix used in the redirection request. OBS replaces the value of **KeyPrefixEquals** with the value you specified here for **ReplaceKeyPrefixWith**. To redirect all requests for objects under docs to objects under documents, set **KeyPrefixEquals** to **docs** in **Condition** and set **ReplaceKeyPrefixWith** to **documents** in **Redirect**. This way, requests for object **docs/a.html** will be redirected to **documents/a.html**.
	ReplaceKeyWith	The object name used in the redirection request. OBS replaces the entire object name in the request with the value you specified here for **ReplaceKeyWith**. For example, to redirect requests for all objects in the **docs** directory to **documents/error.html**, set **KeyPrefixEquals** to **docs** under **Condition** and **ReplaceKeyWith** to **documents/error.html** under **Redirect**. This way, requests for both objects **docs/a.html** and **docs/b.html** will be redirected to **documents/error.html**.
	HttpRedirectCode	HTTP status code returned to the redirection request. The default value is **301**, indicating that requests are permanently redirected to the location specified by **Redirect**. You can also set this parameter based on your service needs.

Example of setting a redirection rule

- Example 1: All requests for objects prefixed with **folder1/** are automatically redirected to pages prefixed with **target.html** on host **www.example.com** using HTTPS.

```
[
   {
   "Condition": {
       "KeyPrefixEquals": "folder1/"
       },
   "Redirect":{
       "Protocol": "https",
       "HostName": "www.example.com",
       "ReplaceKeyPrefixWith": "target.html"
```

```
    }
  }
]
```

- Example 2: All requests for objects prefixed with **folder2/** are automatically redirected to objects prefixed with **folder/** in the same bucket.

```
[
  {
  "Condition": {
     "KeyPrefixEquals": "folder2/"
     },
  "Redirect":{
     "ReplaceKeyPrefixWith": "folder/"
     }
  }
]
```

- Example 3: All requests for objects prefixed with **folder.html** are automatically redirected to the **folderdeleted.html** object in the same bucket.

```
[
  {
  "Condition": {
     "KeyPrefixEquals": "folder.html"
     },
  "Redirect":{
     "ReplaceKeyWith": "folderdeleted.html"
     }
  }
]
```

- Example 4: If the HTTP status code 404 is returned, the request is automatically redirected to the page prefixed with **report-404/** on host **www.example.com**.

  For example, if you request the page **ExamplePage.html** but the HTTP 404 error is returned, the request will be redirected to the **report-404/ExamplePage.html** page on the **www.example.com**. If the 404 redirection rule is not specified, the default 404 error page configured in the previous step is returned when the HTTP 404 error occurs.

```
[
  {
  "Condition": {
     "HttpErrorCodeReturnedEquals": "404"
     },
  "Redirect":{
     "HostName": "www.example.com",
     "ReplaceKeyPrefixWith": "report-404/"
     }
  }
]
```

Step 10 Click **OK**.

When the static website hosting is effective, you can access the static website by using the URL provided by OBS.

NOTE

In some conditions, you may need to clear the browser cache before the expected results are displayed.

----End

Using the API

Configuring Static Website Hosting for a Bucket

Using SDKs

Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

9.6 Redirecting Requests

Redirection Overview

When using static website hosting, you can configure redirection to redirect specific or all requests.

If the structure, address, or file name extension of a website is changed, users will fail to access the website using the old address (such as the address saved in the folder of favorites), and the 404 error message is returned. In this case, you can configure redirection for the website to redirect user access requests to a specified page rather than a 404 error page.

Typical scenarios include:

- Redirecting all requests to another website.
- Redirecting specific requests based on redirection rules.

Prerequisites

Web page files required for static website hosting have been uploaded to the specified bucket.

The static website files hosted in the bucket are accessible to all users.

Static web page files in the Archiveor Deep Archive storage class have been restored. For more information, see **Restoring Archive or Deep Archive Objects**.

Procedure

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Overview**.

Step 4 In the **Basic Configurations** area, click **Static Website Hosting**. The **Static Website Hosting** page is displayed.

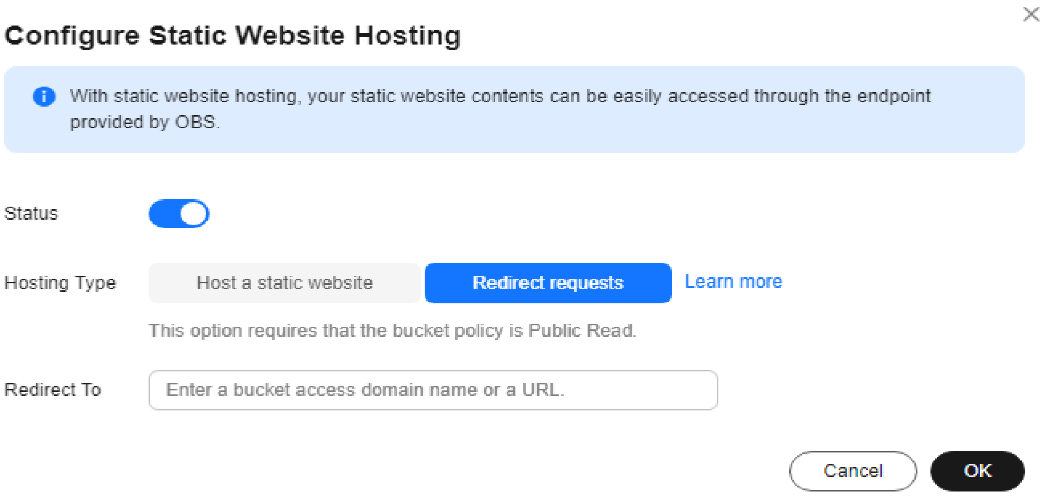
Alternatively, you can choose **Basic Configurations** > **Static Website Hosting** from the navigation pane on the left.

Step 5 Click **Configure Static Website Hosting**. The **Configure Static Website Hosting** dialog box is displayed.

Step 6 Enable **Status**.

Step 7 Set **Hosting Type** to **Redirect requests**, as shown in **Figure 9-15**. In the text box of **Redirect To**, enter the bucket's access domain name or URL.

Figure 9-15 Configuring redirection



Configure Static Website Hosting

With static website hosting, your static website contents can be easily accessed through the endpoint provided by OBS.

Status: On

Hosting Type: Host a static website | Redirect requests (selected)

This option requires that the bucket policy is Public Read.

Redirect To: Enter a bucket access domain name or a URL.

Step 8 Click **OK**.

Step 9 In the bucket list, click the bucket to which requests for the static website are redirected.

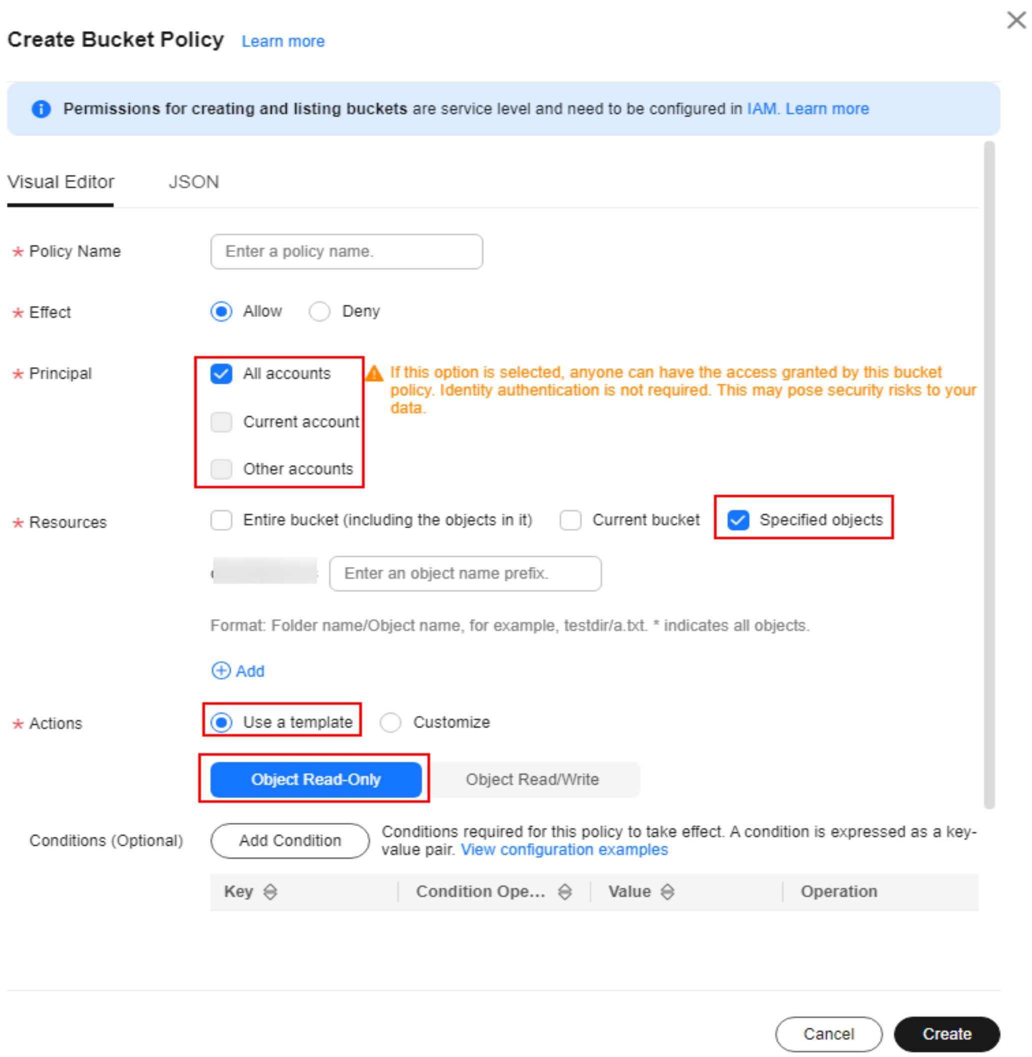
Step 10 (**Optional**) If the static website files in the bucket are not accessible to everyone, perform this step. If they are already accessible to everyone, skip this step.

To grant required permissions, see **Granting All Accounts Read Permission for Specified Objects**.

If the bucket contains only static website files, configure the **Object Read-Only** policy for the bucket, so that all files in it are publicly accessible.

1. Choose **Permissions** > **Bucket Policies**.
2. Click **Create**.
3. Configure bucket policy information.

Figure 9-16 Granting the Object Read-Only permission



Create Bucket Policy Learn more

Permissions for creating and listing buckets are service level and need to be configured in IAM. Learn more

Visual Editor JSON

* Policy Name: Enter a policy name.

* Effect: ● Allow ○ Deny

* Principal: ☑ All accounts ☐ Current account ☐ Other accounts

If this option is selected, anyone can have the access granted by this bucket policy. Identity authentication is not required. This may pose security risks to your data.

* Resources: ☐ Entire bucket (including the objects in it) ☐ Current bucket ☑ Specified objects

Enter an object name prefix.

Format: Folder name/Object name, for example, testdir/a.txt. * indicates all objects.

⊕ Add

* Actions: ● Use a template ○ Customize

Object Read-Only | Object Read/Write

Conditions (Optional): Add Condition

Conditions required for this policy to take effect. A condition is expressed as a key-value pair. View configuration examples

Key	Condition Ope...	Value	Operation

Cancel | Create

Table 9-18 Parameters for configuring a public read policy

Parameter		Description
Configuration method		**Visual Editor** and **JSON** are available. The visual editor is used here. For details about configurations in the JSON view, see **Creating a Custom Bucket Policy (JSON View)**.
Policy Name		Enter a custom policy name.
Policy content	Effect	Select **Allow**.
	Principals	Select **All accounts**.
	Resources	– Select **Specified objects**. – Set the resource path to * (indicating all objects in the bucket).

Parameter		Description
	Actions	– Choose **Use a template**. – Select **Object Read-Only**.

4. Click **Create**. The bucket policy is created.

Step 11 **Verification**: Input the access domain name of the bucket in the web browser and press **Enter**. The bucket or URL to which requests are redirected will be displayed.

NOTE

In some conditions, you may need to clear the browser cache before the expected results are displayed.

----End

10 Data Processing

10.1 Image Processing

10.1.1 Overview

Introduction

Image processing is a feature integrated in Object Storage Service (OBS). It provides stable, secure, efficient, and inexpensive image processing services. By using this feature, you can slim, crop, resize, and watermark images, as well as convert the formats of images.

You can access this feature via OBS Console and REST APIs, to process images stored in OBS anytime and anywhere and obtain the processed images right away.

Application Scenarios

Image processing enables you to resize, crop, or compress images on cloud. You do not have to download space-consuming software to your local computers.

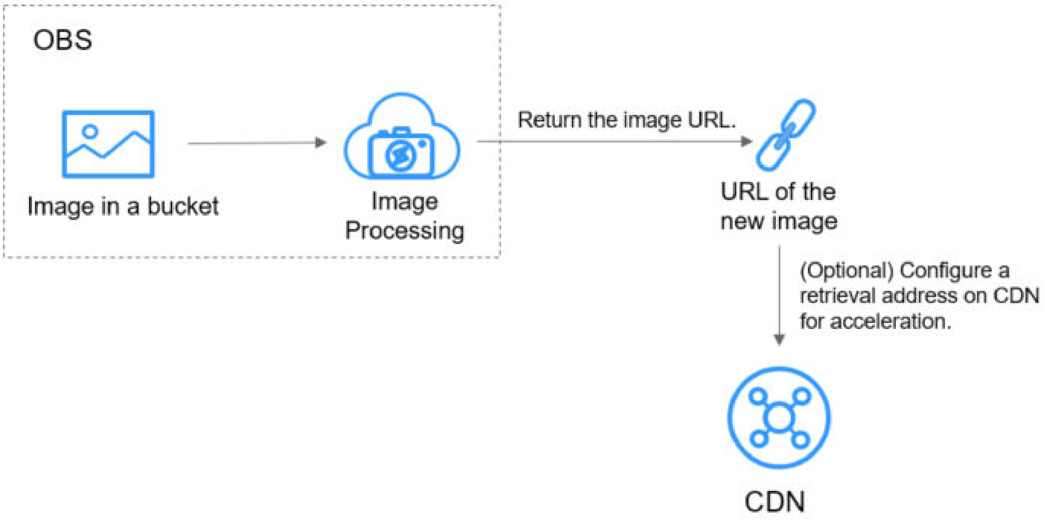
For example, you can add effects to or resize images in your cloud album anytime and anywhere, and quickly share the images with your friends online.

Shopping or other likewise websites where images are frequently accessed can use image processing to simply process the images and delivers them to Content Delivery Network (CDN) for acceleration. The processed images can then be faster downloaded from CDN.

Architecture

You can upload your images to OBS using OBS Console, OBS Browser+, REST APIs, or third-party clients. Before downloading and using an image, you can create an image style or pass image processing parameters to process it, such as cropping and compressing the image. After processing, you can obtain the URL of the new image. Furthermore, you can deliver the new image to CDN for acceleration and directly download it from CDN when needed. **Figure 10-1** shows the flowchart of image processing.

Figure 10-1 Image processing flowchart



NOTE

After processing, new images will be directly displayed in a browser and are not stored in OBS, so they do not incur any storage costs.

Billing

Image processing is currently free of charge.

Ways to Use Image Processing

You can access image processing in either of the following ways:

- Using OBS Console

  On the image processing page, you can create a processing style for an image by specifying parameters on the GUI or by editing code. You can then view the new image effect on the **Preview Image** tab page on the image's details page. You can also click **Copy Link** to get the link for accessing the new image.

  To get started with image processing on OBS Console, see **Using Image Processing on OBS Console**.

- Using the REST API

  You can call OBS Representational State Transfer (REST) APIs to access image processing. In the REST architectural style, resources on a network are identified by Uniform Resource Identifiers (URIs) and applications on clients locate resources using Uniform Resource Locators (URLs). The URL is in the **https://***Endpoint***/***uri* format. By putting a URL that complies with the command rules of image processing in a browser, you can get the new image after processing. For more information, see the **Object Storage Service API Reference**.

  To get started with image processing with API, see **Using Image Processing with APIs**.

Image Processing Functions

Table 10-1 describes what you can do on images stored in OBS by using image processing.

Table 10-1 Image processing functions

Function	Description	How to Use
Getting Information About an Image	Returns an image's basic information, including the format, size, and average color value.	Making an API call
Setting Image Effects	Adjusts an image's effects, including the brightness, contrast, sharpness, and blur.	Using GUI on OBS Console Editing code on OBS Console Making an API call
Resizing Images	Resizes images based on a specified width, height, or percentage.	Editing code on OBS Console Making an API call
Watermarking Images	Adds a text watermark, an image watermark, or both of them to a specific position of an image. Text watermarks can have a color, font, and size configured, and image watermarks can be resized, rotated, or cropped before being added to images.	Using GUI on OBS Console Editing code on OBS Console Making an API call
Converting Formats and Interlacing Images	Converts images to various formats, and supports interlaced rendered images after conversion.	Using GUI on OBS Console Editing code on OBS Console Making an API call
Rotating Images	Rotates images clockwise, and automatically rotates images based on the recorded rotation information of cameras or smartphones.	Editing code on OBS Console Making an API call
Cropping Images	Crops images based on the specified width, height, radius of the inscribed circle, index, and radius of rounded corners.	Editing code on OBS Console Making an API call

Function	Description	How to Use
Changing Image Quality	Compresses JPG images based on the relative quality and absolute quality.	Editing code on OBS Console Making an API call
Slimming Images	Reduces the image size without compromising its quality.	Editing code on OBS Console Making an API call
Image Processing Persistency	Synchronously stores images after processing in a specified OBS bucket for you to directly access the processed images.	Editing code on OBS Console Making an API call
Access with Commands	Orchestrates multiple process commands in sequence. With this function, you can add multiple process commands to the URL of the image that you want to process, and separate each command using the designated delimiter. Then the commands are executed one by one from left to right.	Editing code on OBS Console Making an API call
Procedure 3: Creating an Image Style	Customizes image styles for you to batch process images that require the same processing operations.	Using GUI on OBS Console Editing code on OBS Console

10.1.2 Notes and Constraints

Operations

- All image processing operations will not change original images.
- Archive storage does not support image processing.
- Deep Archive storage does not support image processing.
- OBS buckets that use SSE-KMS do not support image processing.
- Only version 3.0 buckets support image processing. To check a bucket's version, go to the **Basic Information** area on the bucket's **Overview** page on OBS Console.
- Images must be accessible to anonymous users, so that they can be accessed in a web browser by image processing requests that have no signature included. A request URL example is given as follows:

  https://bucketname.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=style/stylename

  For details about how to make images accessible to anonymous users, see **Granting Anonymous Users the Public Read Permission for a Bucket**.

- For security and compliance purposes, using a default bucket domain name (**bucket domain name or static website domain name**) to preview objects in the bucket will be prohibited by OBS. When you use such domain name to access objects (such as videos, images, and web pages) through a browser, no content will be displayed. Instead, the content is downloaded as an attachment.

  For details, see **How Do I Preview OBS Objects in a Browser Online?**

Images

- Supported original image formats include JPG, JPEG, PNG, BMP, WebP, GIF, and TIFF.
- Supported formats after image processing include JPG, PNG, BMP, and WebP.
- An original image allowed for processing cannot exceed 25 MB in size, and its width and height after processing cannot exceed 4,096 pixels and 5,000 pixels, respectively.
- An animated image (like a GIF or WebP image) will be returned without processing if it is greater than 2 MB in size or has over 50 frames.
- Currently, processing images in CMYK may change their color.
- An input image can contain 250 megapixels at most.

Commands

A maximum of 20 commands are allowed, with a total length of no more than 512 characters.

Regions

Image processing is currently only available in some regions. For details about the supported regions, see image processing on the **Function Overview** page.

10.1.3 Basic Concepts

Styles

A style is an aggregation of parameters or image processing operations. When performing the same operations on multiple images, you can create an image style as a template to avoid repetitive operations. A bucket can have a maximum of 100 styles.

EXIF Data

Exchangeable Image File (EXIF) data is contained in images taken by some cameras or smartphones. EXIF data is a group of parameters inserted into JPEG or TIFF images. It encompasses the shooting information, such as the camera model, date and time when an image was taken, shooting mode, thumbnails, and other characteristics.

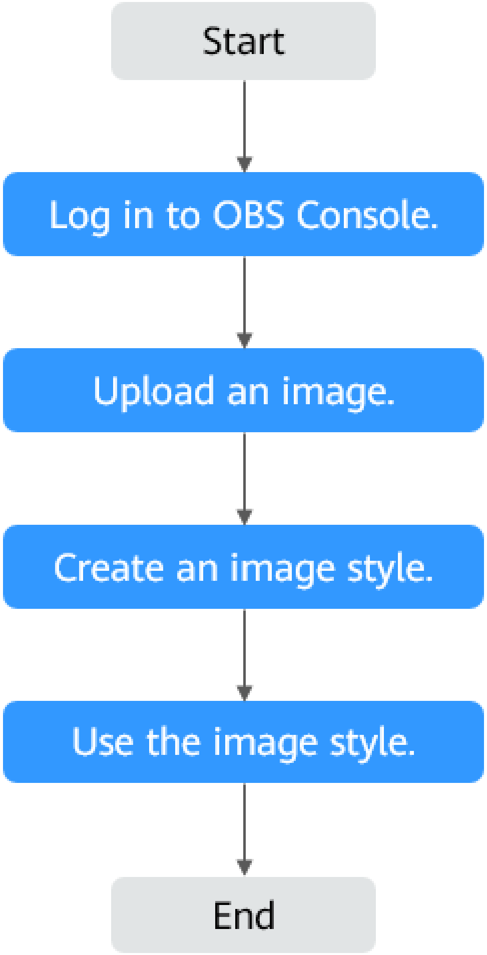
10.1.4 Using Image Processing on OBS Console

Process

On the image processing page of OBS Console, you can create a processing style for an image by specifying parameters on the GUI or by editing code. You can then view the new image effect on the **Preview Image** tab page on the image's details page. You can also click **Copy Link** to get the link for accessing the new image.

You can use OBS Console, OBS SDKs, or REST APIs to process images. **Figure 10-2** shows the flowchart of using image processing on OBS Console.

Figure 10-2 Flowchart of using image processing on OBS Console



Procedure 1: Logging In to OBS Console

Before using image processing, you must have a cloud service account and enable OBS. Once OBS is enabled, image processing is available for use. To use OBS properly, make sure that you have sufficient account balance or you have a valid payment method. If you have already enabled OBS and you can use it properly, skip this section.

Step 1 Visit the **Huawei Cloud official website**.

Step 2 Create a HUAWEI ID.

If you already have one, start from **Step 3**.

1. On the right of the top navigation bar, click **Sign Up**.
2. Complete the account creation as instructed.

   After the creation is complete, you will be navigated to your information page.

Step 3 On the right of the top navigation bar, click **Sign In**, and enter the username and password.

Step 4 On the right of the top navigation menu, click **Console** to go to the management console.

Step 5 Click **Service List** and choose **Storage** > **Object Storage Service**.

Step 6 (Recommended) Make sure that you have sufficient account balance or you have a valid payment method, so that you can properly use OBS.

----End

Procedure 2: Uploading an Image

NOTE

For the constraints on the format and size of images you can upload, see **Notes and Constraints**.

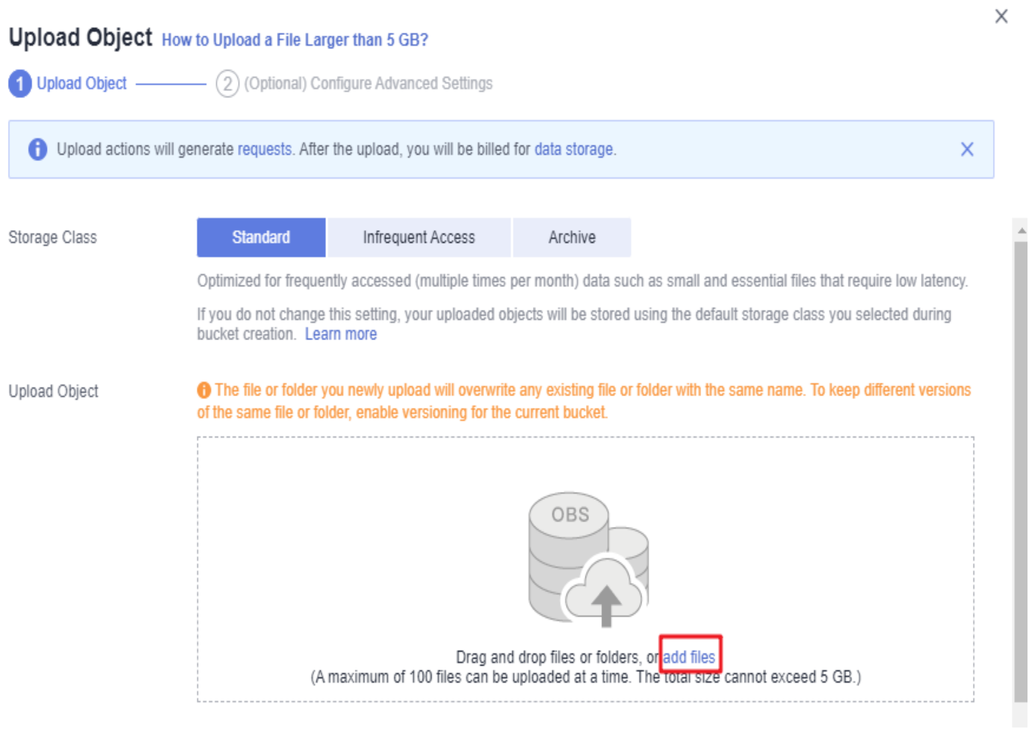
Step 1 On the console homepage, click **Service List** in the upper left corner and choose **Storage** > **Object Storage Service**.

Step 2 In the bucket list, click the bucket you want to go to the **Objects** page.

Step 3 Click **Upload Object** to open the **Upload Object** dialog box.

