ALIBABA CLOUD

Alibaba Cloud

Object Storage Service
Developer Guide

Document Version: 20220324

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	**Danger:** Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	**Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Usage instructions

This topic lists the information that you need to know before you use Alibaba Cloud Object Storage Service (OSS).

For more information about how to get started with OSS, see Get Started with Object Storage Service.

The following table describes the manuals and guides that can help you utilize OSS.

Resource	Description
OSS Developer Guide	Describes the core concepts, features, and implementation methods of Alibaba Cloud OSS, as well as examples on how to use the OSS API and SDK.
OSS Best Practices	Describes the application scenarios and configuration practice of Alibaba Cloud OSS.
OSS SDK Reference	Describes how to use OSS SDK for mainstream programming languages to perform routine operations.
OSS API Reference	Describes the RESTful API operations supported by Alibaba Cloud OSS and provides related examples.
OSS tools	Describes tools provided by Alibaba Cloud to help you manage OSS resources.
Console User Guide	Describes all operations supported by the OSS console, a web-based GUI platform used to manage Alibaba Cloud OSS.
IMG User Guide	Describes how to use Image Processing (IMG) provided by Alibaba Cloud OSS.

2.Terms

This topic describes several basic terms used in Object Storage Service (OSS).

bucket

The container for OSS objects. Each object in OSS is contained in a bucket. You can configure various attributes of a bucket, including the region, access control list (ACL), and storage class. You can create buckets of different storage classes to store data based on your requirements.

- OSS does not use a hierarchical structure for objects, but instead uses a flat structure. Each object belongs to a bucket.
- You can create multiple buckets.
- A bucket name must be unique in OSS within an Alibaba Cloud account. Bucket names cannot be changed after the buckets are created.
- A bucket can contain an infinite number of objects.

OSS supports the following bucket naming conventions:

- The name can contain only lowercase letters, digits, and hyphens (-).
- The name must start and end with a lowercase letter or a digit.
- The name must be 3 to 63 characters in length.

object

The basic unit for data operations in OSS. Objects are also known as files. OSS does not use a hierarchical structure for objects, but instead uses a flat structure. All elements are stored as objects in buckets. However, OSS supports directories as a concept to group objects and simplify management. An object is composed of object metadata, user data, and a key. A key is used to identify an object in a bucket. Object metadata is a group of key-value pairs that define the properties of an object, such as the last modified time and the object size. You can also assign user metadata to the object.

The lifecycle of an object starts when the object is uploaded, and ends when the object is deleted. Throughout the lifecycle, content can be appended only to objects created by using append upload. If you want to modify the content of an object, you must upload a new object that has the same name as the existing object to replace the existing object.

The name of an object must comply with the following conventions:

- The name must be encoded in UTF-8.
- The name must be 1 to 1,023 characters in length.
- The name cannot start with a forward slash (/) or a backslash (\).

> ? **Note** Object names are case-sensitive. Unless otherwise stated, objects and files mentioned in OSS documents are called objects.

ObjectKey

In SDKs for different programming languages, ObjectKey, Key, and ObjectName indicate the full path of the object. You must specify the full path of an object when you perform operations on the object. For example, when you upload an object to a bucket, ObjectKey indicates the full path that includes the extension of the object. For example, you can set ObjectKey to abc/efg/123.jpg.

region

The physical location of OSS resources. When you create a bucket, you can select a region based on the cost and source of the requests. In most cases, the closer a user is located from an OSS region, the faster the access. For more information, see Regions and endpoints.

A region is specified when a bucket is created. After a bucket is created, its region cannot be changed. All objects in this bucket are stored in the corresponding region. Regions are configured for buckets instead of objects.

endpoint

The domain name that is used to access OSS. OSS uses HTTP RESTful APIs to provide services. Different regions are accessed by using different endpoints. A region has different endpoints for access over the internal network and for access over the Internet. For example, the public endpoint used to access OSS data in the China (Hangzhou) region is oss-cn-hangzhou.aliyuncs.com, and the internal endpoint is oss-cn-hangzhou-internal.aliyuncs.com. For more information, see Regions and endpoints.

AccessKey pair

The access credential that is used to identify the requester. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. OSS uses an AccessKey pair to implement symmetric encryption and verify the identity of a requester. The AccessKey ID is used to identify a user. The AccessKey secret is used to encrypt and verify the signature string. The AccessKey secret must be kept confidential. OSS supports AccessKey pairs obtained by using the following methods:

- The bucket owner applies for an AccessKey pair.
- The bucket owner uses Resource Access Management (RAM) to assign an AccessKey pair to a third party.
- The bucket owner uses Security Token Service (STS) to assign an AccessKey pair to a third party.

For more information about AccessKey pairs, see Obtain an AccessKey pair.

strong consistency

The feature that requires object operations in OSS be atomic, which indicates that operations can either succeed or fail without intermediate states. To ensure that users can access only complete objects, OSS does not return partially uploaded objects.

Object operations in OSS are highly consistent. For example, when a user receives an upload (PUT) success response, the uploaded object can be immediately read, and copies of the object are created for redundancy. Therefore, the situations where data is not obtained when a user performs a read-after-write operation do not exist. The same is true for delete operations. After a user deletes an object, the object and its copies no longer exist.

data redundancy mechanism

The data redundancy mechanism that is implemented based on erasure coding and multiple replicas. Copies of each object are stored in different servers within the same region. This way, data durability and availability are ensured when hardware failures occur.

- Operations on objects in OSS are highly consistent. For example, when a user receives an upload or copy success response, the uploaded object can be immediately read, and copies of the object are created for redundancy.

- To ensure complete data transmission, OSS calculates the checksum of the network traffic packets and checks for errors when packets are transmitted between the client and the server.
- The data redundancy mechanism of OSS can prevent data loss even when two storage facilities are damaged at the same time.
  - After data is stored in OSS, OSS regularly checks whether copies of the data are lost. Then, OSS recovers lost copies to ensure data durability and data availability.
  - OSS periodically verifies the integrity of data to detect data corruption caused by errors and hardware failures. If data is partially corrupted or lost, OSS recovers the data by using copies of the data.