Step 4 Click **add files** marked by red box in **Figure 10-3** to open the local file browser.

Figure 10-3 Uploading an image



Upload Object How to Upload a File Larger than 5 GB?

1 Upload Object ——— 2 (Optional) Configure Advanced Settings

Upload actions will generate requests. After the upload, you will be billed for data storage.

Storage Class: Standard | Infrequent Access | Archive

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

If you do not change this setting, your uploaded objects will be stored using the default storage class you selected during bucket creation. Learn more

Upload Object: The file or folder you newly upload will overwrite any existing file or folder with the same name. To keep different versions of the same file or folder, enable versioning for the current bucket.

Drag and drop files or folders, or add files
(A maximum of 100 files can be uploaded at a time. The total size cannot exceed 5 GB.)

Step 5 Select the image that you want to upload and click **Open**.

Step 6 Click **Upload**.

----End

Procedure 3: Creating an Image Style

Each image style specifies a set of processing operations, including cropping, compression, and adding watermarks. For multiple images that require the same processing operations in a bucket, you can create an image style to apply it to the images.

When creating a style, you can preview the style effect of the sample image on the right in real time.

When using REST APIs to access image processing, you can call the style name in the URL to avoid entering complex commands. For details about the domain name rules for API access, see **Access with a Style Name**.

You can create up to 100 styles for a bucket.

You can also replicate existing image processing styles from another bucket.

NOTE

If you are an IAM user, before creating an image style, make sure that programmatic access has been enabled for yourself. For details, see **Viewing or Modifying IAM User Information**.

Step 1 On the console homepage, click **Service List** in the upper left corner and choose **Storage** > **Object Storage Service**.

Step 2 Click a desired bucket name. In the navigation pane, choose **Data Processing** > **Image Processing**.

Figure 10-4 Image processing

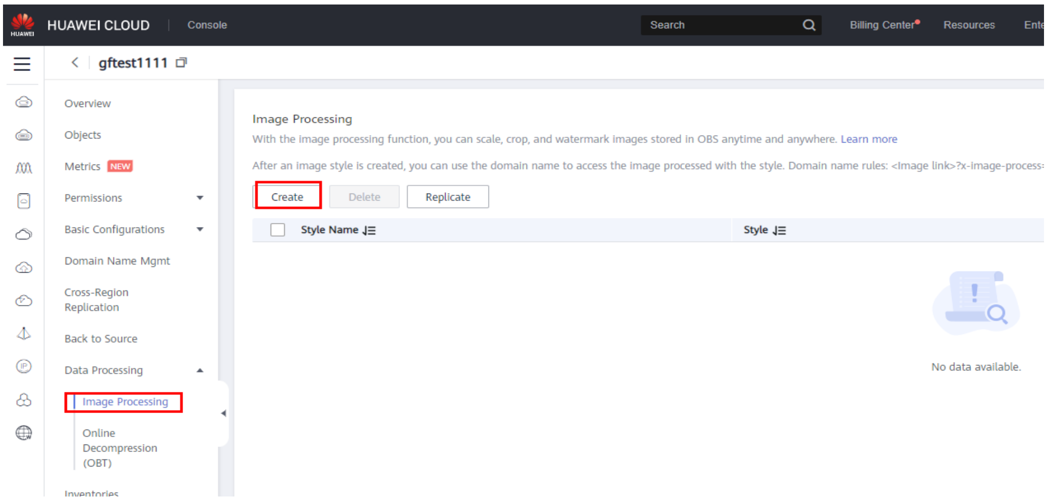


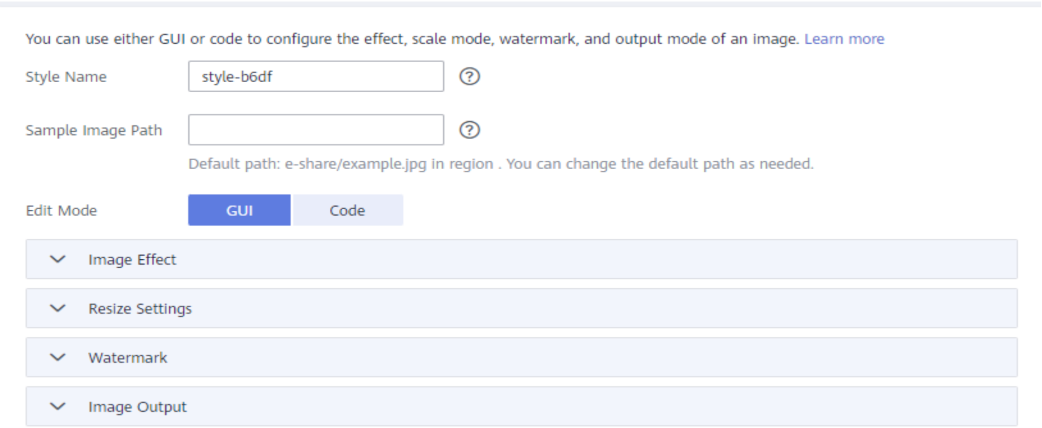
Image Processing

With the image processing function, you can scale, crop, and watermark images stored in OBS anytime and anywhere. Learn more

After an image style is created, you can use the domain name to access the image processed with the style. Domain name rules: <Image link>?x-image-process:

Step 3 Click **Create** to go to the style editing page shown in **Figure 10-5**.

Figure 10-5 Creating a style



You can use either GUI or code to configure the effect, scale mode, watermark, and output mode of an image. Learn more

Style Name: style-b6df

Sample Image Path:

Default path: e-share/example.jpg in region . You can change the default path as needed.

Edit Mode: GUI

Step 4 On the editing page, specify a style name and configure properties:

- **Style Name**

  Enter an easy-to-remember style name. Only letters (uppercase or lowercase), digits, periods (.), underscores (_), and hyphens (-) are allowed. The style name contains 1 to 256 characters, for example, **rotate_0001**.

- **Edit Mode**

  Choose either **GUI** or **Code** to create an image style by configuring parameters or by editing code.

  An example code is as follows:

```
image/sharpen,100/blur,r_1,s_1/resize,m_lfit,h_400,w_400,limit_1
```

- Other settings

  Configure the image effect, resizing, watermark, and output.

Step 5 Click **OK** to save the style. The new style will be displayed in the style list.

----End

Replicating Image Styles

You can replicate an existing image style to quickly create one.

Step 1 On the console homepage, click **Service List** in the upper left corner and choose **Storage** > **Object Storage Service**.

Step 2 Click a desired bucket name. In the navigation pane, choose **Data Processing** > **Image Processing**.

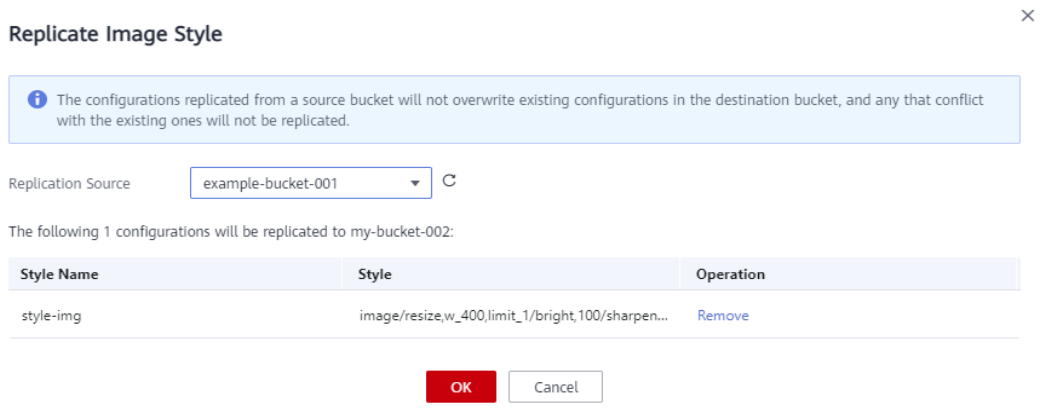
Step 3 Click **Replicate**.

Step 4 Select a replication source, which is the source bucket where the required image styles were configured.

NOTE

- The image styles replicated from a source bucket will not overwrite existing styles in the current bucket, and any that conflict with the existing ones will not be replicated.
- The version of both source and destination buckets must be 3.0.
- You can remove the styles that you do not want to replicate.

Figure 10-6 Replicating image styles



Replicate Image Style

The configurations replicated from a source bucket will not overwrite existing configurations in the destination bucket, and any that conflict with the existing ones will not be replicated.

Replication Source: example-bucket-001

The following 1 configurations will be replicated to my-bucket-002:

Style Name	Style	Operation
style-img	image/resize,w_400,limit_1/bright,100/sharpen...	Remove

Step 5 Click **OK** to replicate the image styles to the current bucket.

----End

Procedure 4: Using the Image Style

You can use existing image styles in a bucket in either of the following ways:

- **Copying a link** (see **Copying a Link**): On an image's details page of OBS Console, you can copy a link for accessing the image with a style applied on the **Preview Image** tab page. By pasting the copied link in a browser, you can get the new image.

- **Editing a domain name**: You can edit a domain name by referring to the format given below and open it in a browser to have the image with the style applied.

  ```
  <Image link>?x-image-process=style/<Style name>
  ```

  Image link can be obtained from the image details page. For details, see **Editing a Domain Name**. *Style name* is the one defined when the style was created. A style name can contain only uppercase or lowercase letters, digits, periods (.), underscores (_), and hyphens (-) and is 1 to 256 characters long, for example, **rotate_0001**.

Copying a Link

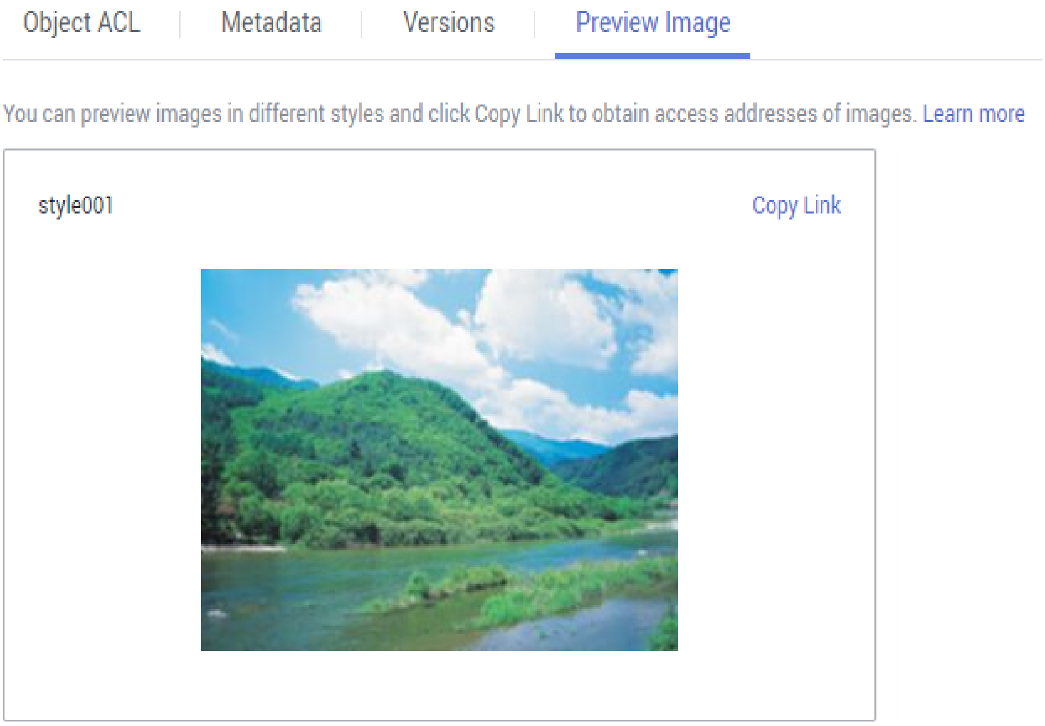
Step 1 On the console homepage, click **Service List** in the upper left corner and choose **Storage** > **Object Storage Service**.

Step 2 Click the name of the bucket that houses desired styles to go to the **Objects** page.

Step 3 Click the name of an existing image or of a newly uploaded image to go to its details page.

Step 4 Click the **Preview Image** tab to preview the image effect in one or more styles.

Figure 10-7 Preview Image



You can preview images in different styles and click Copy Link to obtain access addresses of images. Learn more

Step 5 Click **Copy Link** in the style card you want. If **Copied successfully.** is prompted, you have got the link for accessing the image with the style applied.

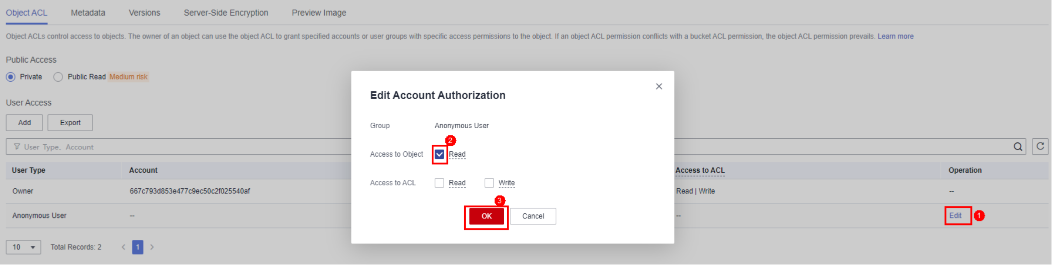
----End

Editing a Domain Name

Step 1 On the OBS object list page, click an image to be processed to go to its details page.

Step 2 Choose **Object ACL** > **User Access** > **Anonymous User** and click **Edit**. In the displayed dialog box, grant the object read permission to anonymous users and click **OK**. For details, see **Figure 10-8**.

Figure 10-8 Granting the object read permission to anonymous users



Object ACLs control access to objects. The owner of an object can use the object ACL to grant specified accounts or user groups with specific access permissions to the object. If an object ACL permission conflicts with a bucket ACL permission, the object ACL permission prevails. Learn more

Public Access

● Private ○ Public Read Medium risk

User Access

User Type	Account	Access to ACL	Operation
Owner	667c793d853e477c9ec50c2f025540af	Read | Write	--
Anonymous User	--	--	Edit

Total Records: 2

Edit Account Authorization

Group Anonymous User

Access to Object ☑ Read

Access to ACL □ Read □ Write

Step 3 Click the icon ⧉ next to **Link** to copy the image link.

Step 4 Add **?x-image-process=style/<*Style name*>** to the end of the copied link and open the new link in a browser. Then, you can access the image with the style applied.

Link example:

https://bucketname.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=style/stylename

NOTE

The links and style names above are for reference only. Change them accordingly in practice.

----End

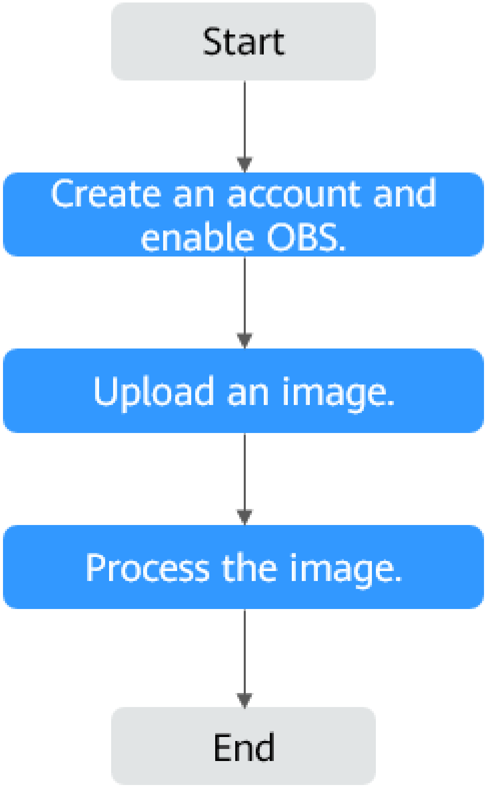
10.1.5 Using Image Processing with APIs

Process

You can call OBS Representational State Transfer (REST) APIs to access image processing. In the REST architectural style, resources on a network are identified by Uniform Resource Identifiers (URIs) and applications on clients locate resources using Uniform Resource Locators (URLs). The URL is in the **https://***Endpoint***/***uri* format. By putting a URL that complies with the command rules of image processing in a browser, you can get the new image after processing.

Figure 10-9 shows the flowchart of using image processing with REST APIs.

Figure 10-9 Flowchart of using image processing with REST APIs



Creating an Account and Enabling OBS

Before using image processing, you must have a cloud service account and enable OBS. Once OBS is enabled, image processing is available for use. To use OBS properly, make sure that you have sufficient account balance or you have a valid payment method. If you have already enabled OBS and you can use it properly, skip this section.

Step 1 Visit the **Huawei Cloud official website**.

Step 2 Click **Sign Up** in the upper right corner.

Step 3 Complete the account creation as instructed.

After the creation is complete, you will be navigated to your information page.

Step 4 Create an OBS bucket, so that you can use image processing properly.

----End

Uploading an image

You can upload images using OBS Console, OBS Browser+, or REST APIs.

For details, see **Uploading an Object**.

Skip this section if the image to be processed already exists in your bucket.

NOTE

If you want to use a user-defined domain name for image processing, you need to grant anonymous users the read permission for the image. For more information, see **Granting All Accounts the Read Permission for Certain Objects**.

Processing the Image

This section describes the URL constitution to process images using REST APIs. Once you have enabled OBS successfully, you can call REST APIs to process images simply by putting a URL that complies with the command rules of image processing in a browser.

URL Composition

A URL consists of the OBS domain name, bucket name, the original image name, and processing command or style name.

Access with Commands

URL format: **https://***bucketName.endpoint***/***objectName***?x-image-process=image/***commands*

- *endpoint* represents the endpoint address of the region where the bucket is located. You can obtain the endpoint from the bucket's basic information. For more information, see **Regions and Endpoints**.
- *bucketName* represents the name of the bucket that accommodates the image to be processed on OBS.
- *objectName* represents the name of the original image stored in the *bucketName* bucket. The name suffix must be supported by image processing.
- *commands* represents the processing commands. Three types of **delimiters** can be used between commands or command parameters. If no commands are specified, the original image will be returned.

Example URL: **https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/crop,x_100,y_50**

- **Delimiters**

Delimiters are used to separate one field from another in the commands in a URL. For details, see **Table 10-2**.

Table 10-2 Delimiters

Name	Character	Sequence	Description
Parameter delimiter	_	Fixed	Delimiter between a parameter and its value
Command delimiter	,	Irrelevant	Delimiter between command parameters

Name	Character	Sequence	Description
Pipe delimiter	/	Relevant	Delimiter between two processing commands. See **Pipes**.

- **Pipes**

If an image needs to be processed by multiple operations (for example, it needs to be cropped first and then resized), the operation commands must be separated with pipe delimiters (/). The operations (also pipes) will be executed sequentially from left to right.

For example, **https://image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_100,h_100/quality,q_80** are two pipes executed from left to right. The output of the left command becomes the input of the right command.

Access with a Style Name

URL format: **https://***bucketName.endpoint***/***objectName***?x-image-process=style/***stylename*

- *endpoint* represents the endpoint address of the region where the bucket is located. You can obtain the endpoint from the bucket's basic information. For more information, see **Regions and Endpoints**.
- *bucketName* is the bucket name on OBS.
- *objectName* represents the name of the original image stored in the *bucketName* bucket. The name suffix must be supported by image processing.
- *stylename* is the name of an existing style in the *bucketName* bucket on OBS Console. At present, REST APIs cannot be used to perform operations related to styles, such as creating, changing, or deleting styles.

Example URL: **https://image-demo.obs.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=style/stylename**

10.1.6 Getting Information About an Image

You can get the information about an image only by making an API call.

The image information you can get includes the width, height, size, and format of an image. If the image contains **EXIF Data**, the EXIF data will be returned in JSON.

This operation is represented by **info**.

Example

Below gives an example of querying information about the **example.jpg** image:

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/info

```
{
    "format":"JPEG",
    "height":2000,
    "size":1046583,
    "width":2668
}
```

10.1.7 Getting the Average RGB Value of an Image

You can get the average RGB value of an image only by making an API call.

The returned average RGB value is a hexadecimal string in JSON.

This operation is represented by **average-hue**.

Example

Below gives an example of querying the average RGB value of the **example.jpg** image:

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/average-hue

If information similar to the following is displayed, the average RGB value of the image is obtained:

```
{"RGB":"#28577D"}
```

Original **example.jpg** image:

Color corresponding to the average RGB value (#28577D):

10.1.8 Setting Image Effects

Setting the Brightness

You can use the GUI or code mode on OBS Console or make an API call to adjust the brightness of an image.

Table 10-3 describes the parameters.

This operation is represented by **bright**.

Table 10-3 Brightness parameters

Parameter	Value Description	Code Example
value	It adjusts the brightness of an image and ranges from **-100** to **100**. The original brightness value is **0**. A larger value means a higher level of brightness.	image/bright,50

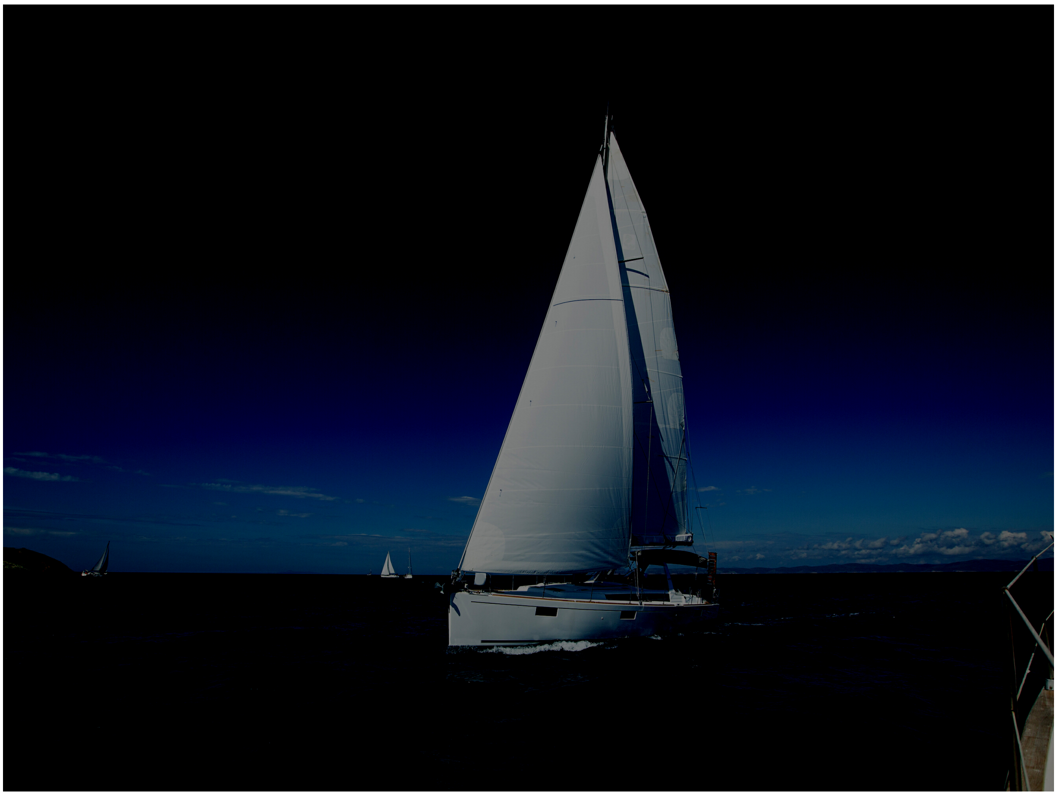
Examples of Setting the Brightness

- Set the brightness to **50**.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/bright,50

- Set the brightness to **-50**.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/bright,-50



Setting the Contrast

You can use the GUI or code mode on OBS Console or make an API call to adjust the contrast of an image.

Table 10-4 describes the parameters.

This operation is represented by **contrast**.

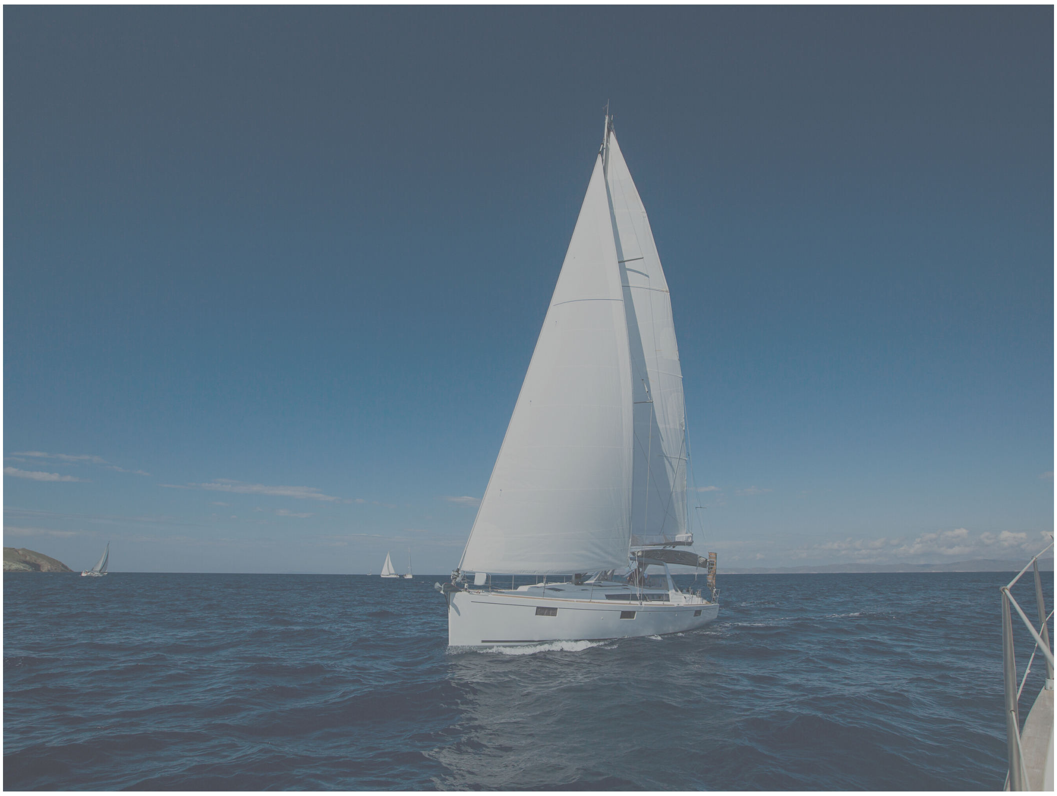
Table 10-4 Contrast parameters

Parameter	Value Description	Code Example
value	It adjusts the contrast of an image and ranges from **-100** to **100**. The original contrast value is **0**. A larger value means a higher level of contrast.	image/contrast,-50

Example of Setting the Contrast

- Set the contrast to **-50**.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/contrast,-50



Setting the Sharpness

You can use the GUI or code mode on OBS Console or make an API call to adjust the sharpness of an image.

Table 10-5 describes the parameters.

This operation is represented by **sharpen**.

Table 10-5 Sharpness parameters

Parameter	Value Description	Code Example
value	It adjusts the sharpness of an image and ranges from **50** to **399**. The recommended value is **100**. Technically, a larger value means a clearer image, but a too large value will make the image grainy.	image/sharpen,100

Example of Setting the Sharpness

- Set the sharpness to **100**.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/sharpen,100

Setting the Blur

You can use the GUI or code mode on OBS Console or make an API call to adjust the blur of an image.

Table 10-6 describes the parameters.

This operation is represented by **blur**.

Table 10-6 Blur parameters

Parameter	Value Description	Code Example
r	It represents the blur radius. Its value ranges from **1** to **50**. A larger radius means a larger blurred area.	image/blur,r_3,s_2
s	It represents the standard deviation of normal distribution. Its value ranges from **1** to **50**. A large standard deviation significantly blurs, while a small standard deviation blurs less.	

NOTE

In GUI mode, the parameter **r** and **s** increase or decrease simultaneously.

Example of Setting the Blur

- Set **r** to **3** and **s** to **2**.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/blur,r_3,s_2

Setting the Grayscale

You can edit code on OBS Console or make an API call to convert an image to grayscale.

Table 10-7 describes the parameters.

This operation is represented by **colorspace**.

Table 10-7 Grayscale parameters

Parameter	Value Description	Code Example
value	It changes the color mode of an image. You can set the parameter to **gray** to convert the image to grayscale.	image/colorspace,gray

Example of Setting the Grayscale

- Convert an image to grayscale.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/colorspace,gray

10.1.9 Resizing Images

You can use the GUI or code mode on OBS Console or make an API call to resize images. Images can be resized based on a specific rule or a fixed width, height, or percentage.

NOTE

- A long side refers to the side with a larger ratio of its original size to its target size, and a short side refers to the side with a smaller ratio. Assume that the original size of an image is 400 × 200 pixels and it is resized to 100 × 100 pixels. The ratio of 400 pixels to 100 pixels is 4 and that of 200 pixels to 100 pixels is 2, so the long side is 400 pixels and the short side is 200 pixels.
- For a target image after resizing, its long side cannot exceed 9,999 pixels, and the product of its width and height cannot exceed 24,999,999 pixels.
- If you only specify the height or width for resizing, the target image keeps the same aspect ratio and format as the original image.
- By default, resizing is not allowed to scale up an image. If you want an image to become larger after resizing, you need to set **limit** to **0** to obtain the enlarged image, or the original image will be returned. To do this, use the following format:

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_500,limit_0

Table 10-8 describes the parameters.

This operation is represented by **resize**.

Table 10-8 Resizing parameters

Parameter	Value Description	Code Example
m	Type of resizing. The value can be **lfit** (the default value), **mfit**, **fill**, **pad**, or **fixed**. ● **lfit**: Specify a rectangle with a given width (indicated by **w**) and height (indicated by **h**), lock the aspect ratio, and obtain the largest image in the rectangle. ● **mfit**: Specify a rectangle with a given width (indicated by **w**) and height (indicated by **h**), lock the aspect ratio, and obtain the smallest image in the rectangle's extended area. ● **fill**: Specify a rectangle with a given width (indicated by **w**) and height (indicated by **h**) and lock the aspect ratio. Obtain the smallest image in the rectangle's extended area, and center and crop the image. **fill**-based resizing actually centers and crops a target image resized with **mfit**. ● **pad**: Specify a rectangle with a given width (indicated by **w**) and height (indicated by **h**) and lock the aspect ratio. Obtain the largest image in the rectangle and fill the blank area with color. **pad**-based resizing actually fills the blank area of a target image resized with **lfit**. ● **fixed**: Resize an image based on a fixed width and height. ● **ratio**: Specify an aspect ratio (a ratio of **w** to **h**), in the range of **1** to **1000**, and	image/resize,m_lfit,h_100,w_100

Parameter	Value Description	Code Example
	obtain the largest image that meets the specified ratio.	
p	Percentage of the aspect ratio, in the range of **1** to **1000**. If the value is: - < 100: The image is scaled down. - = 100: The image is kept unchanged in size. - > 100: The image is scaled up.	image/resize,p_50
h	Height of the target image, in the range of **1** to **9999**. The product of the target image's width and height cannot exceed 24,999,999.	image/resize,m_lfit,h_100
w	Width of the target image, in the range of **1** to **9999**. The product of the target image's width and height cannot exceed 24,999,999.	image/resize,m_fixed,h_100,w_100
l	The long side of the target image, in the range of **1** to **4096**. The product of the target image's width and height cannot exceed 24,999,999. The long side has a specified value, and the short side is scaled based on the ratio.	image/resize,l_100
s	The short side of the target image, in the range of **1** to **4096**. The product of the target image's width and height cannot exceed 24,999,999. The short side has a specified value, and the long side is scaled based on the ratio.	image/resize,s_100

Parameter	Value Description	Code Example
color	Color for filling the blank area after resizing. This parameter can be used when you set **m** to **pad**. The value is a hexadecimal code, from **000000** to **FFFFFF** (default value, representing white).	image/ resize,m_pad,h_100,w_ 100,color_FF0000
limit	Whether to limit the size of the target image when the target image is larger than the original one. The value can be **0** or **1** (default value). ● **0**: The size is not limited. ● **1**: The size is limited.	image/ resize,p_150,limit_0

⚠ CAUTION

If a resized image is aliased, you can add **/marker,u_plus** to the end of the image processing URL for optimization.

For example, the image processing URL is as follows:

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_fixed,w_2668,h_1999,limit_0

After the parameter is added, the link is as follows:

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_fixed,w_2668,h_1999,limit_0/marker,u_plus

Examples

- Set **h** to **100** and **m** to **lfit** (the default value) to process the width proportionally.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_lfit,h_100

- Lock the aspect ratio and specify the short side to resize the image into 100 × 100 pixels.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_lfit,h_100,w_100

- Set the long side size to **100** and scale the short side based on the ratio.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,l_100

- Fix the width and height, center and crop the image, resize the image into 100 × 100 pixels.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_fill,h_100,w_100

- Fix both the width and height to **100**.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_fixed,h_100,w_100

- Fix the width and height. Resize the image into 100 × 100 pixels by specifying the short side and fill the blank area with white.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_pad,h_100,w_100

- Fix the width and height. Resize the image into 100 × 100 pixels by specifying the short side and fill the blank area with red.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_pad,h_100,w_100,color_FF0000

- Scale up the image to 150% of its original size and set **limit** to **0** to obtain the enlarged image.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,p_150,limit_0

- Set **p** to **30** to scale down the image to 30% of its original size.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,p_30

- Set an image's aspect ratio to 3:2.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_ratio,w_3,h_2

Resizing an Image by Editing Code on OBS Console

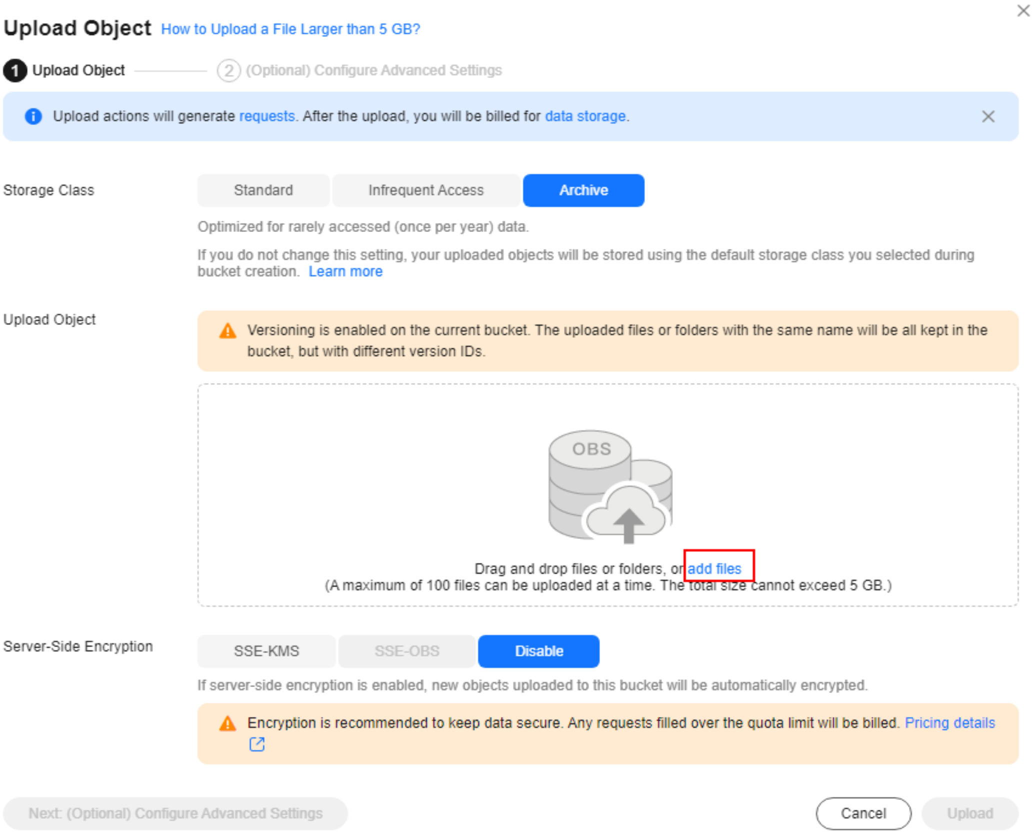
Step 1 On the console homepage, click **Service List** in the upper left corner and choose **Storage** > **Object Storage Service**.

Step 2 In the bucket list, click the bucket you want to go to the **Objects** page.

Step 3 Click **Upload Object** to open the **Upload Object** dialog box.

Step 4 Click **add files** marked by red box in **Figure 10-10** to open the local file browser.

Figure 10-10 Uploading an image



Upload Object How to Upload a File Larger than 5 GB?

1 Upload Object — 2 (Optional) Configure Advanced Settings

Upload actions will generate requests. After the upload, you will be billed for data storage.

Storage Class: Standard | Infrequent Access | **Archive**

Optimized for rarely accessed (once per year) data.

If you do not change this setting, your uploaded objects will be stored using the default storage class you selected during bucket creation. Learn more

Upload Object: Versioning is enabled on the current bucket. The uploaded files or folders with the same name will be all kept in the bucket, but with different version IDs.

Drag and drop files or folders, or add files
(A maximum of 100 files can be uploaded at a time. The total size cannot exceed 5 GB.)

Server-Side Encryption: SSE-KMS | SSE-OBS | **Disable**

If server-side encryption is enabled, new objects uploaded to this bucket will be automatically encrypted.

Encryption is recommended to keep data secure. Any requests filled over the quota limit will be billed. Pricing details

Step 5 Select the image that you want to upload and click **Open**.

Step 6 Click **Upload** to upload the image. The uploaded image is displayed in the object list.

Step 7 In the navigation pane, choose **Data Processing** > **Image Processing**.

Step 8 Click **Create** to go to the style editing page.

Step 9 Set **Style Name** to **style002** and choose **Code** for **Edit Mode**.

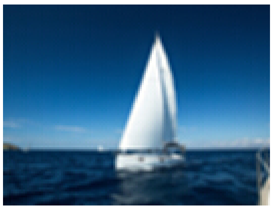
Step 10 In the code input box, type the following command for resizing.

Specify a rectangle whose width (indicated by **w**) and height (indicated by **h**) are both 100. Lock the aspect ratio, and obtain the smallest image in the extended area of the 100 × 100 rectangle.

```
image/resize,m_mfit,h_100,w_100
```

The style effect will be displayed on the right in real time. **Figure 10-11** shows the final style effect.

Figure 10-11 Effect of style **style002**

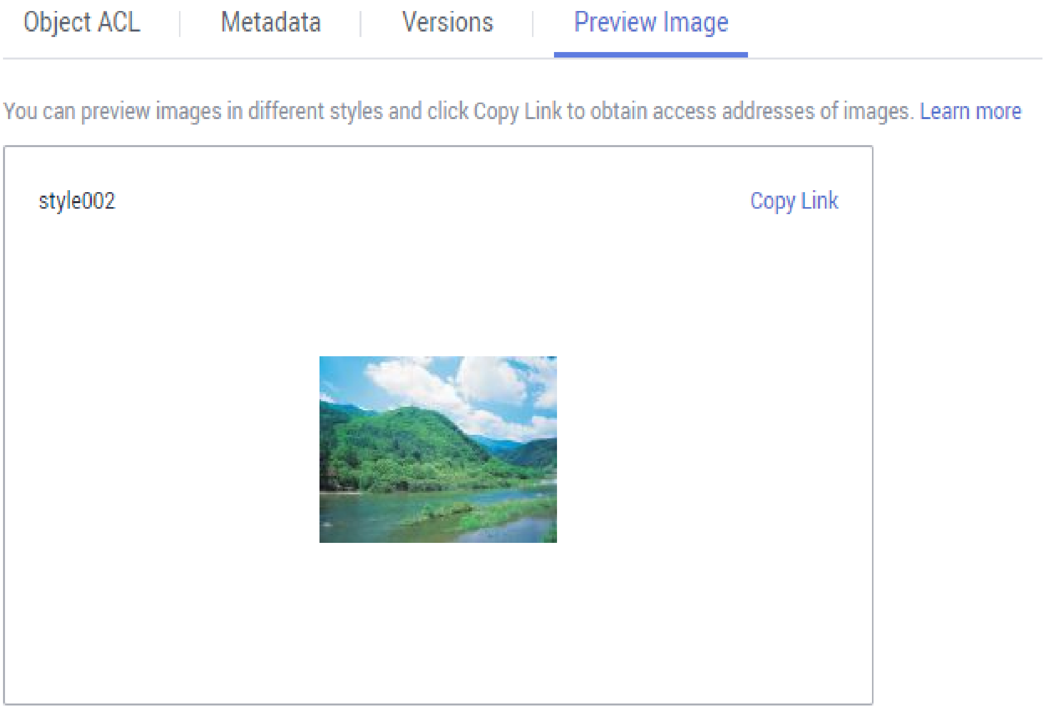


Step 11 Click **OK** to save the style. The new style **style002** will be displayed in the style list.

Step 12 In the navigation pane, choose **Objects**. Click **mountain.jpg** in the object list to go to the file details page.

Step 13 Click the **Preview Image** tab to preview the effect of the image with style **style002** applied.

Figure 10-12 Preview Image



You can preview images in different styles and click Copy Link to obtain access addresses of images. Learn more

Step 14 In the image style card, click **Copy Link**. When **Copied successfully.** is prompted, you will have the address for accessing the image with the style applied.

----End

10.1.10 Rotating Images

Setting the Rotation

You can edit code on OBS Console or make an API call to rotate images clockwise.

Table 10-9 describes the parameters.

This operation is represented by **rotate**.

Table 10-9 Rotation description

Parameter	Value Description	Code Example
value	It defines the angle of a clockwise rotation and ranges from **0** to **360**. The default value is **0**, which means the image is not rotated. A larger value means the image is clockwise rotated by a larger angle.	image/rotate,90

NOTE

- After rotation, dimensions of an image may increase.

Examples of Setting the Rotation

Set the width to **100** and the rotation angle to **90**.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_100/rotate,90

Set the width to **100** and the rotation angle to **220**.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_100/rotate,220

Setting the Adaptive Orientation

You can use the GUI or code mode on OBS Console or make an API call to use adaptive orientation.

Images shot by cameras or smartphones may contain **EXIF Data**, like an orientation parameter. Since an image's rotation information is recorded in the orientation parameter, a browser can automatically rotate the image to the right position based on the orientation parameter.

With adaptive orientation configured, images that contain the orientation parameter will be rotated automatically. For details, see **Table 10-10**.

This operation is represented by **auto-orient**.

Table 10-10 Adaptive orientation description

Parameter	Value Description	Code Example
value	The value can be **0** or **1** (default value). **0**: Adaptive orientation is not configured. The image will not rotate automatically and will keep the default orientation. **1**: Adaptive orientation is configured. The image will rotate automatically before being resized.	image/resize,w_100/auto-orient,0

NOTE

- The adaptive orientation can be configured only when the height and width of an image are both smaller than 4,096.
- If an image's EXIF data does not contain any rotation parameters or the image has no EXIF data at all, the **auto-orient** parameter does not work for the image.

Example of Setting the Adaptive Orientation

- Set the width of an image to **100**, and do not configure the adaptive orientation.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_100/auto-orient,0

Flipping Images

You can edit code on OBS Console or make an API call to flip images.

Images can be flipped horizontally or vertically. **Table 10-11** describes the parameters.

This operation is represented by **flip**.

Table 10-11 Flipping description

Parameter	Value Description	Code Example
value	If this parameter is set to **vertical**, an image is flipped vertically. If this parameter is set to **horizontal**, an image is flipped horizontally.	image/flip,vertical

NOTE

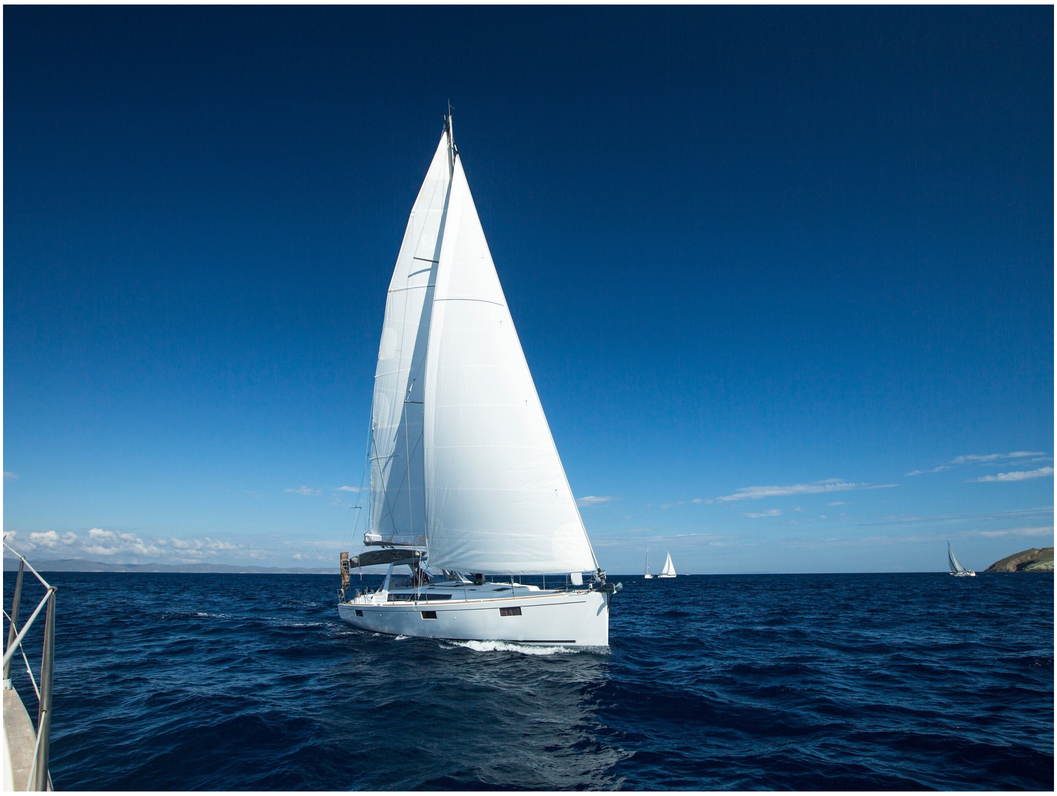
- After flipping, dimensions of an image may increase.

Examples of Flipping Images

Flip an image horizontally.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/flip,horizontal

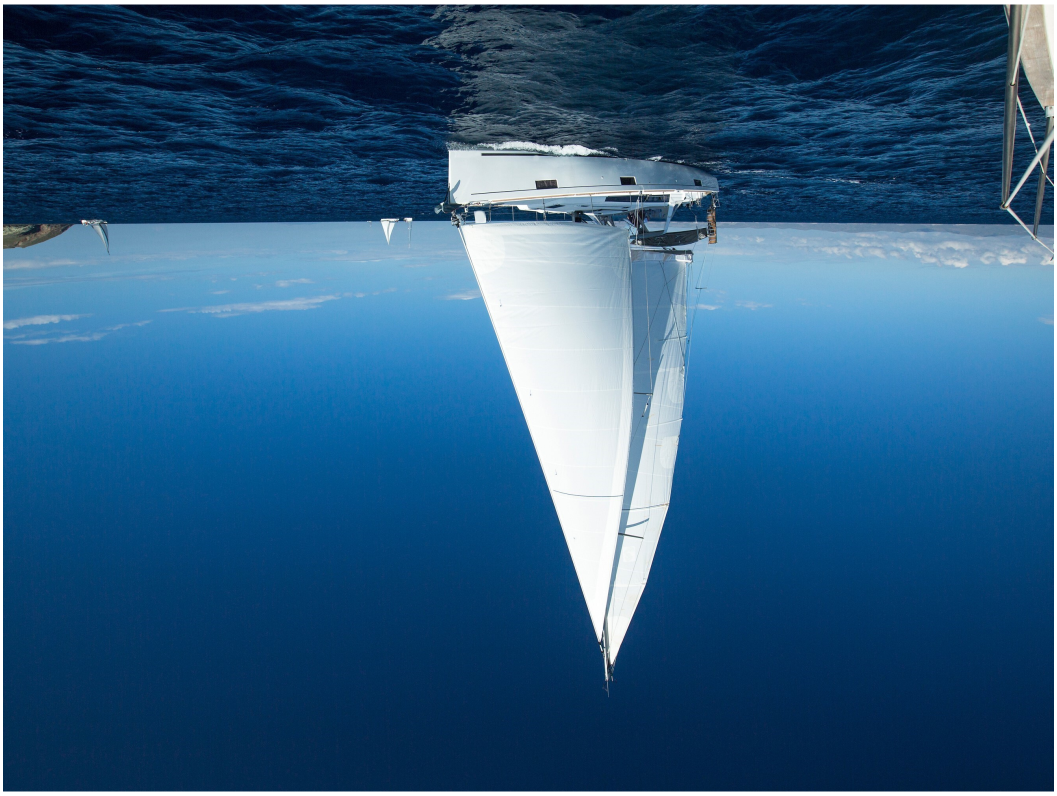
Figure 10-13 Flipping an image horizontally



Flip an image vertically.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/flip,vertical

Figure 10-14 Flipping an image vertically



10.1.11 Cropping Images

Common Cropping

You can edit code on OBS Console or make an API call to crop images.

You can start at any point on an image and crop the image into a rectangle with specified width and height. Table 10-12 describes the parameters.

This operation is represented by **crop**.

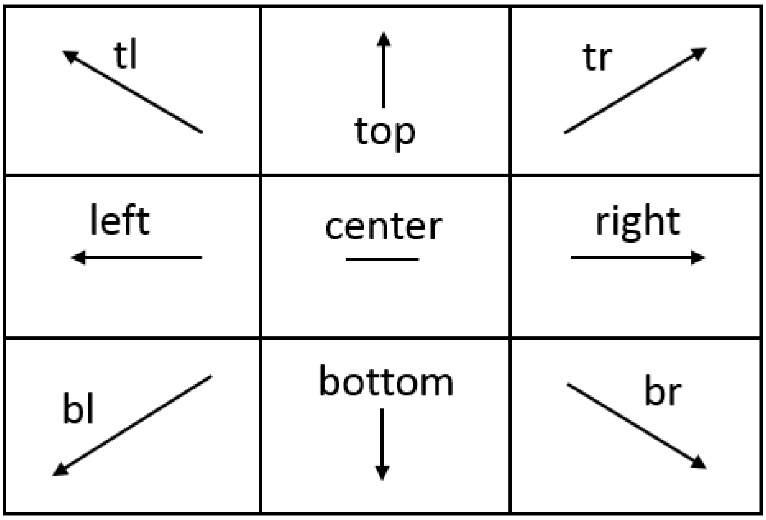
Table 10-12 Common cropping parameters

Parameter	Value Description	Code Example
g	The location where cropping starts. **g** can be set to **tl**, **top**, **tr**, **left**, **center**, **right**, **bl**, **bottom**, or **br**. Figure 10-15 shows a 3×3 grid illustrating these values. Each value locates at the top left corner of the cell.	image/crop,x_10,y_10,w_200,h_200,g_br
h	Cropping height. Its value ranges from **0** to *Original height*.	

Parameter	Value Description	Code Example
w	Cropping width. Its value ranges from **0** to *Original width*.	
x	x-coordinate of the start point. The top left corner is the default origin. The parameter value ranges from **0** to *Original image width*.	
y	y-coordinate of the start point. The top left corner is the default origin. The parameter value ranges from **0** to *Original image height*.	

Figure 10-15 shows the cropping origins.