Comparison between OSS and file systems

Item	OSS	File system
Data model	OSS is a distributed object storage service that stores data as key-value pairs.	A file system uses a typical tree structure for directory indexing.
Data retrieval	Objects are retrieved based on unique object names (keys). For example, the object name test1/test.jpg does not necessarily indicate that the object is stored in a directory named test1. In OSS, test1/test.jpg is only a string. test1/test.jpg is not essentially different from example.jpg. Therefore, similar amounts of resources are consumed regardless of which object you access.	To access a file named test1/test.jpg, you must first access the test1 directory and then query the test.jpg file in this directory.
Advantages	OSS supports concurrent access from a large number of users.	The file system supports modifications on files, such as modifying the content at a specified offset location or truncating the end of a file. It also supports directory operations such as rename, delete, and move directories.
Disadvantages	Objects stored in OSS cannot be modified. A specific operation must be called to append an object. Generated objects are different from objects uploaded by using other methods. To modify an object, you must upload the entire object again. OSS can simulate features similar to those of directories. However, such operations consume a large amount of resources. For example, if you want to rename the test1 directory to test2, OSS must copy all objects whose names start with test1/. Then, OSS creates objects whose names start with test2/. This operation consumes a large amount of resources. Therefore, we recommend that you do not perform such operations in OSS.	The performance of the file system is subject to the performance of a single device. More files and directories in the file system consume greater resources and take longer processing time.

We recommend that you do not map operations on OSS objects to file systems because it is inefficient. If you mount OSS as a file system, we recommend that you perform only add, delete, and read operations on objects. You can take advantage of OSS to process and store large amounts of unstructured data such as images, videos, and documents.

The following table describes the differences in some terms between OSS and file systems.

OSS	File system
Object	file
Bucket	home directory
Region	N/A
Endpoint	N/A
AccessKey	N/A
N/A	multilevel directory
GetService	obtain the list of home directories
GetBucket	obtain the list of files
PutObject	add a file
AppendObject	append data to an existing file
GetObject	read a file
DeleteObject	delete a file
N/A	modify file content
CopyObject (same destination and source objects)	modify file attributes
CopyObject (different destination and source objects)	copy a file
N/A	rename a file

3.Endpoint

3.1. Regions and endpoints

Object Storage Service (OSS) data centers are located in regions. Endpoints are the domain names that the other services can use to access OSS. This topic describes the mappings between regions and endpoints.

Regions and OSS endpoints for access in the public cloud

The following table describes the public endpoints, internal endpoints, and accelerate endpoints of OSS in each region in the public cloud.

Region	Region ID	Public endpoint	Internal endpoint[1]
China (Hangzhou)	oss-cn-hangzhou	oss-cn-hangzhou.aliyuncs.com	oss-cn-hangzhou-internal.aliyuncs.com
China (Shanghai)	oss-cn-shanghai	oss-cn-shanghai.aliyuncs.com	oss-cn-shanghai-internal.aliyuncs.com
China (Qingdao)	oss-cn-qingdao	oss-cn-qingdao.aliyuncs.com	oss-cn-qingdao-internal.aliyuncs.com
China (Beijing)	oss-cn-beijing	oss-cn-beijing.aliyuncs.com	oss-cn-beijing-internal.aliyuncs.com
China (Zhangjiakou)	oss-cn-zhangjiakou	oss-cn-zhangjiakou.aliyuncs.com	oss-cn-zhangjiakou-internal.aliyuncs.com
China (Hohhot)	oss-cn-huhehaote	oss-cn-huhehaote.aliyuncs.com	oss-cn-huhehaote-internal.aliyuncs.com
China (Ulanqab)	oss-cn-wulanchabu	oss-cn-wulanchabu.aliyuncs.com	oss-cn-wulanchabu-internal.aliyuncs.com
China (Shenzhen)	oss-cn-shenzhen	oss-cn-shenzhen.aliyuncs.com	oss-cn-shenzhen-internal.aliyuncs.com
China (Heyuan)	oss-cn-heyuan	oss-cn-heyuan.aliyuncs.com	oss-cn-heyuan-internal.aliyuncs.com
China (Guangzhou)	oss-cn-guangzhou	oss-cn-guangzhou.aliyuncs.com	oss-cn-guangzhou-internal.aliyuncs.com
China (Chengdu)	oss-cn-chengdu	oss-cn-chengdu.aliyuncs.com	oss-cn-chengdu-internal.aliyuncs.com
China (Hong Kong)	oss-cn-hongkong	oss-cn-hongkong.aliyuncs.com	oss-cn-hongkong-internal.aliyuncs.com
US (Silicon Valley) *	oss-us-west-1	oss-us-west-1.aliyuncs.com	oss-us-west-1-internal.aliyuncs.com
US (Virginia) *	oss-us-east-1	oss-us-east-1.aliyuncs.com	oss-us-east-1-internal.aliyuncs.com
Singapore *	oss-ap-southeast-1	oss-ap-southeast-1.aliyuncs.com	oss-ap-southeast-1-internal.aliyuncs.com
Australia (Sydney) *	oss-ap-southeast-2	oss-ap-southeast-2.aliyuncs.com	oss-ap-southeast-2-internal.aliyuncs.com
Malaysia (Kuala Lumpur) *	oss-ap-southeast-3	oss-ap-southeast-3.aliyuncs.com	oss-ap-southeast-3-internal.aliyuncs.com
Indonesia (Jakarta) *	oss-ap-southeast-5	oss-ap-southeast-5.aliyuncs.com	oss-ap-southeast-5-internal.aliyuncs.com
Japan (Tokyo) *	oss-ap-northeast-1	oss-ap-northeast-1.aliyuncs.com	oss-ap-northeast-1-internal.aliyuncs.com
India (Mumbai) *	oss-ap-south-1	oss-ap-south-1.aliyuncs.com	oss-ap-south-1-internal.aliyuncs.com
Germany (Frankfurt) *	oss-eu-central-1	oss-eu-central-1.aliyuncs.com	oss-eu-central-1-internal.aliyuncs.com
UK (London)	oss-eu-west-1	oss-eu-west-1.aliyuncs.com	oss-eu-west-1-internal.aliyuncs.com
UAE (Dubai) *	oss-me-east-1	oss-me-east-1.aliyuncs.com	oss-me-east-1-internal.aliyuncs.com
Philippines (Manila)	oss-ap-southeast-6	oss-ap-southeast-6.aliyuncs.com	oss-ap-southeast-6-internal.aliyuncs.com