Figure 10-15 3×3 grid of cropping origins



NOTE

- If x is larger than the origin width, or y is larger than the origin height, the cropping cannot be executed and a fault will be returned.
- If h is larger than the origin height and w is larger than the original width, the image will be cropped to the boundaries.

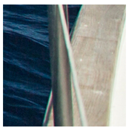
Examples of Common Cropping

- Set the start point of cropping to **(1000, 500)**, and set both the width and height to **1000**.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/crop,x_1000,y_500,w_1000,h_1000

- Set the start point of cropping to **(10, 10)** in the bottom right (**br**) cell, and set both the width and height to **200**.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/crop,x_10,y_10,w_200,h_200,g_br



Inscribed Circles

You can edit code on OBS Console or make an API call to get an inscribed circle of an image.

Choose the image center as the center of a circle, and crop the image according to the specified radius. The image then is cropped into a circle. **Table 10-13** describes the parameters.

This operation is represented by **circle**.

Table 10-13 Inscribed circle parameters

Parameter	Value Description	Code Example
r	Radius specified for the inscribed circle. Its value ranges from **0** to *Half of the image's short side*.	image/circle,r_100

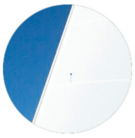
NOTE

- If the output format is set to JPG, the area outside the inscribed circle is white. If the output format is set to a vector format such as PNG, WebP, or BMP, the area outside the inscribed circle is transparent. PNG is the recommended output format.
- Even if **r** is larger than half of the short side, the generated inscribed circle is still the image's largest inscribed circle (with a radius equal to half of the short side).

Example of Making an Inscribed Circle

Set the radius to **100** and output format to **jpg** (with the area outside the inscribed circle being white).

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/circle,r_100



Indexcropping

You can edit code on OBS Console or make an API call to indexcrop images.

Set the top left corner of the image as the starting point. Set x-axis overlapping with the width, and y-axis overlapping with the height. Crop the image into several consecutive partitions horizontally or vertically, each having equal width or height. Get the partition you want according to the index. **Table 10-14** describes the parameters.

This operation is represented by **indexcrop**.

Table 10-14 Indexcropping parameters

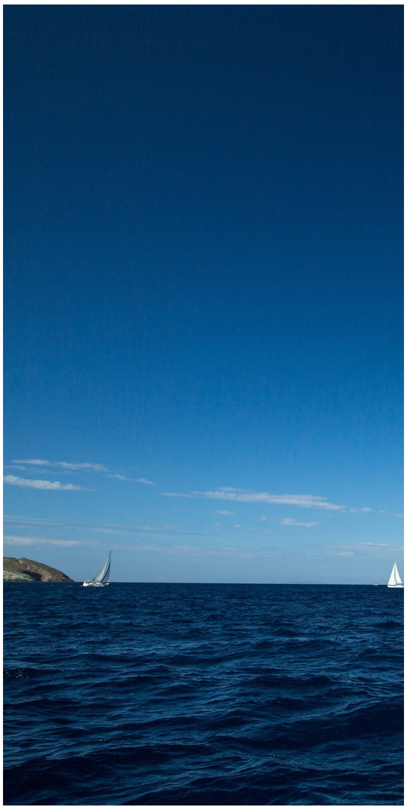
Parameter	Value Description	Code Example
x	Width of each partition in horizontal cropping, ranging from **1** to *Original image width*. **x** and **y** cannot be chosen at the same time.	image/indexcrop,x_1000,i_0

Parameter	Value Description	Code Example
y	Height of each partition in vertical cropping, ranging from **1** to *Original image height*. **x** and **y** cannot be chosen at the same time.	
i	If there are *n* partitions in total, the value of **i** ranges from **0** to *n*–*1*. When **i** is set to **0**, you obtain the first partition. If you set **i** to a value larger than *n*–*1*, the original image will be returned.	

Examples of Indexcropping

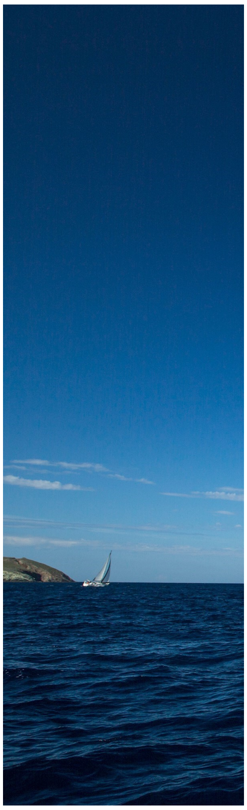
- Indexcrop an image into partitions with a width of **1000** horizontally and choose the first partition.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/indexcrop,x_1000,i_0

- Indexcrop an image into partitions with a width of **600** horizontally and choose the first partition.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/indexcrop,x_600,i_0



Rounded Corner Cropping

You can edit code on OBS Console or make an API call to round corners of an image.

This operation allows you to cut off the corners of an image based on the specified radiuses. You can configure the radius of image corners to round in two ways. For details, see **Table 10-15**.

This operation is represented by **rounded-corners**.

Table 10-15 Rounded corner parameters

Parameter	Value Description	Code Example
r	It indicates both horizontal and vertical radiuses of image corners you will round. You can configure it by using the number of pixels (for example, **200**) or the percentage (for example, **25p**). The number of pixels ranges from **1** to **4096**. If this number is greater than half of the original image's short side, the half of the short side is applied. The percentage ranges from **1p** to **50p**. **r** cannot be used together with **rx** or **ry**.	image/rounded-corners,r_100
rx	It indicates the horizontal radius of image corners you will round. You can configure it by using the number of pixels (for example, **200**) or the percentage (for example, **25p**). The number of pixels ranges from **1** to **4096**. If this number is greater than half of the original image's short side, the half of the short side is applied. The percentage ranges from **1p** to **50p**. **rx** must be used together with **ry**.	image/rounded-corners,rx_100,ry_200
ry	It indicates the vertical radius of image corners you will round. You can configure it by using the number of pixels (for example, **200**) or the percentage (for example, **25p**). The number of pixels ranges from **1** to **4096**. If this number is greater than half of the original image's short side, the half of the short side is applied. The percentage ranges from **1p** to **50p**. **ry** must be used together with **rx**.	

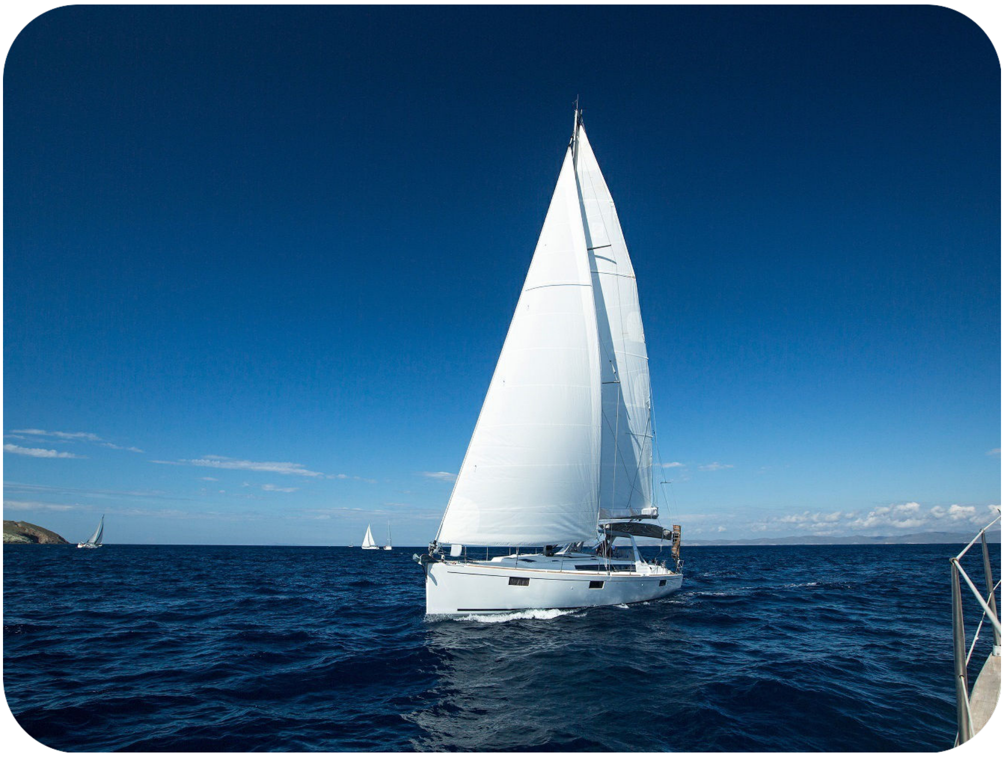
NOTE

If the output image format is JPG, the cut-out corner area is white. If the output image format is PNG, WebP, or BMP, the cut-out corner area is transparent. You are advised to save rounded corner images in PNG.

Examples of Rounded Corner Cropping

- Round the corners of image **example.jpg** based on the horizontal and vertical radiuses of **100** and save the new image as PNG.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/rounded-corners,r_100/format,png

- Round the corners of image **example.jpg** based a horizontal radius of **100** and a vertical radius of **200**.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/rounded-corners,rx_100,ry_200

10.1.12 Watermarking Images

Public Parameters

You can use the GUI or code mode on OBS Console or make an API call to configure public parameters for watermarking.

You can add a text or image watermark to the original image.

The Base64 code for URL transmission applies to paths of content and fonts of text watermarks, or paths of image watermarks. It is not advisable to put standard Base64 code directly into the URL for transmission. In Base64 encoding for URL transmission, contents are coded into character strings by standard Base64 code. After verifying these strings, replace the plus sign (+) with hyphen (-), and slash (/) with underline (_). For details about encoding, see those specified in RFC4648.

This operation is represented by **watermark**.

Public parameters are those applicable to both image watermarks and text watermarks. You can add these two types of watermarks to the same image. **Table 10-16** describes the parameters.

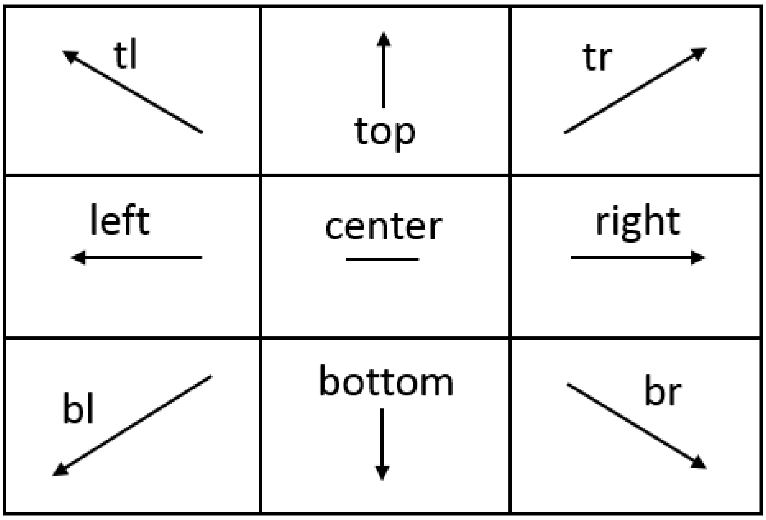
Table 10-16 Public parameters

Parameter	Value Description	Code Example
g	Optional. **g** represents the watermark location. It can be set to **tl**, **top**, **tr**, **left**, **center**, **right**, **bl**, **bottom**, or **br**. The default value is **tl**. **Figure 10-16** shows a 3×3 grid illustrating these values. Each value locates at the top left corner of the cell.	image/watermark,image_aW1hZ2UtZGVtby 9sb2dvLnBuZw==,g_br,t_90,x_10,y_10
x	Optional. **x** represents the horizontal distance from the image edge. By default, the origin is at the top left corner. The value ranges from **0** to **4096**, in pixels. The default value is **10**.	
y	Optional. **y** represents the vertical distance from the image edge. By default, the origin is at the top left corner. The value ranges from **0** to **4096**, in pixels. The default value is **10**.	
voffset	Optional. **voffset** represents the vertical offset from the horizontal centerline of the image. It moves a watermark upward or downward from the horizontal centerline of the image. The value ranges from **-1000** to **1000**, in pixels. The default value is **0**. **voffset** is meaningful only when **g** is set to **left**, **center**, or **right**.	
align	Optional. **align** represents the alignment of the text and image watermarks. The value can be **0** (default value), **1**, or **2**. ● **0**: top alignment ● **1**: center alignment ● **2**: bottom alignment	

Parameter	Value Description	Code Example
order	Optional. **order** represents the placement order of the text and image watermarks. The value can be **0** (default value) or **1**. ● **0**: image watermark in front of the text watermark ● **1**: text watermark in front of the image watermark	
t	Optional. It represents the transparency of a watermark. Its value ranges from **0** to **100**. The default value is **100**, indicating the watermark is not transparent at all.	
interval	Optional. It represents the distance between a text watermark and an image watermark. Its value ranges from **0** to **1000**.	

Figure 10-16 shows a 3×3 grid illustrating the watermark locations.

Figure 10-16 3×3 grid of watermark locations



NOTE

If both a text watermark and an image watermark are added to the same image, you can configure **x**, **y**, and **voffset** to adjust the watermark locations and their layout.

Image Watermarks

You can pre-process an image watermark before adding it to an image.

The pre-processing operations include those described in **Resizing Images**, **Rotating Images**, and **Cropping Images**, but cropping an image watermark into an inscribed circle is not allowed. When resizing an image watermark, you can scale it to a specific percentage of the original image size.

Table 10-17 describes the image watermark parameters.

Table 10-17 Image watermark parameters

Parameter	Value Description	Code Example
image	Mandatory. It indicates the watermark image path. The watermark image path is *bucketName*/*objectName* (encoding required) or *bucketName*/*objectName***?x-image-process=***image*/*command* (encoding required). **NOTICE** The content must be URL-safe Base64 encoded in the **encodedObject = url_safe_base64_encode(object)** format. For example, object **panda.png** will be encoded as **cGFuZGEucG5n**.	image/resize,w_400/watermark,image_aW1h Z2UtZGVtby9sb2dvLnBuZz94LWltYWdlLXByb 2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA=,t_90, g_br,x_10,y_10
P	It indicates the percentage of the original image the image watermark will be scaled to. The value ranges from **1** to **100**. **NOTICE** **P** used here for resizing is case-sensitive. To adjust the size of an image watermark, see **Resizing Images** (except the **p** parameter).	image-demo/logo.png?x-image-process=image/resize,P_50

Examples of Making API Calls to Add Image Watermarks

- Use the watermark image **logo.png** (path: **hw-image-demo/logo.png**) and make the image watermark located at the bottom right, with a transparency of **90** and default horizontal and vertical margins of **10**.

  Relevant parameters:

  Watermark image: **hw-image-demo/logo.png**

  Safe Base64 code: **aHctaW1hZ2UtZGVtby9sb2dvLnBuZw==**

  Watermark position (bottom right): **g_br**

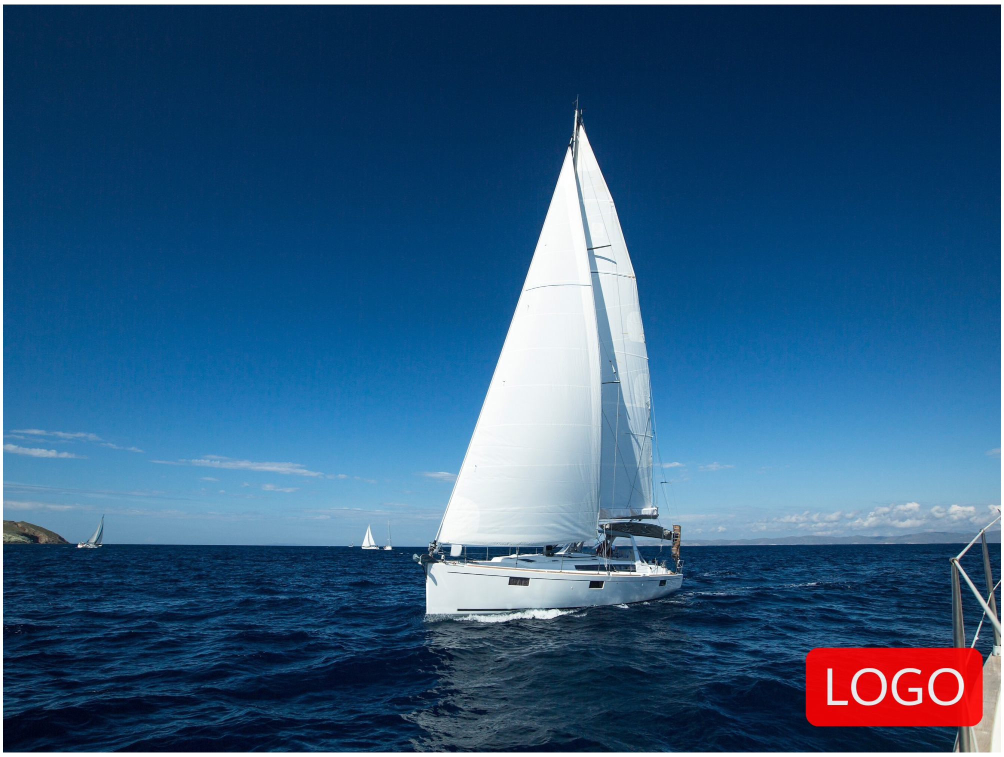
  Transparency: **t_90**

  Horizontal and vertical margins: **x_10,y_10**

  Request URL:

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/watermark,image_aHctaW1hZ2UtZGVtby9sb2dvLnBuZw==,g_br,t_90,x_10,y_10

Figure 10-17 Example 1



- Use the watermark image **logo.png** (path: **hw-image-demo/logo.png**) and resize the image watermark by setting its width to **50**. Keep other parameters the same as those in the previous example.

  Relevant parameters:

  Watermark image: **hw-image-demo/logo.png?x-image-process=image/resize,w_50**

  Safe Base64 code: **aHctaW1hZ2UtZGVtby9sb2dvLnBuZz94LWltYWdlLXByb2Nlc3M9aW1hZ2UvcmVzaXplLHdfNTA=**

  Watermark position (bottom right): **g_br**

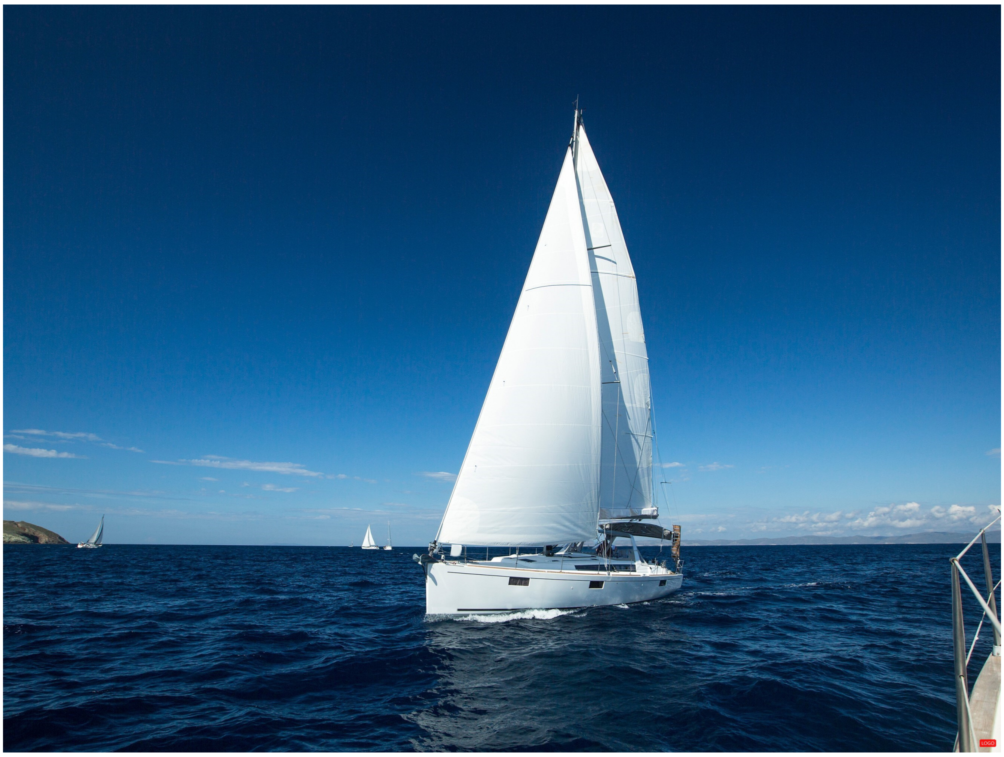
  Transparency: **t_90**

  Horizontal and vertical margins: **x_10,y_10**

  Request URL:

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/watermark,image_aHctaW1hZ2UtZGVtby9sb2dvLnBuZz94LWltYWdlLXByb2Nlc3M9aW1hZ2UvcmVzaXplLHdfNTA=,g_br,t_90,x_10,y_10

Figure 10-18 Example 2



- Use the watermark image **logo.png** (path: **hw-image-demo/logo.png**) and resize the image watermark to 50 percent of the original image's size. Keep other parameters the same as those in the previous example.

  Relevant parameters:

  Watermark image: **hw-image-demo/logo.png?x-image-process=image/resize,P_50**

  Safe Base64 code: **aHctaW1hZ2UtZGVtby9sb2dvLnBuZz94LWltYWdlLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfNTA=**

  Watermark position (bottom right): **g_br**

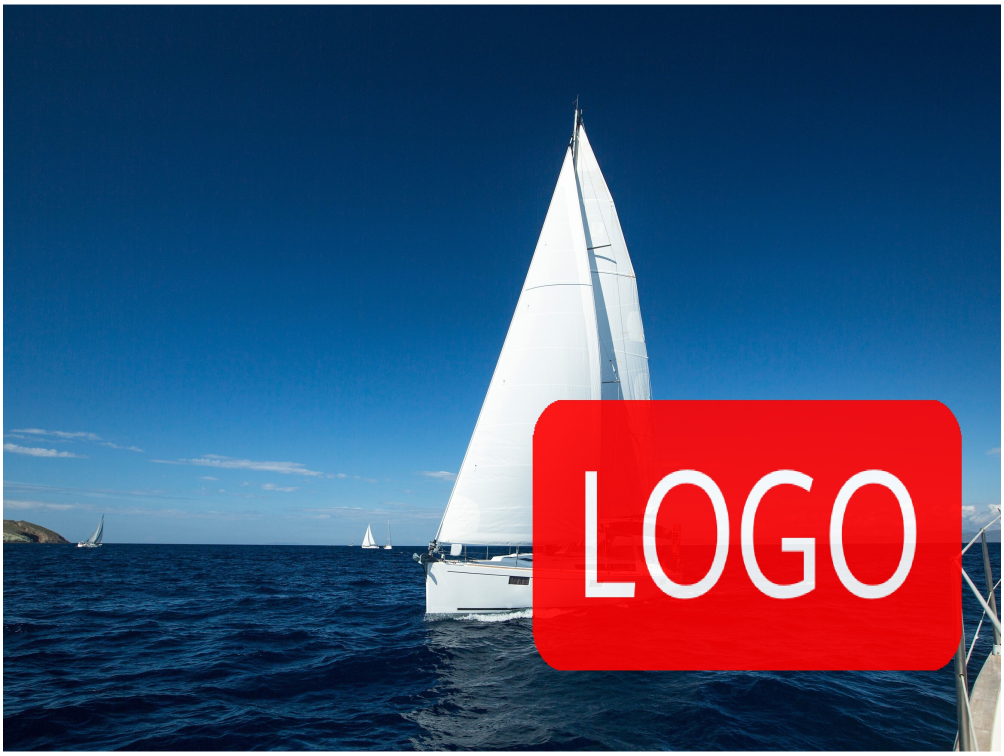
  Transparency: **t_90**

  Horizontal and vertical margins: **x_10,y_10**

  Request URL:

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/watermark,image_aHctaW1hZ2UtZGVtby9sb2dvLnBuZz94LWltYWdlLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfNTA=,g_br,t_90,x_10,y_10

Figure 10-19 Example 3



Example of Using the OBS Console GUI Mode to Add Image Watermarks

To add an image watermark to an image, you can create a watermark style on OBS Console and then apply it to the image. The following gives an example of creating a watermark style on OBS Console:

Step 1 Log in to OBS Console and click the required bucket name. In the navigation pane, choose **Data Processing** > **Image Processing**.

Step 2 Click **Create** to go to the style editing page.

Step 3 On the editing page, specify a style name, choose **GUI** for **Edit Mode**, and click **Watermark**.

> NOTE
>
> A style name consists of letters (uppercase or lowercase), digits, periods (.), underscores (_), and hyphens (-) and contains 1 to 256 characters, for example, **rotate_0001**.

Step 4 Configure the following watermark parameters.

- **Watermark Type**: Choose **Image watermark**.
- **Watermark Image Path**: Enter **e-share/image-demo/logo.png**.
- **Image Size (%)**: Set this parameter based on whether the watermark image is zoomed out.
- **Brightness**: Retain the default value **0**.
- **Contrast**: Retain the default value **0**.
- **Transparency**: Set this parameter to **90**.

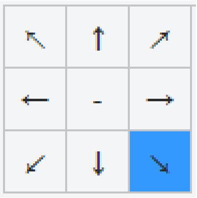
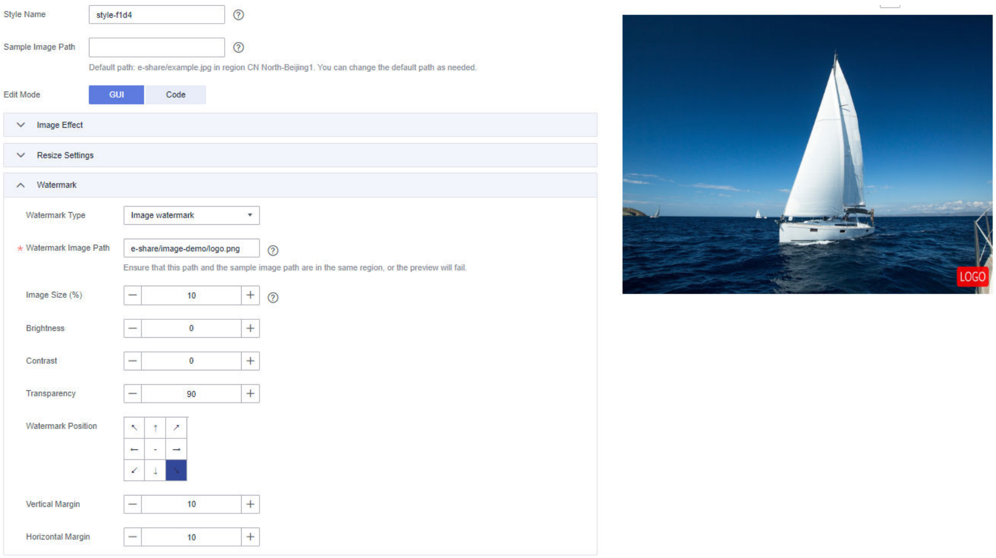
- **Watermark Position**: Select the arrow at the bottom right ().
- **Vertical Margin**: Retain the default value **10**.
- **Horizontal Margin**: Retain the default value **10**.

Figure 10-20 Configuring an image watermark



Style Name: style-f1d4

Sample Image Path:

Default path: e-share/example.jpg in region CN North-Beijing1. You can change the default path as needed.

Edit Mode: GUI | Code

Image Effect

Resize Settings

Watermark

Watermark Type: Image watermark

* Watermark Image Path: e-share/image-demo/logo.png

Ensure that this path and the sample image path are in the same region, or the preview will fail.

Image Size (%): 10

Brightness: 0

Contrast: 0

Transparency: 90

Watermark Position

Vertical Margin: 10

Horizontal Margin: 10

Step 5 Click **OK** to save the style. The new style will be displayed in the style list.

You can use the new watermark style to process images by referring to **Procedure 4: Using the Image Style**.

----End

Text Watermarks

You can configure the size, font, and color for text images. **Table 10-18** describes the parameters.

Table 10-18 Text watermark parameters

Parameter	Value Description	Code Example
text	Mandatory for adding text watermarks **NOTICE** The content must be URL-safe Base64 encoded in the **encodeText = url_safe_base64_encode(fontText)** format, with a maximum of 64 characters (16 full-width characters at most).	image/watermark,text_SGVsbG8g5Zu-54mH5pyN5YqhIQ,size_60,color_FF0000,type_ZmFuZ3poZW5nc2h1c29uZw==,g_center,rotate_30
size	Optional. It represents the size of a text watermark, ranging from **0** (exclusive) to **1000** (inclusive). The default value is **40**.	
type	Optional. It represents the font of a text watermark. Available values are those shown in **Table 10-19**. The default value is **wqy-zenhei** (**d3F5LXplbmhlaQ** after encoding). **NOTICE** - The content must be URL-safe Base64 encoded in the **encodeText = url_safe_base64_encode(fontType)** format. - Line breaks are currently not supported.	
color	Optional. It represents the color of a text watermark. The value is a six-digit hexadecimal code, from **000000** (default value, representing black) to **FFFFFF**.	
shadow	Optional. It represents the transparency of a text watermark. The value ranges from **0** (exclusive) to **100** (inclusive).	
fill	Optional. It represents the filling effect of a text watermark. The value can be **0** or **1**. ● **0**: filling not applied ● **1**: filling applied	
rotate	Optional. It represents the clockwise rotation angle of a text watermark. The value ranges from **0** (exclusive) to **360** (exclusive).	

Table 10-19 Text font encoding

Parameter	Value After URL-Safe Base64 Encoding	Value Description	Remarks
droidsansfallback	ZHJvaWRzYW5zZmFsbGJhY2s=	DroidSansFallback	According to Request for Comments (RFC), the fuller "=" can be omitted, and the value becomes **ZHJvaWRzYW5zZmFsbGJhY2s**.
fangzhengfangsong	ZmFuZ3poZW5nZmFuZ3Nvbmc=	FZFongSong	According to RFC, the fuller "=" can be omitted, and the value becomes **ZmFuZ3poZW5nZmFuZ3Nvbmc**.
fangzhengheiti	ZmFuZ3poZW5naGVpdGk=	FZSimHei	According to RFC, the fuller "=" can be omitted, and the value becomes **ZmFuZ3poZW5naGVpdGk**.
fangzhengkaiti	ZmFuZ3poZW5na2FpdGk=	FZKaiTi	According to RFC, the fuller "=" can be omitted, and the value becomes **ZmFuZ3poZW5na2FpdGk**.
fangzhengshusong	ZmFuZ3poZW5nc2h1c29uZw==	FZShuSong	According to RFC, the fuller "=" can be omitted, and the value becomes **ZmFuZ3poZW5nc2h1c29uZw**.
wqy-microhei	d3F5LW1pY3JvaGVp	WenQuanYi Micro Hei	-
wqy-zenhei	d3F5LXplbmhlaQ==	WenQuanYi Zen Hei	According to RFC, the fuller "=" can be omitted, and the value becomes **d3F5LXplbmhlaQ**.

Examples of Making API Calls to Add Text Watermarks

- Add the text watermark **Hello** to an image, with the text size set to **60**, color to red, and font to FZShuSong.

  Relevant parameters:

  Safe Base64 code: **SGVsbG8g5Zu**, namely **text_SGVsbG8g5Zu**

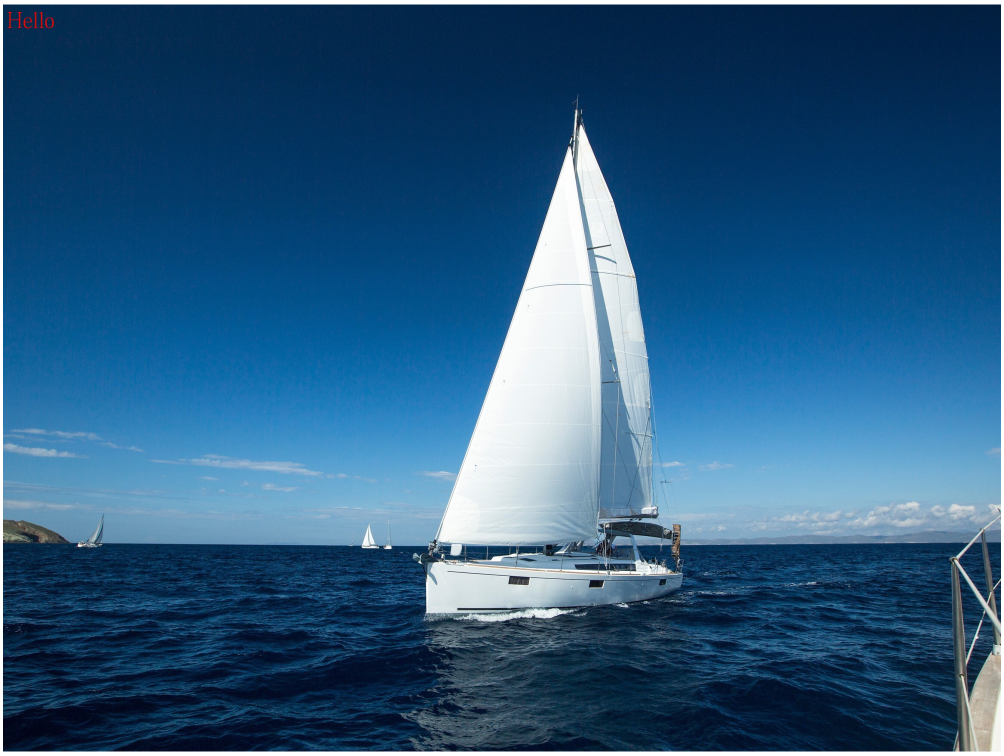
  Text size: **size_60**

  Text color: **color_FF0000**

  Text font: **type_ZmFuZ3poZW5nc2h1c29uZw==**

  Request URL:

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/watermark,text_SGVsbG8g5Zu,size_60,color_FF0000,type_ZmFuZ3poZW5nc2h1c29uZw==

- Clockwise rotate a text watermark through an angle of 30 degrees and add it to the center of an image. Keep other parameters the same as those in the previous example.

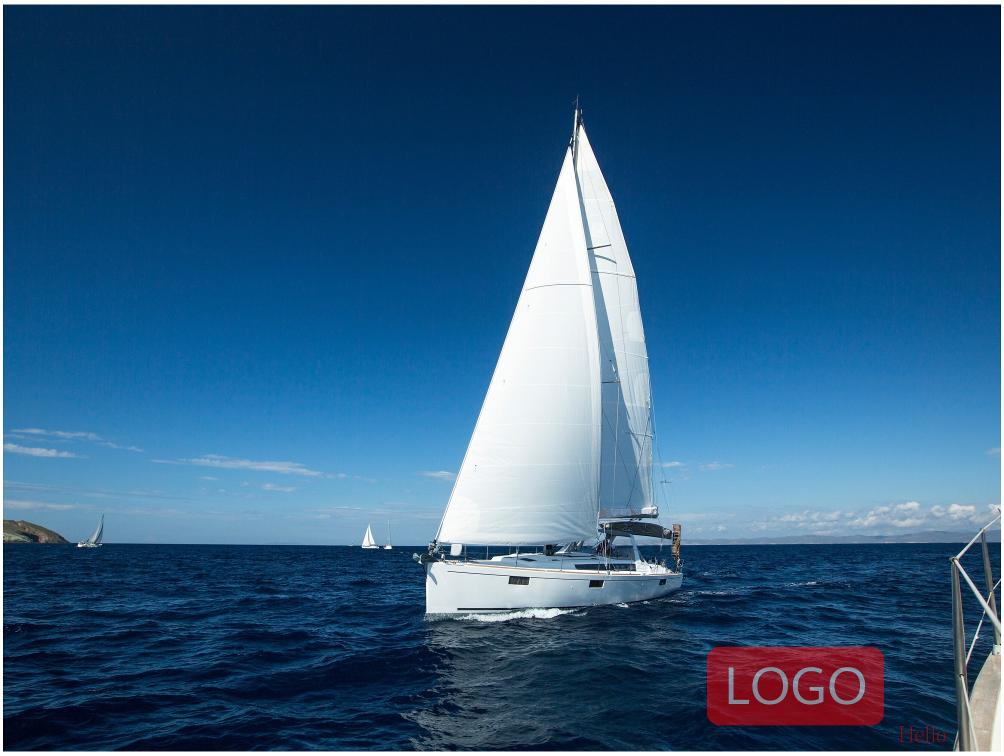
  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/watermark,text_SGVsbG8g5Zu,size_60,color_FF0000,type_ZmFuZ3poZW5nc2h1c29uZw==,g_center,rotate_30

- Add image and text watermarks at the same time. Add text watermark **Hello** at the bottom right, with the size set to **60**, shadow to **50**, and color to red.

  Add image watermark **logo.png**, with both horizontal and vertical margins set to **10**.

  Set the transparency of both text and image watermarks to **50** and put the image watermark in front and the text watermark behind, in the bottom alignment.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/watermark,image_aHctaW1hZ2UtZGVtby9sb2dvLnBuZw==,text_SGVsbG8g5Zu,size_60,color_FF0000,shadow_50,type_ZmFuZ3poZW5nc2h1c29uZw==,g_br,x_10,y_10,align_2,order_0

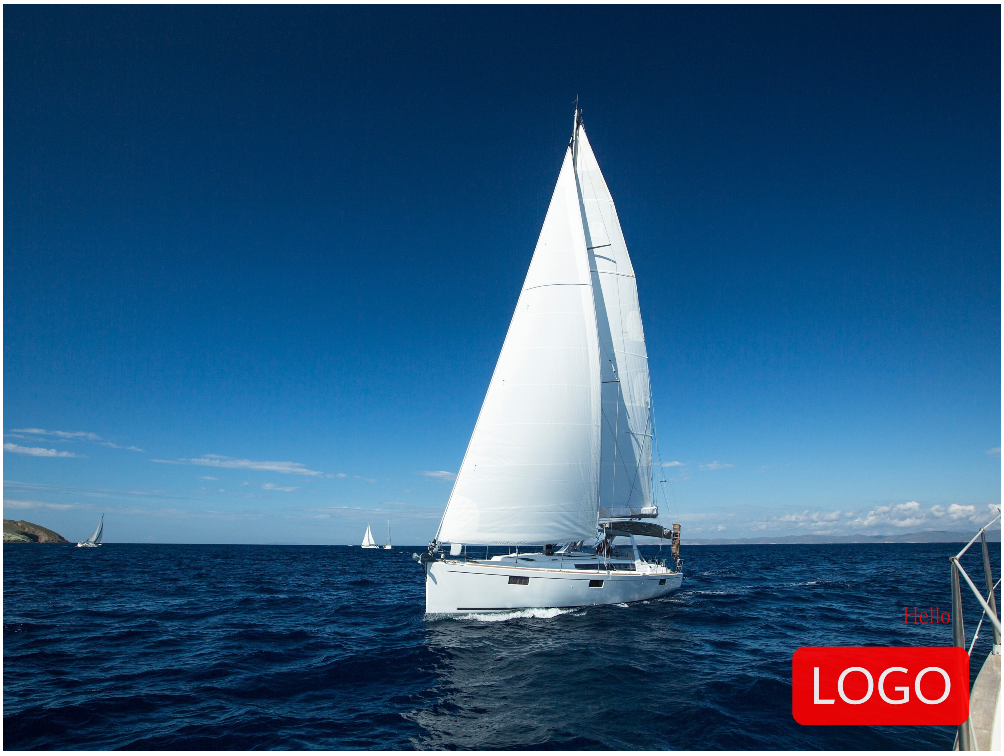


- Add an image watermark first and then a text watermark. Add text watermark **Hello** at the bottom right, with the size set to **60** and color to red.

  Add image watermark **logo.png**, with a horizontal margin of **40** and a vertical margin of **10**.

  Set the transparency of the image watermark to **90**.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/watermark,image_aHctaW1hZ2UtZGVtby9sb2dvLnBuZw==,g_br,t_90,x_45,y_10/watermark,text_SGVsbG8g5Zu,size_60,color_FF0000,type_ZmFuZ3poZW5nc2h1c29uZw==,g_br,x_0,y_330



Adding a Text Watermark in the GUI Mode on OBS Console

This case describes how to add a FZShuSong text watermark to the top left of an image in the GUI mode on OBS Console.

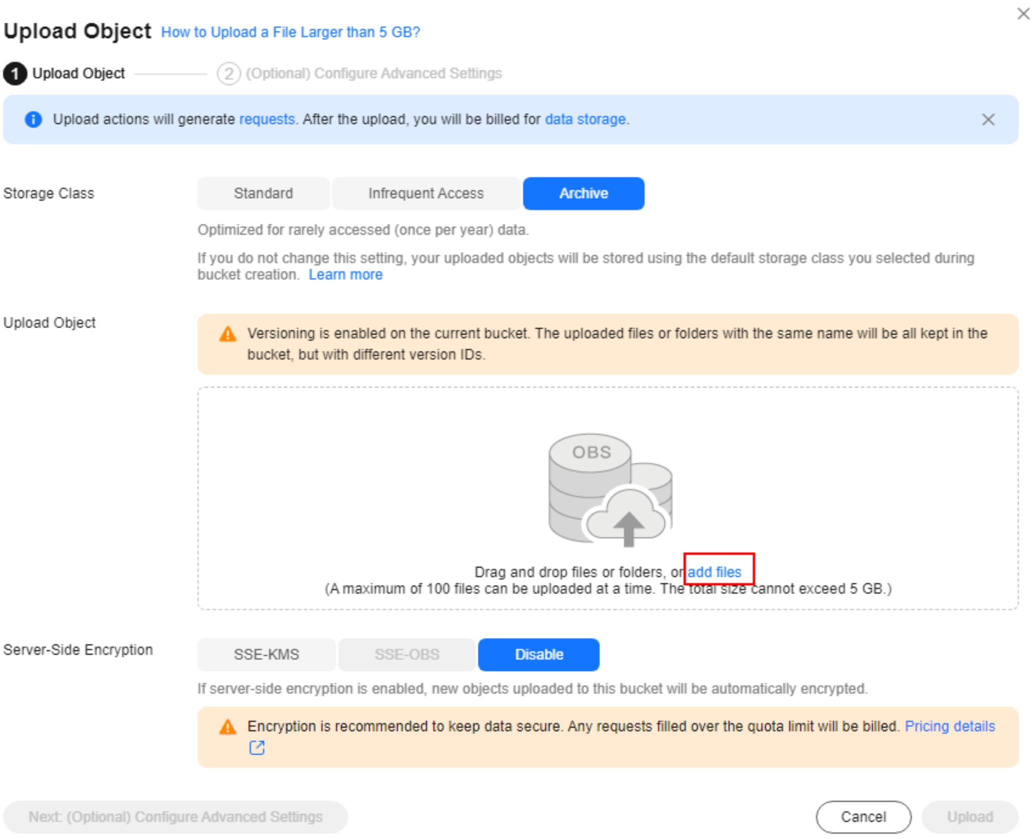
Step 1 On the console homepage, click **Service List** in the upper left corner and choose **Storage** > **Object Storage Service**.

Step 2 In the bucket list, click the bucket you want to go to the **Objects** page.

Step 3 Click **Upload Object** to open the **Upload Object** dialog box.

Step 4 Click **add files** marked by red box in **Figure 10-21** to open the local file browser.

Figure 10-21 Uploading an image



Upload Object How to Upload a File Larger than 5 GB?

1 Upload Object —— 2 (Optional) Configure Advanced Settings

Upload actions will generate requests. After the upload, you will be billed for data storage.

Storage Class: Standard | Infrequent Access | Archive (selected)

Optimized for rarely accessed (once per year) data.

If you do not change this setting, your uploaded objects will be stored using the default storage class you selected during bucket creation. Learn more

Upload Object: Versioning is enabled on the current bucket. The uploaded files or folders with the same name will be all kept in the bucket, but with different version IDs.

Drag and drop files or folders, or add files
(A maximum of 100 files can be uploaded at a time. The total size cannot exceed 5 GB.)

Server-Side Encryption: SSE-KMS | SSE-OBS | Disable (selected)

If server-side encryption is enabled, new objects uploaded to this bucket will be automatically encrypted.

Encryption is recommended to keep data secure. Any requests filled over the quota limit will be billed. Pricing details

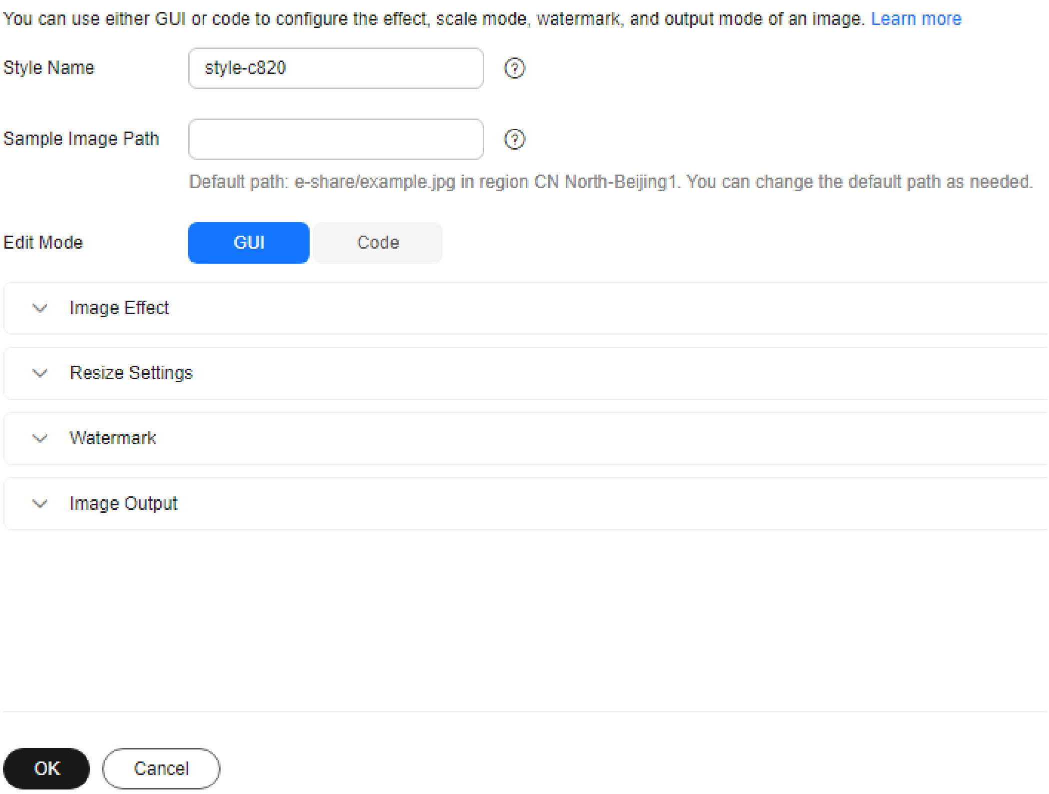
Step 5 Select the image that you want to upload and click **Open**.

Step 6 Click **Upload** to upload the image. The uploaded image is displayed in the object list.

Step 7 In the navigation pane, choose **Data Processing** > **Image Processing**.

Step 8 Click **Create** to go to the style editing page shown in **Figure 10-22**.

Figure 10-22 Creating a style



You can use either GUI or code to configure the effect, scale mode, watermark, and output mode of an image. Learn more

Style Name: style-c820

Sample Image Path:

Default path: e-share/example.jpg in region CN North-Beijing1. You can change the default path as needed.

Edit Mode: GUI | Code

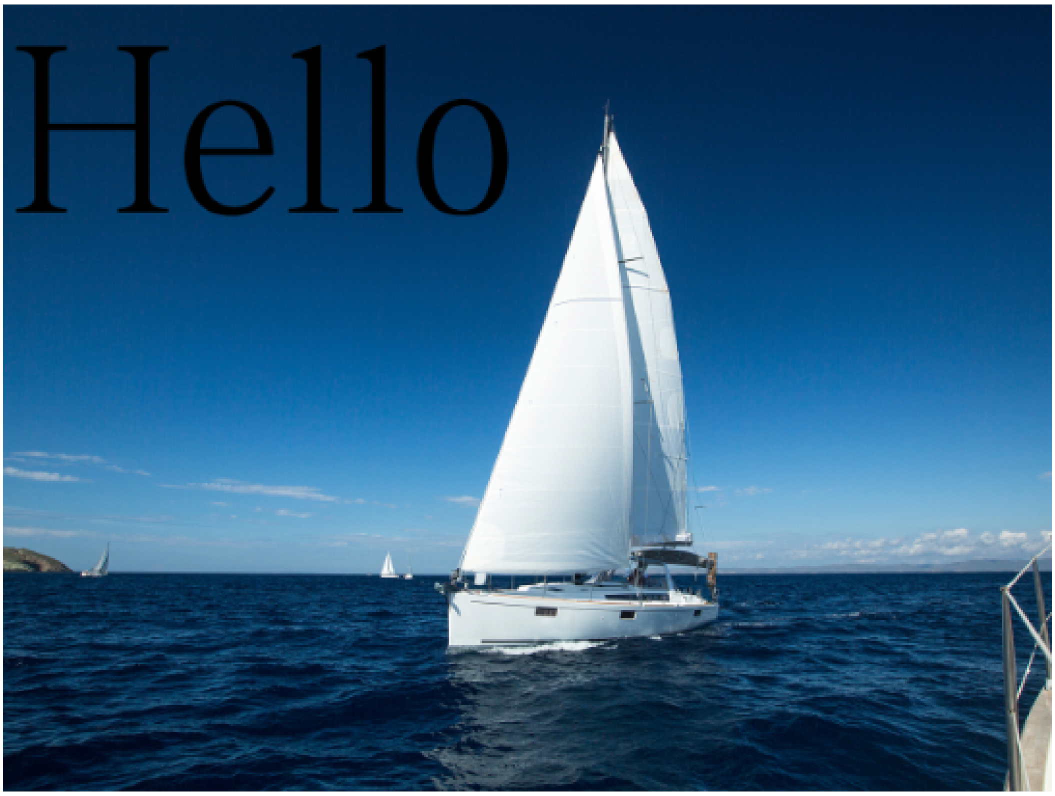
Step 9 Set **Style Name** to **obs-111** and choose **GUI** for **Edit Mode**.

Step 10 Expand **Watermark**.

1. Choose **Text watermark** for **Watermark Type**.
2. Specify **Content** to **Hello**.
3. Choose **FZShuSong** for **Text Font**.
4. Set **Text Size** to **600**.
5. Keep the default value **100** for **Watermark Transparency**.
6. Select the top left sign (↖) for **Watermark Position**.
7. Keep **Text Shadow** (○) disabled.
8. Keep the default value **10** for both **Vertical Margin** and **Horizontal Margin**.

The style effect will be displayed on the right in real time. Figure 10-23 shows the final style effect.