Note

- For more information about the usage and composition rules of OSS domain names, see OSS domain names.
- By default, `oss.aliyuncs.com` maps to the public endpoint of the China (Hangzhou) region, and `oss-internal.aliyuncs.com` maps to the internal endpoint of the China (Hangzhou) region.
- [1]: The internal endpoint can be used by other Alibaba Cloud services in the same region as OSS to access OSS. When you access OSS from Elastic Compute Service (ECS) instances, we recommend that you use the internal endpoint of OSS. For more information, see Access to OSS resources from ECS instances by using an internal endpoint of OSS.
- The name of a region marked with the * in this topic indicates that the name of the region may be different from that of the region on the website of OSS Pricing or the website of OSS Resource Plans. However, the two different names represent the same region. For more information, visit OSS Pricing or OSS Resource Plans.
- You can use a dual-stack endpoint to access your bucket by using both IPv4 and IPv6 from your client. For more information, see OSS domain names.

Accelerate endpoints

If transfer acceleration is enabled for a bucket, the following accelerate endpoints are added:

- Global accelerate endpoint: `oss-accelerate.aliyuncs.com`. Transfer acceleration access points are distributed across the world. You can use this endpoint to accelerate data transfer for buckets in all regions.
- Accelerate endpoint of regions outside mainland China: `oss-accelerate-overseas.aliyuncs.com`. Transfer acceleration access points are distributed across

regions outside mainland China. You can use these accelerate endpoints to map a custom domain name without an ICP filing to a bucket in the China (Hong Kong) region or a region outside mainland China.

For more information, see Transfer acceleration.

3.2. OSS domain names

Object Storage Service (OSS) assigns domain names to each bucket. This topic describes the composition of OSS domain names and how to use the domain names.

Format

All requests except the GetService (ListBuckets) and DescribeRegions requests to OSS include third-level domains. The third-level domains contain bucket information.

The format is `BucketName.Endpoint` . BucketName indicates the name of your bucket. Endpoint indicates the domain name used to access the region in which your bucket is located.

OSS endpoints include internal endpoints, public endpoints, and accelerate endpoints. Accelerate endpoints include global accelerate endpoints and the accelerate endpoint of regions outside the Chinese mainland. For example, the following endpoints are used to access buckets located in the China (Hangzhou) region:

- Public endpoint: oss-cn-hangzhou.aliyuncs.com
- Internal endpoint: oss-cn-hangzhou-internal.aliyuncs.com
- Global accelerate endpoint: oss-accelerate.aliyuncs.com
- Accelerate endpoint of regions outside the Chinese mainland: oss-accelerate-overseas.aliyuncs.com

You can use internal endpoints and public endpoints without additional configurations. Before you use an accelerate endpoint, you must enable the transfer acceleration feature. For more information, see Enable transfer acceleration.

> Note
> - If you want to access OSS resources in different regions, you must use different endpoints.
> - For more information about regions and endpoints, see Regions and endpoints.
> - You can also replace a public endpoint with a custom domain name to access OSS resources. For more information, see Map custom domain names or Bind accelerate endpoints.

Access OSS over the public network

The public network indicates the Internet. OSS allows you to upload or write data to OSS over the Internet free of charge. You are charged when you download or read data from OSS.

> Note For more information about OSS fees, visit Object Storage Service Pricing and see Billable items and billing methods.

You can use one of the following methods to access OSS over the Internet:

- Method 1: Use a URL to access an OSS object

  If you use a URL to access an OSS object, the access control list (ACL) that is configured for the object determines the permissions to read and write the object.

Object ACL	Public read or public read/write	Private
URL format	`<Schema>://<Bucket>.<Public endpoint>/<Object>`	`<Schema>://<Bucket>.<Public endpoint>/<Object>?Signature information`
Parameter description	○ Schema: HTTP or HTTPS. ○ Bucket: the name of the OSS bucket. ○ Public endpoint: the domain name used to access the region where the bucket is located over the Internet. For more information about the endpoints used to access each region, see Regions and endpoints. ○ Object: the path to access the OSS object.	Parameters in URLs, except for signature information, are configured in a similar manner in which public read objects and public read/write objects are configured. The signature information of a URL includes the Expires, AccessKey ID, and Signature elements. Expires specifies the expiration time of the URL. For more information about how to add signatures to an object URL, see Add signatures to a URL.
Examples	You create a bucket named examplebucket in the China (Hangzhou) region. The bucket contains the example.txt object. The object is in the exampledir directory and allows anonymous access. In this case, the object URL is `https://examplebucket.oss-cn-hangzhou.aliyuncs.com/exampledir/example.txt` .	You create a bucket named examplebucket in the China (Hangzhou) region. The bucket contains the example.txt object. The object is in the exampledir directory. The ACL of the object is set to private. In this case, the object URL is `https://examplebucket.oss-cn-hangzhou.aliyuncs.com/exampledir/example.txt?OSSAccessKeyId=nz2pc56s936****&Expires=1141889120&Signature=vjbyPxybdZaNmGa%2ByT272YEAiv****` .

> Notice To access an object, you must include the path of an object in an OSS endpoint. If you use examplebucket.oss-cn-hangzhou.aliyuncs.com to access an object without including the path of the object in the OSS endpoint, an error is reported. To use an OSS endpoint to access an object without including the path of the object, configure static website hosting. For more information, see Overview.

- Method 2: Use an OSS SDK to configure a public endpoint

  An OSS SDK concatenates each operation to generate an endpoint. A different endpoint is generated for operations on a bucket located in a different region.