Figure 10-23 Effect of style **obs-111**

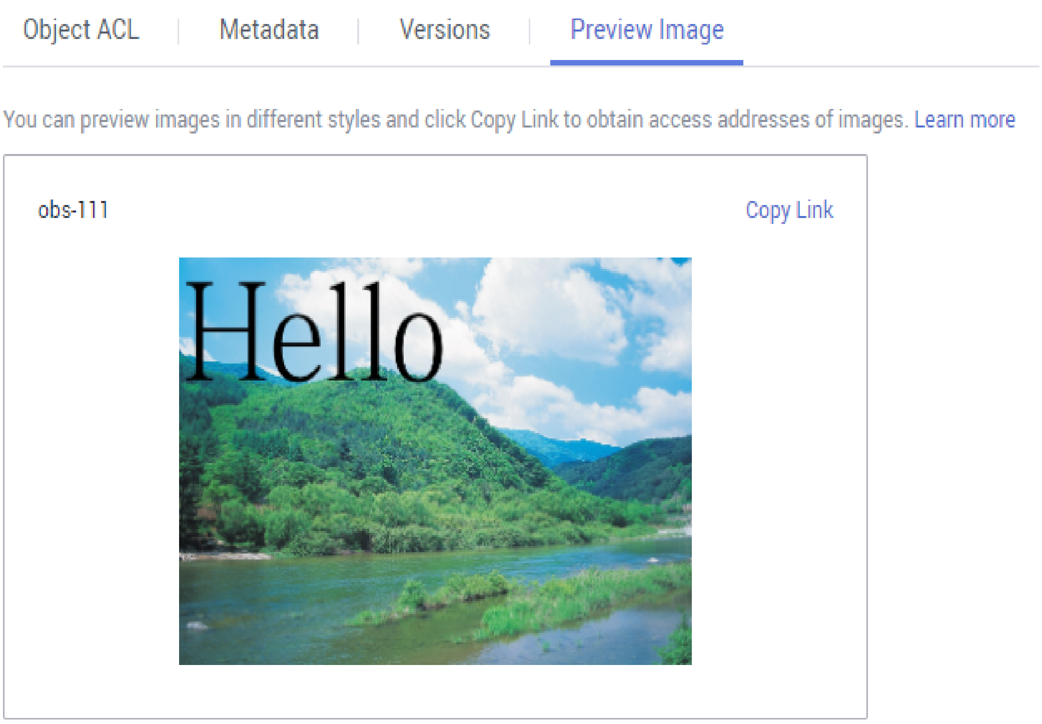


Step 11 Click **OK** to save the style. The new style **obs-111** will be displayed in the style list.

Step 12 In the navigation pane, choose **Objects**. Click **mountain.jpg** in the object list to go to the file details page.

Step 13 Click the **Preview Image** tab to preview the effect of the image with style **obs-111** applied.

Figure 10-24 Preview Image



You can preview images in different styles and click Copy Link to obtain access addresses of images. Learn more

Step 14 In the image style card, click **Copy Link**. When **Copied successfully.** is prompted, you will have the address for accessing the image with the style applied.

----End

10.1.13 Converting Formats and Interlacing Images

Converting Formats

You can use the GUI or code mode on OBS Console or make an API call to convert images to different formats:

- Supported original formats: JPG, JPEG, PNG, BMP, WebP, GIF, and TIFF
- Supported target formats: JPG, PNG, BMP, and WebP (described in **Table 10-20**)

This operation is represented by **format**.

Table 10-20 Format conversion

Parameter	Value Description	Code Example
jpg	The image is saved in JPG format. If the original image is in vector format such as WebP, BMP, or PNG, the transparent part will be padded to white.	image/format,jpg

Parameter	Value Description	Code Example
webp	The image is saved in WebP format.	image/format,webp
bmp	The image is saved in BMP format.	image/format,bmp
png	The image is saved in PNG format.	image/format,png

Example of Converting Formats

- Save a JPG image as a PNG image.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/format,png

Interlaced Image Loading

You can edit code on OBS Console or make an API call to load images in an interlaced way.

With format conversion, images are generated in baseline JPEG format by default. If you want to generate an image in progressive JPEG, use the **interlace** parameter. **Table 10-21** describes the parameters.

- Display of baseline JPG images: from top to bottom
- Display of progressive JPEG images: from blurry to clearer and clearer

This operation is represented by **interlace**.

Table 10-21 Interlaced loading

Parameter	Value Description	Code Example
value	The value can be **0** or **1**. **0**: The output is a JPG image that is displayed from top to bottom. **1**: The output is a JPEG image in an interlaced display.	image/format,jpg/interlace,1

Examples of Interlaced Image Loading

- Create a JPG image in an interlaced display.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/format,jpg/interlace,1

- Create a JPG image that is displayed from top to bottom.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/format,jpg/interlace,0

10.1.14 Changing Image Quality

You can edit code on OBS Console or make an API call to change the quality of an image.

To save space, you can compress JPG images. **Table 10-22** describes the parameters.

This operation is represented by **quality**.

Table 10-22 Image compression parameters

Parameter	Value Description	Code Example
q	Relative quality of the image. The image is compressed to q% of the original. The value ranges from **1** to **100**. Formula for compression: Target quality = Original quality × q% For example, if the original image quality is 100% and the relative quality is 80%, then the target image quality is 80%. If the original quality is 80% and the relative quality is 80%, then the target quality is 64%.	image/resize,w_100,h_100/quality,q_80

Parameter	Value Description	Code Example
Q	Absolute quality of the image. The image is compressed into Q%. Q is irrelevant to and does not depend on the original image. The value ranges from **1** to **100**. Formulas for compression: ● Original quality > Q%: target quality = Q% ● Original quality = Q%: target quality = original quality = Q% ● Original quality < Q%: target quality = original quality For example, if the original image quality is 100% and the absolute quality is 80%, then the target image quality is 80%. If the original quality is 70% and the absolute quality is 80%, then the target quality is 70%.	

NOTE

- **q** is valid only for JPG images.
- If both **q** and **Q** are used, the output is based on **Q**.
- **q** and **Q** are only valid for JPG images. For images in other formats, **q** and **Q** bring no effect and cause no impact.

Examples

- Resize the image by setting the height and width both to **100** and generate a JPG image with a relative quality of 80%.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_100,h_100/quality,q_80

- Resize the image by setting the height and width both to **100** and generate a JPG image with an absolute quality of 80%.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_100,h_100/quality,Q_80

10.1.15 Slimming Images

You can edit code on OBS Console or make an API call to slim images.

Slimming is a simplified image compression, which generates images with an absolute quality of 75%. You do not need to configure any parameter. You can slim images just by running a command.

This operation is represented by **imageslim**.

NOTE

- Only JPG images can be slimmed.
- It is recommended that you slim images after other image processing operations.

Example

- Resize an image by setting the width and height both to **100** and then slim the resized image.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_100,h_100/imageslim

10.1.16 Image Processing Persistency

Image processing persistency allows the images after processing to be synchronously stored in a specified OBS bucket, so that you can directly access the stored images to save time.

You can use image processing persistency only by making API calls. In an image processing request, the persistency should be configured in the *Parameter name*=*Parameter value* format. **Table 10-23** describes the parameters.

Table 10-23 Image processing persistency parameters

Parameter	Value	Description
x-image-save-object	*objectName*	Mandatory It specifies the name of the processed image that will be stored in the bucket. An object name: ● cannot contain the following special characters: \ : * ? " < > | ● must be 1 to 1023 characters long.
x-image-save-bucket	*bucketName*	Optional It specifies the bucket that will store the processed image. If this parameter is not specified, the processed image will be saved in the bucket where the original image is processed. The bucket must be an existing one in OBS and its name must be 1 to 64 characters long.

Java Sample Code

```
ObsClient obsClient = null;
String endPoint = "obs-endpoint";  // Current region
// Hard-coded or plaintext AK and SK are risky. For security purposes, encrypt your AK and SK and store
```

```
them in the configuration file or environment variables.
// In this example, the AK and SK are stored in environment variables for identity authentication. Before
running this example, configure environment variables ACCESS_KEY_ID and SECRET_ACCESS_KEY.
// Obtain an AK and SK pair on the management console. For details, see https://
support.huaweicloud.com/intl/en-us/usermanual-ca/ca_01_0003.html.
String ak = System.getenv("ACCESS_KEY_ID");
String sk = System.getenv("SECRET_ACCESS_KEY");
try {
    ObsConfiguration config = new ObsConfiguration();
    config.setEndPoint(endPoint);
    obsClient = new ObsClient(ak,sk ,config);
    TemporarySignatureRequest request = new TemporarySignatureRequest();
    request.setObjectKey("test.jpeg");  // Original object name before processing
    Map<String, Object> queryParams = new HashMap<>();
    queryParams.put("x-image-process", "image/resize,w_100");
    String objectName = "your saves objectName";  // Name of the processed object
    // Optional parameters
    String bucketName = "your saves Bucket";  // Bucket that stores the processed object
    queryParams.put("x-image-save-object", ServiceUtils.toBase64(objectName.getBytes("UTF-8")));
    queryParams.put("x-image-save-bucket", ServiceUtils.toBase64(bucketName.getBytes("UTF-8")));
    request.setQueryParams(queryParams);
    request.setBucketName("your bucket");  // Bucket that stores the original object
    TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
    //URL to access
    response.getSignedUrl();
 } catch (Exception e) {
...//Handle exceptions.
 } finally {
     if (obsClient != null) {
         obsClient.close();
       }
 }
```

Python Sample Code

```
from obs import ObsClient
import os
import traceback
import requests

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console. For details, see https://
support.huaweicloud.com/intl/en-us/usermanual-ca/ca_01_0003.html.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
security_token = os.getenv("SecurityToken")
# Set server to the endpoint corresponding to the bucket. CN-Hong Kong (ap-southeast-1) is used here as
an example. Replace it with the one in use.
server = "https://obs.ap-southeast-1.myhuaweicloud.com"
# Create an ObsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Generate a signed URL for image persistency.
    # Name of the bucket that stores the original object
    bucketName = 'originBucketName';
    # Original object name
    objectKey = 'test.png';

    # Name of the object after processing
    targetObjectName ="save.png"
    # (Optional) Name of the bucket that stores the new object
    targetBucketName ="saveBucketName"
    queryParams={}
    queryParams["x-image-process"]="image/resize,w_100"
    queryParams["x-image-save-object"]=base64.b64encode(targetObjectName .encode("utf-8")).decode()
    # Optional parameter
    queryParams["x-image-save-bucket"]=base64.b64encode(targetBucketName .encode("utf-8")).decode()

    res = obsClient.createSignedUrl(method='GET', bucketName=bucketName, objectKey=objectKey, 
queryParams=queryParams, expires=3600)
    print('signedUrl:', res.signedUrl)
    print('actualSignedRequestHeaders:', res.actualSignedRequestHeaders)
    // Make a GET request for image persistency.
    r = requests.get(resp.signedUrl)
    print(r)
except:
    print(traceback.format_exc())
```

Node.js Sample Code

```
// Import the OBS library.
const ObsClient = require('esdk-obs-nodejs');
const https = require('https');
const http = require('http');
const urlLib = require('url');

// Hard-coded or plaintext AK and SK are risky. For security purposes, encrypt your AK and SK and store 
them in the configuration file or environment variables.
// In this example, the AK and SK are stored in environment variables for identity authentication. Before 
running this example, configure environment variables ACCESS_KEY_ID and SECRET_ACCESS_KEY.
// Obtain an AK and SK pair on the management console. For details, see https://
support.huaweicloud.com/intl/en-us/usermanual-ca/ca_01_0003.html.
const ak = process.env.ACCESS_KEY_ID;
const sk = process.env.SECRET_ACCESS_KEY;
const server = "obs-endpoint";  // Current region

// Create an ObsClient instance.
const obsClient = new ObsClient({
    access_key_id: ak,
    secret_access_key: sk,
    server: server
});

// Name of the bucket that stores the original object
const bucketName = 'originBucketName';
// Original object name
const objectKey = 'test.png';
const method = 'GET';

// Name of the object after processing
const targetObjectName = "save.png";
// (Optional) Name of the bucket that stores the new object
const targetBucketName = 'saveBucketName';

const queryParams = {
    "x-image-process": "image/resize,w_100",
    "x-image-save-object": Buffer.from(targetObjectName, 'utf8').toString('base64'),
    // Optional parameter
    "x-image-save-bucket": Buffer.from(targetBucketName, 'utf8').toString('base64')
}

const res = obsClient.createSignedUrlSync({
    Method: method,
    Bucket: bucketName,
    Key: objectKey,
    QueryParams: queryParams
});

// Make a GET request for image persistency.
const url = urlLib.parse(res.SignedUrl);
const request = server.startsWith('http://') ? http : https;
const req = request.request({
    method: method,
    host: url.hostname,
    port: url.port,
    path: url.path,
    rejectUnauthorized: false,
    headers: res.ActualSignedRequestHeaders || {}
});
```

NOTE

- Object and bucket names must be URL-safe Base64 encoded in the **encodedObject = url_safe_base64_encode(name)** format. For example, object **panda.png** will be encoded as **cGFuZGEucG5n**. After Base64 encoding, if the name contains plus signs (+) and slashes (/), replace them with hyphens (-) and underscores (_), respectively.
- If a signature matching error is reported, check whether the AK and SK pair is correct and whether the accessed URL is the same as the signature URL generated by the code.
- Currently, image persistency with the range header is not supported.

10.2 Online Decompression

You can compress files into a ZIP package, upload it to OBS for auto decompression online, and store it in a specified directory.

Uploading a large number of small files at a time by using a ZIP package saves time and efforts. For batch upload constraints, see **OBS Batch Upload**.

If the package you uploaded matches the configured decompression policy, it will be automatically decompressed upon upload. A decompression policy does not apply to the ZIP packages that already exist in OBS before the policy is created.

You can create online decompression policies or replicate existing policies from another bucket.

Prerequisites

You have been assigned the **Tenant Administrator** role.

Constraints

Permissions

- To configure, obtain, or delete online decompression policies, you must have the **Tenant Administrator** permission assigned by using IAM.
- Online decompression requires that you create an IAM agency to delegate OBS to access data in the bucket. The permissions the agency should assign to OBS include **obs:object:PutObject**, **obs:object:GetObject**, and **obs:object:AbortMultipartUpload**.

Regions

- Online decompression is only available in some regions. For details, see **Function Overview**.

Package and file size

- A single ZIP package cannot exceed **1 GB** in size.
- A single ZIP package can contain a maximum of 65,536 files.
- A single decompressed file cannot exceed 40 GB in size.

Time

- Decompressing a ZIP package takes 10 minutes at most.

Functions

- Currently, only ZIP packages are supported.
- ZIP package names cannot contain Chinese punctuation marks, special characters, or special codes.
- To decompress the ZIP package that contains other ZIP packages, the event type of the online decompression policy must be set to **ObjectCreated:*** or **ObjectCreated:CompleteMultipartUpload**.
- Currently, no notification will be sent to users after decompression tasks are complete.
- Currently, only deflate-compressed ZIP packages can be decompressed. Deflate is different from the compression algorithms (such as Store and Normal) provided by WinRAR.
- The total length of the decompression path plus the name of any decompressed file cannot exceed 512 characters.
- Currently, encrypted ZIP packages cannot be decompressed.
- Federated users cannot use the online decompression function. For more information about federated users, see **What Are the Relationships Between a Huawei Cloud Account, HUAWEI ID, IAM User, and Federated User?** and **Identity Providers**.

Precautions

- You are advised to set a precise prefix for a decompression policy. In the same bucket, trigger conditions (including events, prefixes, or suffixes) of different decompression policies cannot overlap with each other.

  For example, there are two decompression policies **event-0001** and **event-0002** in a bucket. If the prefix of **event-0001** is **aa**, the prefix of **event-0002** cannot be **aaaa**, because **aa** is contained in **aaaa**.
- If the prefix is left blank, the decompression policy applies to all the ZIP packages in the bucket by default. This may trigger cyclic decompression if a package contains other packages.

  For example, package **AA.zip** contains another package **BB.zip**. If the prefix is left blank, the system continues to decompress **BB.zip** after decompressing **AA.zip**. This issue will not happen if a prefix is set in the policy.
- You must set a directory for storing the decompressed files. If the directory is not set, decompressed files will be stored in the home directory of the current bucket.
- You are advised to encode file or folder names using UTF-8. Otherwise, names of decompressed files or folders may contain garbled characters, or the decompression may be interrupted.
- If you want ZIP packages in the Archive storage class to be automatically decompressed upon upload, ensure that Direct Reading has been enabled for the bucket. ZIP packages in the Deep Archive storage class will not be automatically decompressed after they are uploaded to an OBS bucket.
- A ZIP package decompression will fail if it takes more than 10 minutes.

Creating an Online Decompression Policy

You can use OBS Console or APIs to create an online decompression policy.

Using OBS Console

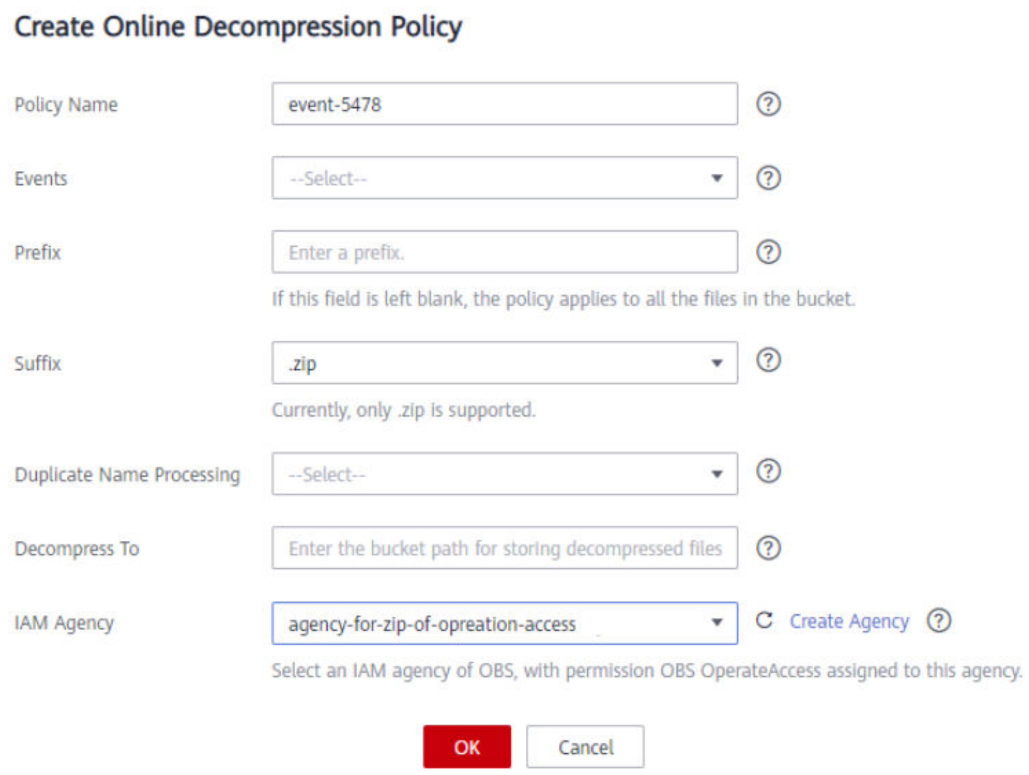
Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Data Processing** > **Online Decompression**. The **Online Decompression** page is displayed.

Step 4 Click **Create**.

Figure 10-25 Creating an online decompression policy



Create Online Decompression Policy

Policy Name	event-5478
Events	--Select--
Prefix	Enter a prefix. If this field is left blank, the policy applies to all the files in the bucket.
Suffix	.zip Currently, only .zip is supported.
Duplicate Name Processing	--Select--
Decompress To	Enter the bucket path for storing decompressed files
IAM Agency	agency-for-zip-of-opreation-access Select an IAM agency of OBS, with permission OBS OperateAccess assigned to this agency.

Step 5 Configure the online decompression policy. **Table 10-24** describes the related parameters.

Table 10-24 Parameter description

Parameter	Description
Policy Name	Enter a policy name that is easy to remember. The value can contain 1 to 256 characters, and only uppercase letters, lowercase letters, digits, underscores (_), and hyphens (-) are allowed, for example, **event_0001**.
Events	Events that you want to trigger the online decompression policy for. Currently, the following event types are supported: ● **ObjectCreated**: all object creation operations, including PUT, POST, COPY, and part assembling ● **Put**: object upload using PUT ● **Post**: object upload using POST ● **Copy**: object copying using COPY ● **CompleteMultipartUpload**: assembling of parts in a multipart upload **NOTE** To decompress the ZIP package that contains other ZIP packages, set the event type to **ObjectCreated** or **CompleteMultipartUpload**.
Prefix	Optional. If this parameter is configured, the decompression policy applies to the packages whose name contains this prefix. The prefix cannot start with a slash (/) or contain double slashes (//), or contain special characters (\ : * ? " < > |). The total length of the prefix and suffix cannot exceed 512 characters. ● With this parameter configured, ZIP packages whose name contains the specified prefix will trigger online decompression. ● With this parameter left blank, the decompression policy applies to all the uploaded ZIP packages. **CAUTION** – You are advised to configure a prefix. Otherwise, cyclic decompression may occur if a package contains other packages. – The configured prefix must contain all levels of the directory for storing the object. For example, there is an object **example123** that is stored under **bucket/file/example123**. If you want **example** to be the prefix, set the prefix to **file/example**.
Suffix	If this parameter is specified, the decompression policy applies to the packages whose name contains this suffix. Currently, **.zip** is the default and only value.

Parameter	Description
Duplicate Name Processing	It specifies how the decompressed objects are processed if they have the same names as the existing objects in the bucket. ● **Do not decompress**: Retains the existing objects, and does not decompress the objects with the same name. ● **Rename the file**: Renames the decompressed objects with the CRC32 value. ● **Overwrite**: Overwrites the existing objects in the bucket.
Decompress To	Optional, this parameter specifies the path for storing decompressed files. It cannot contain special characters (\:*?\"<>|), start or end with a period (.), or contain two or more consecutive slashes (/). The value can contain 0 to 1,023 characters. ● With this parameter configured, the path must end with a slash (/). After a ZIP package is decompressed, the decompressed files are stored in the folder with the same name as the path. If there is no such a folder in the bucket, OBS automatically creates one for storing the files. ● With this parameter left blank, decompressed objects are stored in the home directory of the bucket.
IAM Agency	Select an IAM agency of OBS, with the **OBS OperateAccess** permission assigned. If no such agency is available, create one.

Step 6 Click **OK**. The online decompression policy is created.

----End

Using the API

Setting an Online Decompression Policy

Replicating Online Decompression Policies

You can use OBS Console to replicate online decompression policies.

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Data Processing** > **Online Decompression**. The **Online Decompression** page is displayed.

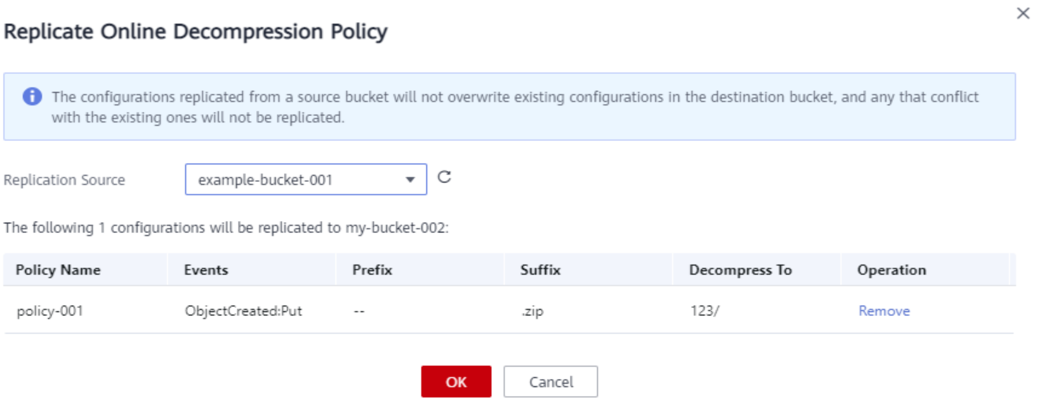
Step 4 Click **Replicate**.

Step 5 Select a replication source, which is the bucket whose online decompression policies you want to replicate.

NOTE

- The online decompression policies replicated from a source bucket will not overwrite existing policies in the destination bucket, and any that conflict with the existing ones will not be replicated.
- Both source and destination buckets must be of the 3.0 version.
- You can remove the policies that you do not want to replicate.
- There can be 10 online decompression policies at most in a bucket. If the number of policies you will replicate plus the number of existing policies in the destination bucket exceeds 10, the replication will fail. Before replicating the policies, delete some if necessary.

Figure 10-26 Replicating online decompression policies



Replicate Online Decompression Policy

The configurations replicated from a source bucket will not overwrite existing configurations in the destination bucket, and any that conflict with the existing ones will not be replicated.

Replication Source: example-bucket-001

The following 1 configurations will be replicated to my-bucket-002:

Policy Name	Events	Prefix	Suffix	Decompress To	Operation
policy-001	ObjectCreated:Put	--	.zip	123/	Remove

Step 6 Click **OK** to replicate the online decompression policies to the destination bucket.

----End

11 Monitoring and Logging

11.1 Using Cloud Eye to Monitor OBS Buckets

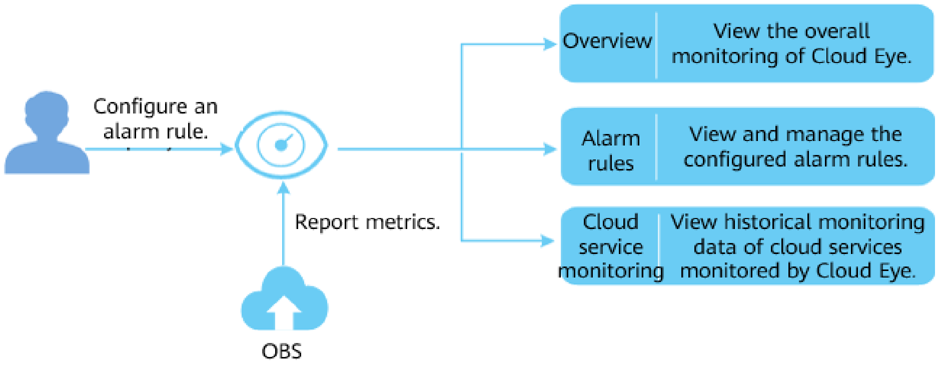
11.1.1 Monitoring OBS

Scenarios

In the use of OBS, you may send PUT and GET requests that generate upload and download traffic, or receive error responses from the server. Huawei Cloud provides Cloud Eye to help you monitor OBS and better understand your bucket statuses. To keep informed of the requests, traffic, and error responses in a timely manner, you can use Cloud Eye to perform automatic and real-time monitoring over your buckets.