  The following code provides an example on how to use OSS SDK for Java to specify an endpoint. In this example, the endpoint is specified when OSSClient is created to manage a bucket located in the China (Hangzhou) region.

```
String accessKeyId = "<yourAccessKeyId>";
  String accessKeySecret = "<yourAccessKeySecret>";
  String endpoint = "oss-cn-hangzhou.aliyuncs.com";
  OSSClient client = new OSSClient(endpoint, accessKeyId, accessKeySecret);
```

Access OSS over the internal network

You can use an internal endpoint to communicate between Alibaba Cloud services located within the same region over the internal network. For example, you can access an OSS bucket from an Elastic Compute Service (ECS) instance over the internal network if the OSS bucket and the ECS instance are located within the same region. You are not charged for the traffic generated over the internal network. However, you are charged for requests that you send.

> **Note** For more information about OSS fees, visit Object Storage Service Pricing and see Billable items and billing methods.

You can use one of the following methods to access OSS over the internal network:

- Method 1: Use a URL to access an OSS object

  If you use a URL to access an OSS object, the access control list (ACL) that is configured for the object determines the permissions to read and write the object.

Object ACL	Public read or public read/write	Private
URL format	`<Schema>://<Bucket>.<Internal endpoint>/<Object>`	`<Schema>://<Bucket>.<Internal endpoint>/<Object>?Signature information`
Parameter description	○ Schema: HTTP or HTTPS. ○ Bucket: the name of the OSS bucket. ○ Internal endpoint: the domain name used to access ECS instances over the internal network within the same region. For more information about the endpoint of each region, see Regions and endpoints. ○ Object: the path to access the OSS object.	Parameters in URLs, except for signature information, are configured in a similar manner in which public read objects and public read/write objects are configured. The signature information of a URL includes the Expires, AccessKey ID, and Signature elements. Expires specifies the expiration time of the URL. For more information about how to add signatures to an object URL, see Add signatures to a URL.
Examples	You create a bucket named examplebucket in the China (Hangzhou) region. The bucket contains the example.txt object. The object is in the exampledir directory and allows anonymous access. In this case, the URL of the object is `https://examplebucket.oss-cn-hangzhou-internal.aliyuncs.com/exampledir/example.txt` .	You create a bucket named examplebucket in the China (Hangzhou) region. The bucket contains the example.txt object. The object is in the exampledir directory. The ACL of the object is set to private. In this case, the URL of the object is `https://examplebucket.oss-cn-hangzhou-internal.aliyuncs.com/exampledir/example.txt?OSSAccessKeyId=nz2pc56s936****&Expires=1141889120&Signature=vjbyPxybdZaNmGa%2ByT272YEAiv****` .

- Method 2: Use an OSS SDK to configure an internal endpoint and access OSS from an ECS instance over the internal network

  The following code provides an example on how to use OSS SDK for Java to specify an internal endpoint. In this example, the endpoint is set to the internal endpoint of the China (Hangzhou) region when you manage a bucket located in the China (Hangzhou) region.

```
String accessKeyId = "<yourAccessKeyId>";
  String accessKeySecret = "<yourAccessKeySecret>";
  String endpoint = "oss-cn-hangzhou-internal.aliyuncs.com";
  OSSClient client = new OSSClient(endpoint, accessKeyId, accessKeySecret);
```

  If the OSS bucket and the ECS instance are located in the same region, you can use an internal endpoint to access an OSS bucket from an ECS instance over the internal network. If OSS and the ECS instance are located in different regions, you cannot use the internal endpoint to access the OSS bucket from the ECS instance over the internal network. For example, you have two buckets in OSS and you have purchased an ECS instance located in the China (Beijing) region.

  - One bucket is named srcbucket and located in the China (Beijing) region. You can use `https://srcbucket.oss-cn-beijing-internal.aliyuncs.com` to access resources in srcbucket from the ECS instance located in the China (Beijing) region.
  - The other bucket is named destbucket and located in the China (Qingdao) region. You cannot use `https://destbucket.oss-cn-qingdao-internal.aliyuncs.com` to access resources in destbucket from the ECS instance located in the China (Beijing) region. To access resources in destbucket from the ECS instance located in the China (Beijing) region, you must use `https://destbucket.oss-cn-qingdao.aliyuncs.com` to access resources in destbucket over the Internet.

Use an accelerate endpoint to access OSS

OSS provides transfer acceleration to improve data upload and download speeds. This feature can improve user experiences in uploads and downloads when you transfer data across countries or continents. To use an accelerate endpoint to access a bucket in OSS, you must enable transfer acceleration for the bucket. After you enable transfer acceleration for a bucket, you can use the accelerate endpoint instead of the public endpoint to access the bucket and accelerate data transfer.

An accelerate endpoint is used in the example. A browser is used to access the *myphoto.jpg* object in the root directory of the examplebucket bucket. The ACL of the object is set to public read or public read/write. In this case, the URL of the object is `https://examplebucket.oss-accelerate.aliyuncs.com/myphoto.jpg` .

If the ACL of the *myphoto.jpg* object is set to private, you must add signature information to the object URL. In this case, the URL of the object is `https://examplebucket.oss-accelerate.aliyuncs.com/myphoto.jpg?OSSAccessKeyId=nz2pc56s936****&Expires=1141889120&Signature=vjbyPxybdZaNmGa%2ByT272YEAiv****` . For more information about how to add signatures to an object URL, see Add signatures to a URL.

For more information about transfer acceleration, see Transfer acceleration.

Use an endpoint that supports IPv6 to access OSS

Internet Protocol version 6 (IPv6) is the latest version of the Internet Protocol developed by the Internet Engineering Task Force (IETF) to replace Internet Protocol version 4 (IPv4). IPv6 can provide sufficient IP addresses to meet future requirements. You can access OSS by using dual-stack endpoints that support both IPv4 and IPv6.

You can use a dual-stack endpoint to access your bucket from your client. The DNS server resolves the endpoint to an IPv6 or an IPv4 address and returns the OSS server address based on the protocol used by the client. For example, the endpoint of the China (Hangzhou) region is `cn-hangzhou.oss.aliyuncs.com` . If you have a bucket named examplebucket in this region, you can use the same endpoint `https://examplebucket.cn-hangzhou.oss.aliyuncs.com` to access the bucket from your client by using both IPv4 and IPv6.

> **Notice** ECS instances that use virtual private cloud (VPC) networks or classic networks cannot use a dual-stack endpoint to access OSS.

You can use the following endpoints to access OSS over IPv6:

Region	Endpoint
China (Hangzhou)	cn-hangzhou.oss.aliyuncs.com
China (Shanghai)	cn-shanghai.oss.aliyuncs.com
China (Qingdao)	cn-qingdao.oss.aliyuncs.com
China (Beijing)	cn-beijing.oss.aliyuncs.com
China (Zhangjiakou)	cn-zhangjiakou.oss.aliyuncs.com
China (Hohhot)	cn-huhehaote.oss.aliyuncs.com
China (Ulanqab)	cn-wulanchabu.oss.aliyuncs.com
China (Shenzhen)	cn-shenzhen.oss.aliyuncs.com
China (Heyuan)	cn-heyuan.oss.aliyuncs.com
China (Guangzhou)	cn-guangzhou.oss.aliyuncs.com
China (Chengdu)	cn-chengdu.oss.aliyuncs.com
China (Hong Kong)	cn-hongkong.oss.aliyuncs.com
China East 1 Finance	cn-hangzhou-finance.oss.aliyuncs.com
China East 2 Finance	cn-shanghai-finance.oss.aliyuncs.com
China South 1 Finance	cn-shenzhen-finance.oss.aliyuncs.com

3.3. OSS internal endpoints and VIP ranges

To access an Object Storage Service (OSS) bucket over the internal network from an Elastic Compute Service (ECS) instance in another region or a device in an on-premises data center, you can use Cloud Enterprise Network (CEN), Express Connect, leased lines, or virtual private networks (VPNs) to connect to the internal network of the region where the OSS bucket is located. Then, you can configure a route that directs to the virtual IP address (VIP) ranges of the internal network. This topic describes the VIP ranges of the OSS internal network in each region.

Warning

- OSS divides the VIP ranges of the internal network in each region into fixed address ranges. When you configure routes based on regions, you must configure complete routes based on the VIP ranges described in the following table.
- When you access OSS over the internal network from an ECS instance, do not forbid access from VIP ranges described in the following table in the security group.

Region	Region ID	Endpoint for access over VPCs	VIP range
China (Hangzhou)	oss-cn-hangzhou	oss-cn-hangzhou-internal.aliyuncs.com	100.118.28.0/24 100.114.102.0/24 100.98.170.0/24 100.118.31.0/24
China (Shanghai)	oss-cn-shanghai	oss-cn-shanghai-internal.aliyuncs.com	100.98.35.0/24 100.98.110.0/24 100.98.169.0/24 100.118.102.0/24
China (Qingdao)	oss-cn-qingdao	oss-cn-qingdao-internal.aliyuncs.com	100.115.173.0/24 100.99.113.0/24 100.99.114.0/24 100.99.115.0/24
China (Beijing)	oss-cn-beijing	oss-cn-beijing-internal.aliyuncs.com	100.118.58.0/24 100.118.167.0/24 100.118.170.0/24 100.118.171.0/24 100.118.172.0/24 100.118.173.0/24
China (Zhangjiakou)	oss-cn-zhangjiakou	oss-cn-zhangjiakou-internal.aliyuncs.com	100.118.90.0/24 100.98.159.0/24 100.114.0.0/24 100.114.1.0/24

Region	Region ID	Endpoint for access over VPCs	VIP range
China (Hohhot)	oss-cn-huhehaote	oss-cn-huhehaote-internal.aliyuncs.com	• 100.118.195.0/24 • 100.99.110.0/24 • 100.99.111.0/24 • 100.99.112.0/24
China (Ulanqab)	oss-cn-wulanchabu	oss-cn-wulanchabu-internal.aliyuncs.com	• 100.114.11.0/24 • 100.114.12.0/24 • 100.114.100.0/24 • 100.118.214.0/24
China (Shenzhen)	oss-cn-shenzhen	oss-cn-shenzhen-internal.aliyuncs.com	• 100.118.78.0/24 • 100.118.203.0/24 • 100.118.204.0/24 • 100.118.217.0/24
China (Heyuan)	oss-cn-heyuan	oss-cn-heyuan-internal.aliyuncs.com	• 100.98.83.0/24 • 100.118.174.0/24
China (Guangzhou)	oss-cn-guangzhou	oss-cn-guangzhou-internal.aliyuncs.com	• 100.115.33.0/24 • 100.114.101.0/24
China (Chengdu)	oss-cn-chengdu	oss-cn-chengdu-internal.aliyuncs.com	• 100.115.155.0/24 • 100.99.107.0/24 • 100.99.108.0/24 • 100.99.109.0/24
China (Hong Kong)	oss-cn-hongkong	oss-cn-hongkong-internal.aliyuncs.com	• 100.115.61.0/24 • 100.99.103.0/24 • 100.99.104.0/24 • 100.99.106.0/24
US (Silicon Valley) *	oss-us-west-1	oss-us-west-1-internal.aliyuncs.com	Submit a ticket to obtain VIP rages.
US (Virginia) *	oss-us-east-1	oss-us-east-1-internal.aliyuncs.com	• 100.115.60.0/24 • 100.99.100.0/24 • 100.99.101.0/24 • 100.99.102.0/24
Singapore *	oss-ap-southeast-1	oss-ap-southeast-1-internal.aliyuncs.com	• 100.118.219.0/24 • 100.99.213.0/24 • 100.99.116.0/24 • 100.99.117.0/24
Australia (Sydney) *	oss-ap-southeast-2	oss-ap-southeast-2-internal.aliyuncs.com	Submit a ticket to obtain VIP ranges.
Malaysia (Kuala Lumpur) *	oss-ap-southeast-3	oss-ap-southeast-3-internal.aliyuncs.com	• 100.118.165.0/24 • 100.99.125.0/24 • 100.99.130.0/24 • 100.99.131.0/24
Indonesia (Jakarta) *	oss-ap-southeast-5	oss-ap-southeast-5-internal.aliyuncs.com	Submit a ticket to obtain VIP ranges.
Japan (Tokyo) *	oss-ap-northeast-1	oss-ap-northeast-1-internal.aliyuncs.com	Submit a ticket to obtain VIP ranges.
India (Mumbai) *	oss-ap-south-1	oss-ap-south-1-internal.aliyuncs.com	• 100.118.211.0/24 • 100.99.122.0/24 • 100.99.123.0/24 • 100.99.124.0/24
Germany (Frankfurt) *	oss-eu-central-1	oss-eu-central-1-internal.aliyuncs.com	100.115.154.0/24
UK (London)	oss-eu-west-1	oss-eu-west-1-internal.aliyuncs.com	Submit a ticket to obtain VIP ranges.
UAE (Dubai) *	oss-me-east-1	oss-me-east-1-internal.aliyuncs.com	Submit a ticket to obtain VIP ranges.
Philippines (Manila)	oss-ap-southeast-6	oss-ap-southeast-6-internal.aliyuncs.com	100.115.16.0/24