You do not need to separately subscribe to Cloud Eye. It starts automatically once you create a resource (a bucket, for example) in OBS. For more information about Cloud Eye, see **Introduction to Cloud Eye**.

Figure 11-1 Cloud Eye monitoring


Configure an alarm rule.
Report metrics.
OBS
Overview
View the overall monitoring of Cloud Eye.
Alarm rules
View and manage the configured alarm rules.
Cloud service monitoring
View historical monitoring data of cloud services monitored by Cloud Eye.

Setting Alarm Rules

In addition to automatic and real-time monitoring, you can configure alarm rules in Cloud Eye to receive alarm notifications when there are exceptions.

For details about how to configure alarm rules for Cloud Eye monitoring over OBS, see **Creating an Alarm Rule**.

On Cloud Eye, you can configure alarm rules for events. When specified events happen, you will receive alarm notifications. For details, see **Creating an Alarm Rule to Monitor an Event**.

Viewing OBS Monitoring Metrics

Cloud Eye monitors **OBS metrics** in real time. You can view monitoring details of each metric on the console of Cloud Eye.

For details about how to view OBS monitoring metrics, see **Querying Metrics of a Cloud Service**.

Cloud Eye monitors **OBS events** in real time. You can view the monitoring data on the Cloud Eye console. For details, see **Viewing Event Monitoring Data**.

11.1.2 OBS Monitoring Metrics

Functions

This section defines the namespace, list, and dimensions of monitoring metrics reported by OBS to Cloud Eye. You can use the management console or **API** provided by Cloud Eye to search for monitoring metrics and alarms generated by OBS.

Namespace

SYS.OBS

Monitoring Metrics

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
download_bytes	Bytes Downloaded	Response bytes of all download requests made to all buckets in a region, including bytes in HTTP entity bodies Unit: byte	≥ 0 bytes	Bucket	5 min

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
upload_bytes	Bytes Uploaded	Bytes of all upload requests made to all buckets in a region, including bytes in HTTP entity bodies Unit: byte	≥ 0 bytes	Bucket	5 min
get_request_count	GET Requests	Number of GET, HEAD, or OPTIONS requests made to all buckets and objects in the buckets of a region Unit: count	≥ 0 counts	Bucket	5 min
put_request_count	PUT Requests	Number of PUT, POST, and DELETE requests made to all buckets and objects in the buckets of a region Unit: count	≥ 0 counts	Bucket	5 min
first_byte_latency	First Byte Download Delay	Average time from receiving a GET, HEAD, or OPTIONS request to the time that the system starts to respond in a measurement period Unit: ms	≥ 0 ms	Bucket	5 min
request_count_4xx	4*xx* Errors	Times that the server responds to requests whose error codes are 4xx Unit: count	≥ 0 counts	Bucket	5 min
request_count_5xx	5*xx* Errors	Times that the server responds to requests whose error codes are 5xx Unit: count	≥ 0 counts	Bucket	5 min
get_mirror_range_bandwidths	Range Retrieval Bandwidth (Mirroring-based Back to Source)	Bandwidth for retrieving object ranges with mirroring-based back to source Unit: byte/s	≥ 0 bytes/s	User	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
get_mirror_fetch_bandwidths	Fetch Bandwidth (Mirroring-based Back to Source)	Bandwidth for fetches triggered by mirroring-based back to source Unit: byte/s	≥ 0 bytes/s	User	1 minute
get_mirror_fetch_task_status	Fetch Status (Mirroring-based Back to Source)	Status of asynchronous fetches triggered by mirroring-based back to source. A value of 1 indicates that asynchronous fetches can be triggered, while a value less than 1 indicates that asynchronous fetches cannot be triggered.	≤ 1	User	1 minute

Monitoring Metrics

Table 11-1 OBS metrics (for requests)

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
get_request_count	GET Requests	Number of GET requests made to all buckets and objects in the buckets of a region. Unit: count	≥ 0 counts	Bucket	1 minute
put_request_count	PUT Requests	Number of PUT requests made to all buckets and objects in the buckets of a region. Unit: count	≥ 0 counts	Bucket	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
first_byte_latency	First Byte Download Delay	Average time from receiving a GET request to the time that the system starts to respond in a measurement period. Unit: ms	≥ 0 ms	Bucket	1 minute
request_count_4xx	4XX Status Codes	Number of requests whose status code returned by the server is 4*xx*. Unit: count	≥ 0 counts	User Bucket	1 minute
request_count_5xx	5XX Status Codes	Number of requests whose status code returned by the server is 5*xx*. Unit: count	≥ 0 counts	User Bucket	1 minute
total_request_latency	Average Request Latency	Average time from receiving a request to the time that the system response ends in a measurement period. Unit: ms	≥ 0 ms	User Bucket	1 minute
request_count_per_second	Total TPS	Average number of requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket	1 minute
request_count_get_per_second	GET Request TPS	Average number of GET requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket	1 minute
request_count_put_per_second	PUT Request TPS	Average number of PUT requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
request_count_delete_per_second	DELETE Request TPS	Average number of DELETE requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket	1 minute
request_success_rate	Request Success Rate	Used to measure the availability of the storage service system. It refers to the percentage of non-server error requests (with status code 5*xx* returned) in the total request count. It is calculated as follows: (1 – 5*xx* requests/Total requests) x 100% Unit: %	≥ 0, ≤ 100	User Bucket	1 minute
effective_request_rate	Valid request rate	Validity of client requests. Percentage of the valid requests in the total requests. It is calculated as follows: (Number of client requests whose returned status code is 2*xx* or 3*xx*/Total number of requests) x 100% Unit: %	≥ 0, ≤ 100	User Bucket	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
request_break_rate	Request interruption rate	Percentage of the number of requests interrupted by a client to the total number of requests. It is calculated as follows: (Number of requests interrupted by a client/ Total number of requests) x 100% Unit: %	≥ 0, ≤ 100	User Bucket	1 minute
request_count_monitor_2XX	2xx Status Codes	Count of server responses to requests whose status codes are 2*xx*. Unit: count	≥ 0 counts	User Bucket	1 minute
request_count_monitor_3XX	3xx Status Codes	Count of server responses to requests whose status codes are 3*xx*. Unit: count	≥ 0 counts	User Bucket	1 minute
download_bytes	Total Download Bandwidth	Total size of objects downloaded per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket	1 minute
download_bytes_extranet	Download Bandwidth (Internet)	Total size of objects downloaded over the Internet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket	1 minute
download_bytes_intranet	Download Bandwidth (Intranet)	Total size of objects downloaded over the Intranet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
upload_bytes	Total Upload Bandwidth	Total size of objects uploaded per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket	1 minute
upload_bytes_extranet	Upload Bandwidth (Internet)	Total size of objects uploaded over the Internet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket	1 minute
upload_bytes_intranet	Upload Bandwidth (Intranet)	Total size of objects uploaded over the Intranet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket	1 minute
download_traffic	Total Download Traffic	Total size of objects downloaded in a measurement period. Unit: byte	≥ 0 bytes/s	User Bucket	1 minute
download_traffic_extranet	Download Traffic (Internet)	Total size of objects downloaded over the Internet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket	1 minute
download_traffic_intranet	Download Traffic (Intranet)	Total size of objects downloaded over the Intranet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket	1 minute
upload_traffic	Total Upload Traffic	Total size of objects uploaded in a measurement period. Unit: byte	≥ 0 bytes	User Bucket	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
upload_traffic_extranet	Upload Traffic (Internet)	Total size of objects uploaded over the Internet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket	1 minute
upload_traffic_intranet	Upload Traffic (Intranet)	Total size of objects uploaded over the Intranet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket	1 minute
get_mirror_range_bandwidths	Range Retrieval Bandwidth (Mirroring-based Back to Source)	Bandwidth for retrieving object ranges with mirroring-based back to source Unit: byte/s	≥ 0 bytes/s	User	1 minute
get_mirror_fetch_bandwidths	Fetch Bandwidth (Mirroring-based Back to Source)	Bandwidth for fetches triggered by mirroring-based back to source Unit: byte/s	≥ 0 bytes/s	User	1 minute
get_mirror_fetch_task_status	Fetch Status (Mirroring-based Back to Source)	Status of asynchronous fetches triggered by mirroring-based back to source. A value of 1 indicates that asynchronous fetches can be triggered, while a value less than 1 indicates that asynchronous fetches cannot be triggered.	≤ 1	User	1 minute

Table 11-2 OBS metrics (for storage)

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
capacity_total	Total Used Storage Space	Measures the storage space occupied by all data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_standard	Used Space - Standard Storage	Measures the storage space occupied by Standard data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_infrequent_access	Used Space - Infrequent Access Storage	Measures the storage space occupied by Infrequent Access data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_archive	Used Space - Archive Storage	Measures the storage space occupied by Archive data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_standard_single_az	Standard Storage (Single-AZ)	Space occupied by Standard data stored in a single AZ Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_standard_multi_az	Standard Storage (Multi-AZ)	Space occupied by Standard data stored in multiple AZs Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_infrequent_access_single_az	Infrequent Access Storage (Single-AZ)	Space occupied by Infrequent Access data stored in a single AZ Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_infrequent_access_multi_az	Infrequent Access Storage (Multi-AZ)	Space occupied by Infrequent Access data stored in multiple AZs Unit: byte	≥ 0 bytes	User Bucket	30 minutes

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
object_num_all	Total Number of Objects	Total number of objects (including folders and all file versions) stored in all storage classes Unit: count	≥ 0	User Bucket	30 minutes
object_num_standard_total	Number of Objects - Standard Storage	Total number of objects (including folders and all file versions) stored in the Standard storage class Unit: count	≥ 0	User Bucket	30 minutes
object_num_infrequent_access_total	Number of Objects - Infrequent Access Storage	Total number of objects (including folders and all file versions) stored in the Infrequent Access storage class Unit: count	≥ 0	User Bucket	30 minutes
object_num_archive_total	Number of Objects - Archive Storage	Total number of objects (including folders and all file versions) stored in the Archive storage class Unit: count	≥ 0	User Bucket	30 minutes
object_num_standard_single_az	Standard Objects (Single-AZ)	Total number of Standard objects (including folders and all object versions) stored in a single AZ Unit: count	≥ 0	User Bucket	30 minutes
object_num_standard_multi_az	Standard Objects (Multi-AZ)	Total number of Standard objects (including folders and all object versions) stored in multiple AZs Unit: count	≥ 0	User Bucket	30 minutes
object_num_infrequent_access_single_az	Infrequent Access Objects (Single-AZ)	Total number of Infrequent Access objects (including folders and all object versions) stored in a single AZ Unit: count	≥ 0	User Bucket	30 minutes

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
object_num_infrequent_access_multi_az	Infrequent Access Objects (Multi-AZ)	Total number of Infrequent Access objects (including folders and all object versions) stored in multiple AZs Unit: count	≥ 0	User Bucket	30 minutes

Event Monitoring

Table 11-3 OBS events that can be monitored

Event Source	Event Name	Event ID	Event Severity
OBS	Deleting a bucket	deleteBucket	Major
	Deleting a bucket policy	deleteBucketPolicy	Major
	Configuring a bucket ACL	setBucketAcl	Minor
	Configuring a bucket policy	setBucketPolicy	Minor

Dimensions

Table 11-4 Dimensions

Key	Value
Bucket_Name	Bucket dimension. The value is the bucket name.

11.2 Using CTS to Audit OBS

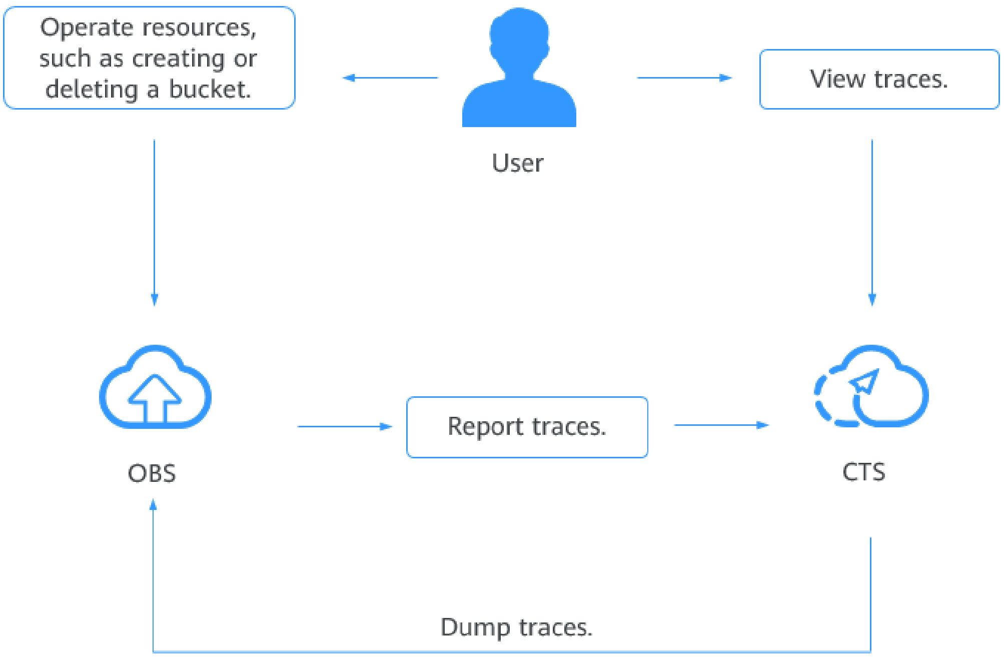
Cloud Trace Service (CTS) records operations on cloud resources in your account. You can use the logs to perform security analysis, track resource changes, audit compliance, and locate faults.

After you enable CTS and configure a tracker, CTS can record management and data traces of OBS for auditing.

For details about how to enable and configure CTS, see **Enabling CTS**.

For details about OBS management and data traces that can be tracked by CTS, see **Cloud Trace Service**.

Figure 11-2 CTS



Procedure

Step 1 Log in to the management console.

Step 2 In the upper left corner of the top navigation menu, click 📍 to select a region.

Step 3 Choose **Service List** > **Management & Governance** > **Cloud Trace Service**. The **Trace List** page is displayed.

Step 4 Configure the cloud audit for OBS by referring to **Configuring a Tracker** in the *Cloud Trace Service User Guide*.

----End

Table 11-5 OBS management operations logged by CTS

Tracker Type	Operation	Resource	Trace Name
Management	Deleting a bucket	bucket	deleteBucket
Management	Deleting the CORS configuration of a bucket	bucket	deleteBucketCors

Tracker Type	Operation	Resource	Trace Name
Management	Deleting the custom domain name configuration	bucket	deleteBucketCustom-domain
Management	Deleting the lifecycle configuration of a bucket	bucket	deleteBucketLifecycle
Management	Deleting a bucket policy	bucket	deleteBucketPolicy
Management	Deleting the cross-region replication configuration of a bucket	bucket	deleteBucketReplication
Management	Deleting the tag configuration of a bucket	bucket	deleteBucketTagging
Management	Deleting the static website hosting configuration of a bucket	bucket	deleteBucketWebsite
Management	Creating a bucket	bucket	createBucket
Management	Configuring the bucket ACL	bucket	setBucketAcl
Management	Configuring the CORS rule for a bucket	bucket	setBucketCors
Management	Setting the custom domain name for a bucket	bucket	setBucketCustomdomain
Management	Configuring the bucket lifecycle rules	bucket	setBucketLifecycle
Management	Configuring the bucket logging function	bucket	setBucketLogging
Management	Configuring the event notification function for buckets	bucket	setBucketNotification

Tracker Type	Operation	Resource	Trace Name
Management	Configuring the bucket policy	bucket	setBucketPolicy
Management	Configuring the bucket quota	bucket	setBucketQuota
Management	Configuring the cross-region replication function for buckets	bucket	setBucketReplication
Management	Configuring the bucket storage class	bucket	setBucketStorageclass
Management	Configuring the bucket tag	bucket	setBucketTagging
Management	Configuring the versioning function for buckets	bucket	setBucketVersioning
Management	Configuring the static domain name for buckets	bucket	setBucketWebsite
Management	Configuring server-side encryption for a bucket	bucket	setBucketEncryption
Management	Deleting the server-side encryption configuration of a bucket	bucket	deleteBucketEncryption

Table 11-6 OBS data operations logged by CTS

Tracker Type	Operation	Resource	Trace Name
Data_Read	Downloading an object	object	GET.OBJECT
Data_Read	Querying the object ACL	object	GET.OBJECT.ACL
Data_Read	Querying the bucket website configuration	object	GET.OBJECT.WEBSITE

Tracker Type	Operation	Resource	Trace Name
Data_Read	Accessing an object through the website	object	HEAD.OBJECT.WEBSITE
Data_Read	Querying the object metadata	object	HEAD.OBJECT
Data_Read	Listing part data	object	LIST.OBJECT.UPLOAD
Data_Write	Deleting an object	object	DELETE.OBJECT
Data_Write	Canceling a part	object	DELETE.UPLOAD
Data_Write	Queries the cross-domain requests for objects	object	OPTIONS.OBJECT
Data_Write	Uploading an object	object	POST.OBJECT
Data_Write	Deleting objects in batches	object	POST.OBJECT.MULTIDELETE
Data_Write	Restoring Archive objects	object	POST.OBJECT.RESTORE
Data_Write	Merging parts	object	POST.UPLOAD.COMPLETE
Data_Write	Initializing multipart tasks	object	POST.UPLOAD.INIT
Data_Write	Uploading an object	object	PUT.OBJECT
Data_Write	Configuring the object ACL	object	PUT.OBJECT.ACL
Data_Write	Copying an object	object	PUT.OBJECT.COPY
Data_Write	Configuring the object storage class	object	PUT.OBJECT.STORAGECLASS
Data_Write	Uploading a part	object	PUT.PART
Data_Write	Copying a part	object	PUT.PART.COPY

Follow-up Procedure

You can click **Disable** under the **Operation** column on the right of a tracker to disable the tracker. After the tracker is disabled, the system will stop recording operations, but you can still view existing operation records.

You can click **Delete** under the **Operation** column on the right of a tracker to delete the tracker. Deleting a tracker has no impact on existing operation records. When you enable CTS again, you can view operation records that have been generated.

11.3 Using Logging to Record OBS Logs

After logging is enabled for a bucket, OBS automatically converts bucket logs into objects following the naming rules and writes the objects into a target bucket.

Scenarios

You can enable logging to facilitate analysis or audit. Access logs enable a bucket owner to analyze the property, type, or trend of requests to the bucket in depth. When the logging function of a bucket is enabled, OBS will log access requests for the bucket automatically, and write the generated log files to the specified bucket (target bucket).

You need to specify a bucket for storing log files when enabling logging for a bucket. Log files can be stored in any bucket in the region where the logged bucket resides, including the logged bucket itself. To better manage logs, you are advised to store log files in a bucket other than the logged bucket. If log files are stored in the logged bucket, OBS creates additional logs for writing log files to the bucket, which takes up extra storage space that will increase your costs and makes it more difficult for you to locate required logs.

NOTICE

- Uploading bucket logs to the target bucket incurs billable PUT requests. For details about the pricing, see **Requests**.

After logging is enabled, the log delivery user group will be automatically granted the permission to read the bucket ACL and write the bucket where logs are saved. If you manually disable such permissions, bucket logging will fail.

OBS can log bucket access requests for further request analysis or log audit.

Logs occupy the OBS storage that incurs costs, so OBS does not collect bucket access logs by default.

OBS creates log files and uploads them to a specified bucket. To perform these operations, OBS must be granted required permissions. Therefore, before configuring logging for a bucket, you need to create an IAM agency for OBS and add this agency when configuring logging for the bucket. By default, when configuring permissions for an agency, you only need to grant the agency the permission to upload log files (PutObject) to the bucket for storing log files. In the following example, **mybucketlogs** is the bucket. If the log storage bucket has server-side encryption enabled, the agency also requires the **KMS Administrator** permission for the region where the bucket is located.

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

After logging is configured, you can view operation logs in the bucket that stores the logs in approximately fifteen minutes.

The following shows an example access log of the target bucket:

```
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" - -
```

The access log of each bucket contains the following information.

Table 11-7 Bucket log format

Name	Example	Description
BucketOwner	787f2f92b20943998a4fe2ab75eb09b8	Account ID of the bucket owner
Bucket	bucket	Name of the bucket
Time	[13/Aug/2015:01:43:42 +0000]	Timestamp of the request (UTC)
Remote IP	xx.xx.xx.xx	IP address from where the request is initiated
Requester	787f2f92b20943998a4fe2ab75eb09b8	Requester ID ● When an account initiates a request, this parameter value is the account ID. When an IAM user initiates a request, this parameter value is the ID of the account where the IAM user belongs. ● When an anonymous user initiates a request, this parameter value is **Anonymous**.
RequestID	281599BACAD9376ECE141B842B94535B	Request ID

Name	Example	Description
Operation	REST.GET.BUCKET.LOCATION	Name of the operation See **Table 11-8** for common operations and their description.
Key	-	Object name
Request-URI	GET /bucket?location HTTP/1.1	URI of the request
HTTPStatus	200	Response code
ErrorCode	-	Error code
BytesSent	211	Size of the HTTP response, expressed in bytes
ObjectSize	-	Object size (bytes) **NOTE** ● When OBS deletes an object, it does not log the object's size. In the object deletion log, the value of **ObjectSize** is **0**. ● If error code 4*XX* is returned, the value of **ObjectSize** is **-**, indicating that the specific object size is not displayed.
TotalTime	6	Processing time on the server (ms)
Turn-AroundTime	6	Total time for processing the request (ms) **NOTE** This parameter can also be written as **TotalTime**.
Referer	-	Header field **Referer** of the request
User-Agent	HttpClient	**User-Agent** header of the request
VersionID	-	Version ID carried in the request
STSLogUrn	-	Federated authentication and agency information
StorageClass	STANDARD_IA	Current storage class of the object

Name	Example	Description
TargetStorageClass	GLACIER	Storage class that the object will be transited to
DentryName	12456/file.txt	● For a parallel file system, this field indicates an internal identifier of a file or directory. Its value consists of a parent directory inode number and a file or directory name. ● For a bucket, the value of this field is **-**.
IAMUserID	8f3b8c53d29244a780084f2b8c106c32	ID of an IAM user. When a request is initiated by an anonymous user, **Anonymous** is logged.

Table 11-8 Common operations

Operation	Description	Operation	Description
REST.GET.SERVICE	Lists buckets.	REST.GET.ENCRYPTION	Obtains the bucket encryption configuration.
REST.PUT.BUCKET	Creates a bucket.	REST.DELETE.ENCRYPTION	Deletes the bucket encryption configuration.
REST.HEAD.BUCKET	Views the bucket information.	REST.PUT.OTM_DIRECT_COLD_ACCESS	Configures direct reading for Archive objects in a bucket.
REST.GET.BUCKETVERSIONS	Lists objects in a bucket.	REST.GET.OTM_DIRECT_COLD_ACCESS	Obtains the direct reading configuration of a bucket.
REST.GET.BUCKET	Obtains the bucket metadata.	REST.DELETE.OTM_DIRECT_COLD_ACCESS	Deletes the direct reading configuration of a bucket.

Operation	Description	Operation	Description
REST.GET.BUCKET.LOCATION	Obtains the bucket location.	REST.PUT.BUCKET.WEBSITE	Configures static website hosting for a bucket.
REST.DELETE.BUCKET	Deletes a bucket.	REST.GET.BUCKET.WEBSITE	Obtains the static website hosting configuration of a bucket.
REST.PUT.POLICY	Configures a bucket policy.	REST.DEL.BUCKET.WEBSITE	Deletes the static website hosting configuration of a bucket.
REST.GET.POLICY	Obtains a bucket policy.	REST.PUT.BUCKET.CORS	Configures CORS for a bucket.
REST.DELETE.POLICY	Deletes a bucket policy.	REST.GET.BUCKET.CORS	Obtains the CORS configuration of a bucket.
REST.PUT.ACL	Configures an ACL for a bucket or an object.	REST.DEL.BUCKET.CORS	Deletes the CORS configuration of a bucket.
REST.GET.ACL	Obtains a bucket ACL or an object ACL.	REST.OPTIONS.BUCKET	Checks bucket OPTIONS.
REST.PUT.LOGGING_STATUS	Configures logging for a bucket.	REST.OPTIONS.OBJECT	Checks object OPTIONS.
REST.GET.LOGGING_STATUS	Obtains the bucket logging configuration.	REST.PUT.OBJECT	Uploads an object with PUT.
REST.PUT.BUCKET.LIFECYCLE	Configures a bucket lifecycle rule.	REST.POST.OBJECT	Uploads an object with POST.
REST.GET.LIFECYCLE	Obtains the lifecycle configuration of a bucket.	REST.COPY.OBJECT	Copies an object.
REST.DEL.LIFECYCLE	Deletes the lifecycle configuration of a bucket.	REST.GET.OBJECT	Obtains the object content.
REST.PUT.VERSIONING	Configures versioning for a bucket.	REST.HEAD.OBJECT	Obtains the object metadata.