Note The name of a region marked with the * in this topic indicates that the name of the region may be different from that of the region on the website of OSS Pricing or the website of OSS Resource Plans. However, the two different names represent the same region. For more information, visit OSS Pricing or OSS Resource Plans.

3.4. Access to OSS resources from ECS instances by using an internal endpoint of OSS

If you access Object Storage Service (OSS) resources by using an internal endpoint of OSS, no fees are charged for the traffic generated. This topic describes how Elastic Compute Service (ECS) instances access OSS resources by using an internal endpoint of OSS.

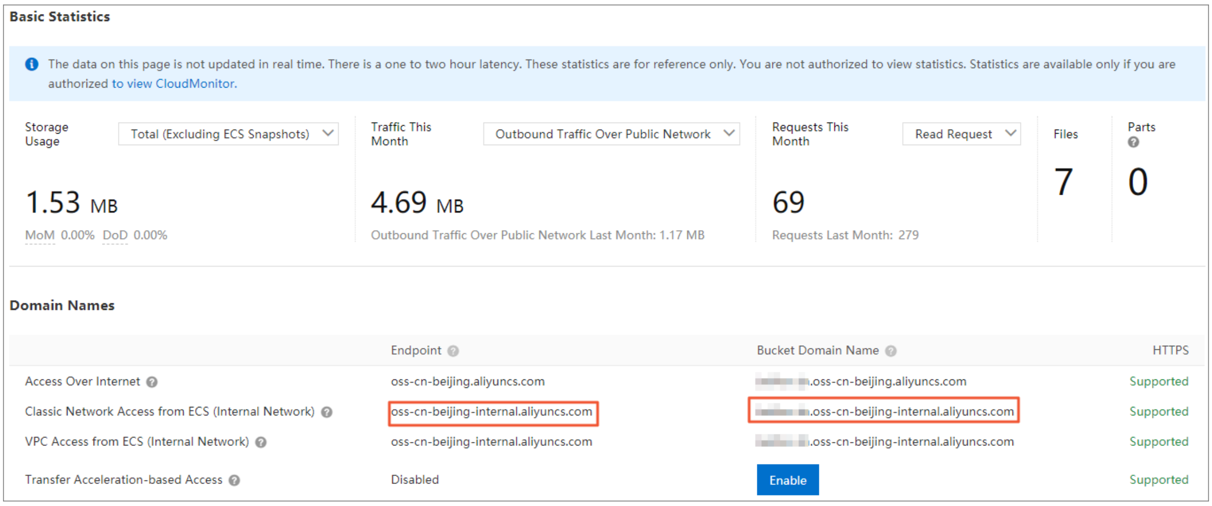
ECS instances can use the following methods to access OSS resources by using an internal endpoint of OSS:

- ECS instances deployed within the same region as an OSS bucket can use the internal endpoint of OSS to access the bucket resources that the ECS instances are authorized to access.
- ECS instances that are not in the same region as the bucket and Internet users can use the ECS reverse proxy to access the OSS resources by using the internal endpoint of OSS.

Obtain the internal endpoint of OSS

- Obtain from the OSS console

  Log on to the OSS console. On the Overview tab of the specified bucket, you can view the endpoints and bucket domain names in the **Domain Names** section. The following figure shows an example.



Basic Statistics

The data on this page is not updated in real time. There is a one to two hour latency. These statistics are for reference only. You are not authorized to view statistics. Statistics are available only if you are authorized to view CloudMonitor.

Storage Usage: Total (Excluding ECS Snapshots) 1.53 MB MoM 0.00% DoD 0.00%

Traffic This Month: Outbound Traffic Over Public Network 4.69 MB Outbound Traffic Over Public Network Last Month: 1.17 MB

Requests This Month: Read Request 69 Requests Last Month: 279

Files: 7

Parts: 0

Domain Names

	Endpoint	Bucket Domain Name	HTTPS
Access Over Internet	oss-cn-beijing.aliyuncs.com	[redacted].oss-cn-beijing.aliyuncs.com	Supported
Classic Network Access from ECS (Internal Network)	oss-cn-beijing-internal.aliyuncs.com	[redacted].oss-cn-beijing-internal.aliyuncs.com	Supported
VPC Access from ECS (Internal Network)	oss-cn-beijing-internal.aliyuncs.com	[redacted].oss-cn-beijing-internal.aliyuncs.com	Supported
Transfer Acceleration-based Access	Disabled	Enable	Supported

- Follow the fixed format

  OSS bucket domain names are in the following format: `BucketName.Endpoint` . In this format, `BucketName` indicates the name of your bucket. `Endpoint` indicates the endpoint used to access the region where your bucket is located. For more information, see OSS domain names.

Access from ECS instances within the same region

ECS instances deployed within the same region as an OSS bucket can use the internal network to access resources in the bucket.

- Use URLs

  You can use the internal endpoint of OSS to access OSS resources that you are authorized to access. For example, a bucket named test is located in the China (Hangzhou) region. The bucket contains an object named 1.jpg in the root directory of the bucket, and the access control list (ACL) of the object is public read. In this case, ECS instances in the China (Hangzhou) region can access this object by using `http://test.oss-cn-hangzhou-internal.aliyuncs.com/1.jpg` . You can embed the access URL of the object in your website and provide the URL to ECS users within the same region or to users who have connected to the internal network by using a leased line.

  Warning For data security reasons, we recommend that you do not set the ACL of your OSS resources to public read or public read/write. You can use bucket policies to authorize other users to access OSS resources. For more information, see Configure bucket policies to authorize other users to access OSS resources.

- Use ossbrowser

  When you set the parameters for logging on to ossbrowser, set Endpoint to the internal endpoint of OSS. For more information, see ossbrowser.

- Use ossutil

  When you set the parameters for logging on to ossutil, set Endpoint to the internal endpoint of OSS. For more information, see ossutil.

- Use SDKs

  When you initialize the client instance, set Endpoint to the internal endpoint of OSS.

- Java SDK

  ```
  String endpoint = "http://oss-cn-hangzhou-internal.aliyuncs.com";// The China (Hangzhou) region is used in the example.
  String accessKeyId = "<yourAccessKeyId>";
  String accessKeySecret = "<yourAccessKeySecret>";
  OSSClient client = new OSSClient(endpoint, accessKeyId, accessKeySecret);
  ```

  For more information, see Initialization.

- PHP SDK

  ```
  $accessKeyId = "<yourAccessKeyId>";
  $accessKeySecret = "<yourAccessKeySecret>";
  $endpoint = "<A data center endpoint that you have selected to access OSS, such as http://oss-cn-hangzhou-internal.aliyuncs.com>";
  ```

  For more information, see Initialization.

- Python SDK

  ```
  auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
  endpoint = 'http://oss-cn-hangzhou-internal.aliyuncs.com' # A data center endpoint that you have selected to access OSS. The China (Hangzhou) region is used in the example.
  bucket = oss2.Bucket(auth, endpoint, 'BucketName')
  ```

  For more information, see Initialization.

- .NET SDK

  ```
  const string accessKeyId = "<yourAccessKeyId>";
  const string accessKeySecret = "<yourAccessKeySecret>";
  const string endpoint = "http://oss-cn-hangzhou-internal.aliyuncs.com";
  var ossClient = new OssClient(endpoint, accessKeyId, accessKeySecret);
  ```

  For more information, see Initialization.

- C SDK

  ```
  ptions->config = oss_config_create(options->pool);
  aos_str_set(&options->config->endpoint, "http://oss-cn-hangzhou-internal.aliyuncs.com");
  aos_str_set(&options->config->access_key_id, "<yourAccessKeyId>");
  aos_str_set(&options->config->access_key_secret, "<yourAccessKeySecret>");
  options->config->is_cname = 0;
  options->ctl = aos_http_controller_create(options->pool, 0);
  ```

  For more information, see Initialization.

Access OSS resources by configuring a reverse proxy on ECS instances

ECS instances or Internet users in different regions cannot directly access OSS resources by using the internal endpoint of OSS. However, you can configure a reverse proxy on ECS instances to access OSS resources. Follow these steps:

1. Create an ECS instance with a public IP address in the same region as the OSS resources. For more information, see Create an ECS instance.
2. Configure the reverse proxy on the ECS instance. For more information, see Configure HTTPS for your own domain name in OSS through reverse proxy.
3. Configure OSS bucket policies to allow access from the internal IP address of the ECS instance. For more information, see Configure bucket policies to authorize other users to access OSS resources.

After you complete the preceding steps, users can access the OSS resources by using the public IP address of the ECS instance. Then, the ECS instance requests OSS resources over the internal network and returns the OSS resources to the users.

3.5. Choose an OSS region

This topic describes how to choose an appropriate region when you create a bucket in Object Storage Service (OSS).

Take note of the following items when you choose an OSS region:

- Location of your users
- Connection between Alibaba Cloud services
- Pricing
- Features

Location of your users

If you want users who access your resources in OSS to have a good user experience, you must consider the network latencies between the users and OSS. Geographical distance and the quality of communication links affect network latencies.

For example, users in the China (Hangzhou) region can run the **ping** command to test the network latencies that occur when the users access OSS resources in different regions.

The results show that latencies increase when the geographical distance between users and OSS increases. Therefore, choose a region that is geographically closer to your users.

Connection between Alibaba Cloud services

If you use OSS with other Alibaba Cloud services, we recommend that you choose the same region for OSS and the other services. When OSS and these services are in the same region, OSS can be accessed by using Virtual Private Cloud (VPC) endpoints. In this case, you are not charged traffic fees, and the access is quicker than over the Internet.

Pricing

The pricing of resource plans varies between regions. You can choose a region that has better pricing of resource plans.

Features

When a new OSS feature is released, public previews of the feature are launched in some regions. If you want to try out the new feature, you must create a bucket in one of these specified regions. For information about OSS product updates, see Release notes.

4.Storage classes

4.1. Overview

Object Storage Service (OSS) provides the following storage classes to cover a variety of data storage scenarios from hot data storage to cold data storage: Standard, Infrequent Access (IA), Archive, and Cold Archive.

> ⓘ **Note** For more information about the pricing of each storage class, see Object Storage Service Pricing. For more information about the billing method for each storage class, see Storage fees.

Standard

OSS provides highly reliable, highly available, and high-performance storage services for Standard objects. Frequent data access is supported. Standard storage is suitable for storing images for social networking and sharing applications and data for audio and video applications, large websites, and big data analytics. Standard storage supports the following data redundancy storage mechanisms:

- Standard locally redundant storage (LRS)

  If you use Standard LRS, OSS stores the copies of each object on multiple devices of different facilities in the same zone. This way, OSS ensures data durability and availability even if hardware failures occur.