Operation	Description	Operation	Description
REST.GET.VERSIONING	Obtains the bucket versioning status.	REST.DELETE.OBJECT	Deletes an object.
REST.GET.BUCKET.STORAGE.POLICY	Configures the default storage class for a bucket.	REST.TRANSITION.STORAGECLASS.OBJECT	Changes the storage class of an object.
REST.PUT.BUCKET.STORAGE.POLICY	Obtains the default storage class of a bucket.	OP_MULTIPLE_DELETEOBJECT	Batch deletes objects.
REST.PUT.REPLICATION	Configures cross-region replication for a bucket.	REST.POST.RESTORE	Restores an Archive object.
REST.DELETE.REPLICATION	Deletes the cross-region replication configuration of a bucket.	REST.APPEND.OBJECT	Appends data to an object.
REST.GET.REPLICATION	Obtains the cross-region replication configuration of a bucket.	REST.MODIFY.OBJECT.META	Modifies object metadata.
REST.PUT.TAGGING	Configures tags for a bucket.	REST.TRUNCATE.OBJECT	Truncates an object.
REST.GET.TAGGING	Obtains bucket tags.	REST.RENAME.OBJECT	Renames an object.
REST.DEL.TAGGING	Deletes bucket tags.	REST.GET.UPLOADS	Lists the initiated multipart uploads in a bucket.
REST.PUT.BUCKET_QUOTA	Configures a storage quota for a bucket.	REST.POST.UPLOADS	Initiates a multipart upload.
REST.GET.BUCKET.QUOTA	Queries the bucket storage quota.	REST.PUT.PART	Uploads a part.
REST.GET.BUCKET.STORAGEINFO	Queries the information about the used space in a bucket.	REST.COPY.PART	Copies a part.
REST.PUT.BUCKET.INVENTORY	Configures inventories for a bucket.	REST.GET.UPLOAD	Lists uploaded parts.

Operation	Description	Operation	Description
REST.GET.BUCKET.INVENTORY	Obtains or lists bucket inventories.	REST.POST.UPLOAD	Assembles parts.
REST.DELETE.BUCKET.INVENTORY	Deletes bucket inventories.	REST.DELETE.UPLOAD	Aborts a multipart upload.
REST.PUT.CUSTOMDOMAIN	Configures a custom domain name for a bucket.	REST.CLEAR.EXPIRE.UPLOAD	Deletes expired segments.
REST.GET.CUSTOMDOMAIN	Obtains the custom domain name of a bucket.	REST.DELETE.CUSTOMDOMAIN	Deletes a custom domain name of a bucket.
REST.PUT.ENCRYPTION	Configures encryption for a bucket.	-	-

Ways to Configure Logging for a Bucket

You can use OBS Console, APIs, or SDKs to configure logging for a bucket.

Using OBS Console

Step 1 In the navigation pane of **OBS Console**, choose **Object Storage**.

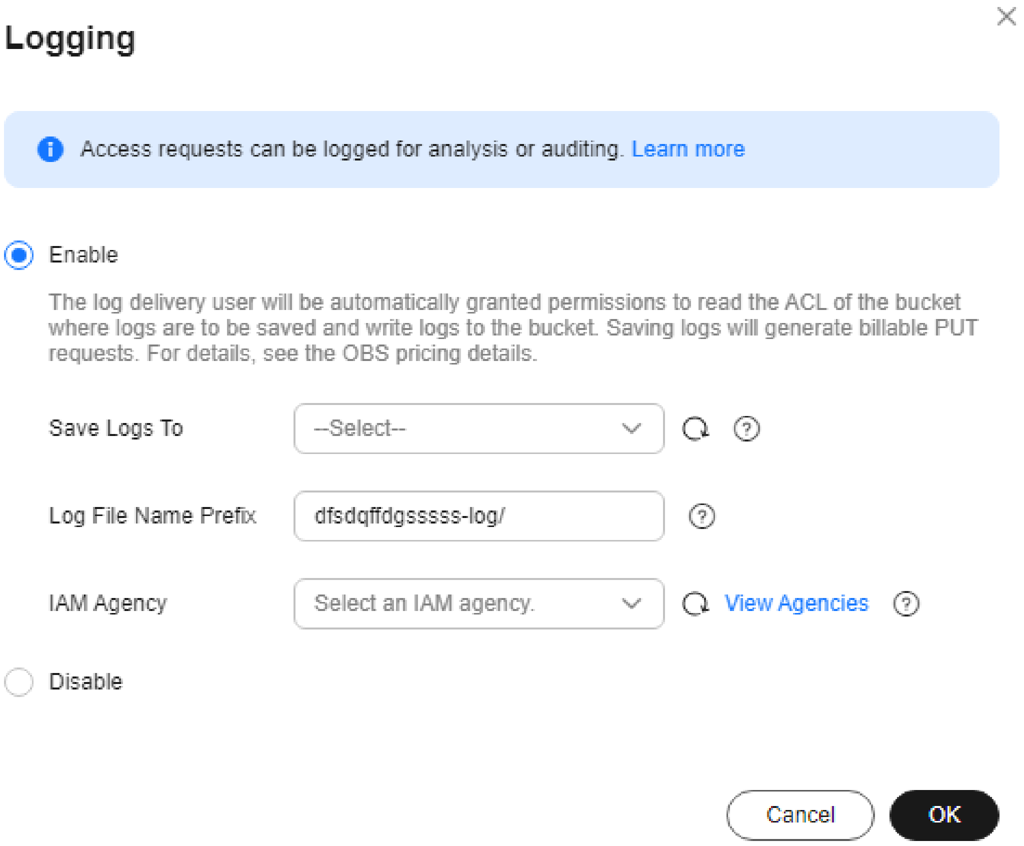
Step 2 In the bucket list, click the bucket you want to operate. The **Objects** page is displayed.

Step 3 In the navigation pane, choose **Overview**.

Step 4 In the **Basic Configurations** area, click **Logging**. The **Logging** dialog box is displayed.

Step 5 Select **Enable**. For details, see **Figure 11-3**.

Figure 11-3 Logging



Logging

Access requests can be logged for analysis or auditing. Learn more

● Enable

The log delivery user will be automatically granted permissions to read the ACL of the bucket where logs are to be saved and write logs to the bucket. Saving logs will generate billable PUT requests. For details, see the OBS pricing details.

Save Logs To: --Select--

Log File Name Prefix: dfsdqffdgsssss-log/

IAM Agency: Select an IAM agency.

○ Disable

Step 6 Select an existing bucket where you want to store log files. Log delivery users of the selected bucket will be automatically granted the permissions to read the bucket ACL and write logs to the bucket.

Step 7 Enter a prefix for the **Log File Name Prefix**.

After logging is enabled, generated logs are named in the following format:

*<Log File Name Prefix>*YYYY-mm-DD-HH-MM-SS-*<UniqueString>*

- *<Log File Name Prefix>* is the shared prefix of log file names.
- **YYYY-mm-DD-HH-MM-SS** indicates when the log is generated.
- *<UniqueString>* indicates a character string generated by OBS.

On OBS Console, if the configured *<Log File Name Prefix>* ends with a slash (/), logs generated in the bucket are stored in the *<Log File Name Prefix>* folder in the bucket, facilitating the management of log files.

Example:

- If the bucket named **bucket** is used to save log files, and the log file name prefix is set to **bucket-log/**, all log files delivered to this bucket are saved in the **bucket-log** folder. A log file is named as follows: **2015-06-29-12-22-07-N7MXLAF1BDG7MPDV**.
- If the bucket named **bucket** is used to save log files, and the log file name prefix is set to **bucket-log**, all log files are saved in the root directory of the bucket. A log file is named as follows: **bucket-log2015-06-29-12-22-07-N7MXLAF1BDG7MPDV**.

Step 8 Select an IAM agency to grant OBS the permission to upload log files to the specified bucket.

By default, you only need to grant the agency the permission to upload log files (PutObject) to the log storage bucket (**mybucketlogs** as an example). If the log storage bucket has Server-Side Encryption enabled, the agency also requires the **KMS Administrator** permission for the region where the bucket is located.

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

You can choose an existing IAM agency from the drop-down list or click **Create Agency** to create one. For details about how to create an agency, see **Creating an Agency for Uploading Logs**.

Step 9 Click **OK**.

After logging is configured, you can view operation logs in the bucket that stores the logs in approximately fifteen minutes.

----End

Using the API

Configuring Logging for a Bucket

Using SDKs

Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

Related Operations

- Disable bucket logging.

If you no longer need to record logs, in the **Logging** dialog box, click **Disable** and then click **OK**. After logging is disabled, logs are not recorded, but existing logs in the target bucket will be retained.

- Configure an agency for uploading logs.

Creating an Agency for Uploading Logs

Step 1 In the **Logging** dialog box, click **Create Agency** to jump to the **Agencies** page on the **Identity and Access Management** console.

Step 2 Click **Create Agency**.

Step 3 Enter an agency name.

Step 4 Select **Cloud service** for the **Agency Type**.

Step 5 Select **Object Storage Service (OBS)** for **Cloud Service**.

Step 6 Set a validity period.

Step 7 Click **Next**.

Step 8 On the **Select Policy/Role** page, select a custom policy that has the permission to upload data to the log storage bucket and click **Next**.

If no custom policy is available, create one by referring to **Creating a Custom Policy**.

Select **Global services** for **Scope**. Select **JSON** for **Policy View**. The policy content is as follows.

NOTE

When coding the policy content in an actual scenario, replace **mybucketlogs** with the actual bucket name:

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

Step 9 On the **Select Scope** page, select **Global services** for **Scope** and click **OK**.

Step 10 (Optional) If the log storage bucket has server-side encryption enabled, the agency also requires the **KMS Administrator** permission for the region where the bucket is located.

1. Go to the **Agencies** page of the IAM console and click the name of the agency created in the previous step.
2. Choose the **Permissions** tab and click **Authorize**.
3. On the **Select Policy/Role** page, search for and select **KMS Administrator**. Then, click **Next**.
4. On the **Select Scope** page, select **Region-specific projects** for **Scope**. Then, select the project in the region where the log storage bucket is located.

----End

12 Parallel File System

12.1 Overview of PFS

About PFS

Parallel File System (PFS) is a high-performance semantic file system provided by OBS. It features access latency in milliseconds, TB/s-level bandwidth, and millions of IOPS.

You can access data in a parallel file system via OBS APIs.

Application Scenarios

PFS is highly compatible, scalable, and reliable, and delivers amazing performance.

It is mainly used in the following scenarios:

Big data: log analysis, content recommendation, operation reports, user profiling, and interactive analysis

Using PFS

You can use OBS Console or RESTful APIs to process files stored in PFS anytime, anywhere and retrieve the processed files quickly. PFS supports both Portable Operating System Interface (POSIX) and OBS APIs, so you can process files the same way you process objects. This achieves interoperability between objects and files.

The table below describes the ways to use PFS in detail.

NOTE

Access permissions for OBS also apply to PFS. Before using PFS, make sure that you have the permissions required to access OBS resources.

Table 12-1 Ways to use PFS

Way	Function	Reference
PFS Console	On the console, you can create parallel file systems and manage them.	**Creating a Parallel File System**
OBS API	You can make API calls to use parallel file systems.	**Compatibility Between OBS APIs and PFS**
Other OBS tools	PFS allows for interoperability between objects and files, so you can also use other OBS tools (such as obsutil and OBS Browser+) to access PFS.	**OBS Tools**

Billing

Parallel file systems support both pay-per-use and yearly/monthly (resource packages) billing modes. For details about resource packages, see **Resource Package Overview**.

For details about PFS pricing, see **Product Pricing Details**.

12.2 Features of PFS

Compatibility Between OBS APIs and PFS

You can call some OBS APIs to use PFS. There may be additional requirements when you call these APIs.

For details about the OBS APIs, see **Object Storage Service API Reference**.

APIs for Basic Bucket Operations

Table 12-2 APIs for basic bucket operations

API	PFS Compatible	Differences
Listing buckets	Yes	The **x-obs-bucket-type:POSIX** header is required for obtaining the list of parallel file systems.
Creating a bucket	Yes	The **x-obs-fs-file-interface:Enabled** header is required for creating a parallel file system.
Lists objects in a bucket	Yes	-

API	PFS Compatible	Differences
Obtaining bucket metadata	Yes	-
Obtaining bucket region locations	Yes	-
Deleting a bucket	Yes	-

APIs for Advanced Bucket Settings

Table 12-3 APIs for advanced bucket settings

API	PFS Compatible	Differences
Configuring a Bucket Policy	Yes	-
Obtaining bucket policy information	Yes	-
Deleting a bucket policy	Yes	-
Configuring a bucket ACL	Yes	-
Obtaining bucket ACL information	Yes	-
Configuring logging for a bucket	Yes	-
Obtaining a bucket logging configuration	Yes	-
Configuring bucket lifecycle rules	Yes	-
Obtaining bucket lifecycle configuration	Yes	-
Deleting bucket lifecycle rules	Yes	-
Configuring versioning for a bucket	No	-
Obtaining bucket versioning status	No	-

API	PFS Compatible	Differences
Configuring event notification for a bucket	Yes	-
Obtains the event notification configuration of a bucket	Yes	-
Configuring storage class for a bucket	No	-
Obtaining bucket storage class information	No	-
Configuring cross-region replication for a bucket	No	-
Obtaining the cross-region replication configuration of a bucket	No	-
Deleting the cross-region replication configuration of a bucket	No	-
Configuring tags for a bucket	Yes	-
Obtaining bucket tags	Yes	-
Deleting bucket tags	Yes	-
Configuring bucket storage quota	Yes	-
Querying bucket storage quota	Yes	-
Querying information about used space in a bucket	Yes	-
Configuring bucket inventories	No	-
Obtaining bucket inventories	No	-

API	PFS Compatible	Differences
Listing bucket inventories	No	-
Deleting bucket inventories	No	-
Configuring a custom domain name for a bucket	Yes	-
Obtaining the custom domain name of a bucket	Yes	-
Deleting a custom domain name of a bucket	Yes	-
Configuring bucket encryption	No	-
Obtaining bucket encryption configuration	No	-
Deleting the encryption configuration of a bucket	No	-
Configuring the direct reading policy for Archive objects in a bucket	Yes	-
Obtaining the direct reading policy for Archive objects in a bucket	Yes	-
Deleting the direct reading policy for Archive objects in a bucket	Yes	-

APIs for Static Website Hosting

Table 12-4 APIs for static website hosting

API	PFS Compatible	Differences
Configuring static website hosting for a bucket	No	-
Obtaining the static website hosting configuration of a bucket	No	-
Deleting the static website hosting configuration of a bucket	No	-
Configuring bucket CORS	Yes	-
Obtaining the CORS configuration of a bucket	Yes	-
Deleting the CORS configuration of a bucket	Yes	-
OPTIONS buckets	No	-
OPTIONS objects	No	-

APIs for Object Operations

Table 12-5 APIs for object operations

API	PFS Compatible	Differences
PUT objects	Yes	● Headers not supported: **x-obs-storage-class**, **x-obs-website-redirect-location**, **success-action-redirect**, and **x-obs-expires** ● Objects uploaded using this API cannot be directly stored in the Infrequent Access or Archive storage class and are stored in the Standard storage class by default. You can later change the storage class by using a lifecycle rule or modifying the metadata.
POST objects	Yes	Headers not supported: **x-obs-storage-class**, **x-obs-website-redirect-location**, **success-action-redirect**, and **x-obs-expires**
Copying objects	Yes	Data can be replicated only between parallel file systems or buckets that are in the same cluster.
Obtaining object content	Yes	-
Obtaining object metadata	Yes	-
Deleting an object	Yes	-
Batch deleting objects	Yes	-
Restoring Archive objects	√	-
Appending objects	No	-
Configuring object ACL	Yes	-
Obtaining object ACL information	Yes	-
Modifying object metadata	Yes	In a parallel file system, the storage class of a directory cannot be changed. To change the storage class of a file in the directory, modify the metadata of the file or use a lifecycle rule to change the storage class of files in batches.

API	PFS Compatible	Differences
Modifying an object	Yes	This is a PFS only API, and is not supported by OBS buckets.
Truncating an object	Yes	This is a PFS only API, and is not supported by OBS buckets.
Renaming an object	Yes	This is a PFS only API, and is not supported by OBS buckets.

APIs for Multipart Uploads

Table 12-6 APIs for multipart uploads

API	PFS Compatible	Differences
Listing initialized multipart tasks in a bucket	Yes	-
Initiating multipart upload tasks	Yes	-
Uploading parts	Yes	-
Copying parts	Yes	Copying parts is not supported for an appended file.
Listing uploaded parts	Yes	-
Merging parts	Yes	-
Canceling multipart tasks	Yes	-

Permissions Configuration

The use cases and main functions of object access control also work on files in parallel file systems. For more information, see **Permissions Configuration Guide**.

Differences Between File and Object Permission Configurations

To exactly match a specific directory, the resource path in the policy must end with a slash (/). When checking permissions, parallel file systems consider objects as directories. Since objects do not end with a slash (/), PFS will add a slash (/) to the end of objects and then perform a policy matching.

IAM Permission Configuration Examples

Example 1: Grant a user the permissions required to download **dir_1**, excluding its subdirectories.

In the following configuration, the resource path ends with a slash (/). In such case, a success response can be returned when **dir_1** or **dir_1/** is contained in the URL of a head request.

Note that this configuration is not applied to subdirectories or files in **dir_1**. Therefore, a failure response will be returned if a head request is sent to **dir_1/file1**.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "obs:object:GetObject",
            ],
            "Resource": [
                "obs:*:*:object:examplebucket/dir_1/",
            ]
        }
    ]
}
```

Example 2: Grant a user the permissions required to download **dir_1** and its subdirectories.

In the following configuration, the resource path uses prefix-based matching and ends with a wildcard (*). In such case, a success response can be returned when a head request is sent to **dir_1/file1**.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "obs:object:GetObject",
            ],
            "Resource": [
                "obs:*:*:object:examplebucket/dir_1/*",
            ]
        }
    ]
}
```

Bucket Policy Configuration Examples

Example 1: Grant a user the permissions required to download **dir_1**, excluding its subdirectories.

In the following configuration, the resource path ends with a slash (/). In such case, a success response can be returned when **dir_1** or **dir_1/** is contained in the URL of a head request.

```
{
    "Statement":[
    {
      "Sid":"test",
      "Effect":"Allow",
    "Principal": {"ID": ["domain/b4bf1b36d9ca43d984fbcb9491b6fce9:user/
71f3901173514e6988115ea2c26d1999"]},
    "Action":["*"],
    "Resource":[
      "examplebucket/dir_1/",
    ]
   }
  ]
}
```

Lifecycle Management

The use cases and main functions of object lifecycle management also work on files in parallel file systems. For more information, see **Creating a Lifecycle Rule**.

To manage the lifecycle using SDKs, see **SDK Overview**.

Differences Between File and Object Lifecycle Management

- A lifecycle rule can be used to manage files, but it cannot transition a directory to the Archive storage class. However, a lifecycle rule can delete an empty directory upon expiration.
- File deletion upon expiration and transition to the Archive storage class can be configured using either the API or console, while transition to Infrequent Access can only be configured using the API. PFS does not support versioning, so lifecycle actions (including deletion upon expiration and transition to the Archive or Infrequent Access storage class) that are related to versioning cannot be applied.
- If direct reading is enabled for a parallel file system, you can read files stored in the Archive storage class without restoring them first.
- A maximum of 20 lifecycle rules can be configured for a parallel file system.
- The time applied for a lifecycle rule to work on a file is when the data of the file was last updated.
- Lifecycle rules cannot transition files to the Deep Archive storage.
- A lifecycle rule configured for a parallel file system has limits on how many directories it can apply to. It will take longer time to execute the actions defined in the rule if:

  a. There are over 100,000 level-1 subdirectories in each directory.

  b. There are over 10 million subdirectories (folders) matching the specified prefix in total.

  c. There are over 30 million files matching the specified prefix in total.

Notes for Lifecycle Management

- If renamed files or files in a renamed directory meet the conditions specified in a lifecycle rule, the time when the file data was most recently updated, not when the files were renamed, will be applied for the lifecycle rule to take effect. In addition, the effective time of the lifecycle rule may be delayed for up to seven days.
- For a file copy on a client, the lifecycle rule determines when to expire the file copy or transition it to the Archive storage class based on the file copy creation time.

  - For example, if a file, **src.txt**, was created on January 1, 2019, and was then copied to the **des.txt** file by running the **cp -a src.txt des.txt** command on September 1, 2019. Then the lifecycle rule calculated when to perform the specified actions on the file copy based on September 1, 2019.
- A lifecycle rule periodically scans directories in the file system and then deletes the directories that meet the expiration conditions. A scan starts from the deepest directory. The scanned empty directories that meet the expiration conditions will be deleted, and those non-empty directories will not be processed. Scan intervals (usually seven days) vary depending on cluster configurations. For this mechanism, a single-level directory that meets the expiration conditions will be deleted 0 to 7 days after it has been emptied. Accordingly, a two-level directory will be deleted 0 to 14 days after it has been emptied. Each time a directory level is added, the waiting time for deletion increases by seven days.

12.3 Constraints of PFS

Operations

- An existing OBS bucket cannot be changed to a parallel file system. For details about how to create a parallel file system, see **Creating a Parallel File System**.
- Custom domain names of parallel file systems cannot be configured on OBS Console. You can configure them using APIs by referring to **Configuring a Custom Domain Name**.

Functions

- **Image processing** cannot be used to process (such as slim, resize, or watermark) images stored in parallel file systems.
- Server-side encryption is not supported.
- Cross-region replication is not supported.
- Versioning is not supported.
- Bucket inventory is not supported.
- Static website hosting is not supported.
- Configuration of default storage class for a parallel file system is not supported.
- Deep Archive storage is not supported.
- A parallel file system can be mounted to multiple Linux servers for concurrent reads, but this is not recommended for concurrent writes.

Performance

- A parallel file system provides a maximum bandwidth of 10 MB/s per TB by default.

Naming

- In a parallel file system, a file name cannot contain two consecutive slashes (//). For example, if you name a file as **test//123.txt**, an error will be reported.

12.4 Creating a Parallel File System

You can create a parallel file system on OBS Console.

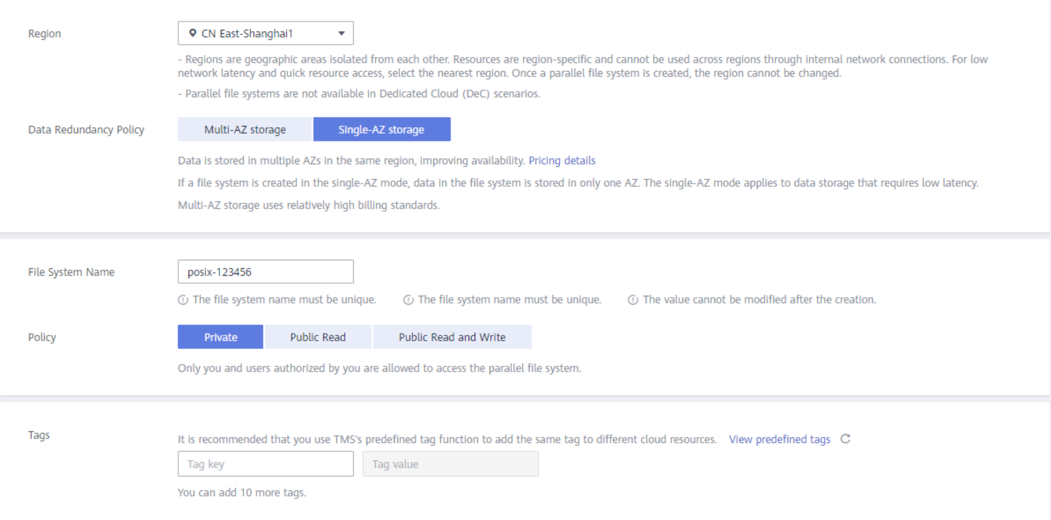
Procedure

Step 1 On the console homepage, click **Service List** in the upper left corner and choose **Storage** > **Object Storage Service**.

Step 2 In the navigation pane, choose **Parallel File Systems**.

Step 3 In the upper right corner of the page, click **Create Parallel File System**.

Figure 12-1 Creating a parallel file system



Region: CN East-Shanghai1

- Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region. Once a parallel file system is created, the region cannot be changed.

- Parallel file systems are not available in Dedicated Cloud (DeC) scenarios.

Data Redundancy Policy: Multi-AZ storage | Single-AZ storage

Data is stored in multiple AZs in the same region, improving availability. Pricing details

If a file system is created in the single-AZ mode, data in the file system is stored in only one AZ. The single-AZ mode applies to data storage that requires low latency.

Multi-AZ storage uses relatively high billing standards.

File System Name: posix-123456

ⓘ The file system name must be unique. ⓘ The file system name must be unique. ⓘ The value cannot be modified after the creation.

Policy: Private | Public Read | Public Read and Write

Only you and users authorized by you are allowed to access the parallel file system.

Tags

It is recommended that you use TMS's predefined tag function to add the same tag to different cloud resources. View predefined tags

Tag key | Tag value

You can add 10 more tags.

Step 4 Select a region and enter a name for the parallel file system.

NOTE

- Once a parallel file system is created, its name cannot be changed.
- URLs do not support uppercase letters and cannot distinguish between names containing uppercase or lowercase letters. Therefore, a parallel file system name can contain only lowercase letters, digits, periods (.), and hyphens (-). For example, if you attempt to access the parallel file system **MyFileSystem** using a URL, the file system name will be resolved to **myfilesystem**, causing an access error.

Step 5 Configure a data redundancy policy. If you select multi-AZ storage, data will be stored in multiple AZs for a higher reliability.

NOTE

Once a parallel file system is created, you cannot change the data redundancy policy, so take care when selecting a data redundancy policy.

Step 6 Configure a policy. You can select **Private**, **Public Read**, or **Public Read/Write** for the parallel file system.

Step 7 Configure direct reading. With direct reading enabled, you can directly download objects in the Archive storage class without restoring them first. Direct reading is a billable function. For details, see **Product Pricing Details**.

Step 8 (Optional) Add tags. Tags are used to identify parallel file systems in OBS, for the purpose of classification. Each tag is represented by one key-value pair. For details about how to add a tag, see **Tags**.

Step 9 (Optional) Buy storage packages. By default, you are billed on a pay-per-use basis for using parallel file systems. You can also purchase **storage packages** to save more. After selecting a required package, go to the **Confirm** page to complete the purchase.

Storage packages can also be purchased after the parallel file system is created.

Step 10 Confirm the settings at the bottom of the page and click **Create Now**.

Step 11 View the file system you created just now in the parallel file system list.

Then, you can use the parallel file system the same way you use a bucket. For details about how to use PFS, see **Using PFS**.

----End