- Standard zone-redundant storage (ZRS)

  Standard ZRS uses the multi-zone mechanism to distribute user data across three zones within the same region. Even if one zone becomes unavailable, your data will still be accessible.

IA

OSS provides high-durability storage services for IA objects at prices lower than Standard. Objects of the IA storage class have a minimum storage period of 30 days and a minimum billable size of 64 KB. You can access objects of the IA storage class in a real-time manner, and you are charged data retrieval fees. IA storage applies to scenarios where stored data is infrequently accessed, such as once or twice a month. IA storage supports the following data redundancy storage mechanisms:

- IA LRS

  If you use IA LRS, OSS stores the copies of each object on multiple devices of different facilities in the same zone. This way, OSS ensures data durability and availability even if hardware failures occur.

- IA ZRS

  IA ZRS uses the multi-zone mechanism to distribute user data across three zones within the same region. Even if one zone becomes unavailable, your data will still be accessible.

Archive

OSS provides high-durability storage services for Archive objects at prices lower than Standard and IA. Objects of the Archive storage class have a minimum storage period of 60 days and a minimum billable size of 64 KB. You must restore an object of the Archive storage class before you can access it. The restoration takes about a minute, and you are charged the data retrieval fees. Archive storage is suitable for data that needs to be stored for a long period, such as archival data, medical images, scientific materials, and video footage.

Cold Archive

OSS provides high-durability storage services for Cold Archive objects at prices lower than Standard, IA, and Archive. Objects of the Cold Archive storage class have a minimum storage period of 180 days and a minimum billable size of 64 KB. You must restore an object of the Cold Archive storage class before you can access it. The time required to restore a Cold Archive object depends on the object size and the restore mode. You are charged for the data retrieval fees when you restore a Cold Archive object. Cold Archive storage is suitable for storing extremely cold data over an ultra-long period of time. Such data includes data that must be retained for an extended period of time due to compliance requirements, raw data that is accumulated over an extended period of time in the big data and AI fields, retained media resources in the film and television industries, and archived videos from the online education industry.

> ⓘ **Note** Cold Archive is supported in the following regions: China (Qingdao), China (Beijing), China (Zhangjiakou), China (Hangzhou), China (Shanghai), China (Shenzhen), China (Chengdu), China (Ulanqab), China (Hong Kong), Australia (Sydney), Singapore (Singapore), US (Silicon Valley), Germany (Frankfurt), Malaysia (Kuala Lumpur), Indonesia (Jakarta), India (Mumbai), and UAE (Dubai). To apply for a trial, contact technical support.

Comparison of storage classes

Index	Standard LRS	Standard ZRS	IA LRS	IA ZRS	Archive	Cold Archive
Data durability (designed for)	99.999999999% (eleven 9's)	99.9999999999% (twelve 9's)	99.999999999% (eleven 9's)	99.9999999999% (twelve 9's)	99.999999999% (eleven 9's)	99.999999999% (eleven 9's)
Service availability	99.99%	99.995%	99.00%	99.50%	99.00% (restored data)	99.00% (restored data)
Minimum billable size	None	None	64 KB	64 KB	64 KB	64 KB
Minimum storage period	None	None	30 days	30 days	60 days	180 days
Data retrieval fees	None	None	Based on the size of retrieved data. Unit: GB.	Based on the size of retrieved data. Unit: GB.	Based on the size of restored data. Unit: GB.	Based on the size of restored data and the data retrieval capability that is selected. Unit: GB.

Index	Standard LRS	Standard ZRS	IA LRS	IA ZRS	Archive	Cold Archive
Data access	Real-time access and low latency within milliseconds.	Real-time access and low latency within milliseconds.	Real-time access and low latency within milliseconds.	Real-time access and low latency within milliseconds.	Supported after data is restored. It takes a minute to restore data.	You must restore a Cold Archive object before you can read it. The time required to restore a Cold Archive object to the readable state is determined based on the restoration priority of the object: • Expedited: The object is restored within 1 hour. • Standard: The object is restored within 2 to 5 hours. • Bulk: The object is restored within 5 to 12 hours.
Image Processing (IMG)	Supported	Supported	Supported	Supported	Supported after data is restored.	Supported after data is restored.
Scenario	Suitable for storing images for social networking and sharing applications and data for audio and video applications, large websites, and big data analytics. Examples: program download and mobile applications.	Suitable for storing images for social networking and sharing applications and data for audio and video applications, large websites, and big data analytics. Standard ZRS applies to scenarios where higher durability and availability are required. Examples: important documents of enterprises and sensitive information.	Suitable for storing data that is infrequently accessed, such as once or twice each month. Examples: hot backup data and surveillance video data.	Suitable for storing data that is infrequently accessed, such as once or twice each month. IA ZRS applies to scenarios where higher durability and availability are required. Examples: enterprise business data and recent medical records.	Suitable for storing data that you want to store for a long period of time. Examples: archival data, medical images, scientific materials, and video footage.	Suitable for storing extremely cold data that you want to store for an ultra-long period of time. Examples: data that must be retained for an extended period of time due to compliance requirements, raw data that is accumulated over an extended period of time in the big data and AI fields, retained media resources in the film and television industries, and archived videos from the online education industry.

Note OSS charges data retrieval fees based on the amount of data read from the underlying distributed storage system. Data transmitted over the Internet is billed as outbound traffic.

4.2. Convert storage classes

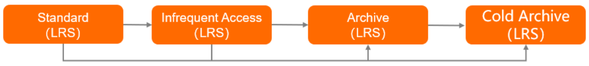
Object Storage Service (OSS) provides the following storage classes: Standard, Infrequent Access (IA), Archive, and Cold Archive. You can configure lifecycle rules or call the CopyObject operation to convert the storage class of an object based on your business requirements.

Note For more information about the four storage classes, see Overview.

Configure lifecycle rules to automatically convert the storage class of an object

You can configure lifecycle rules to allow OSS to automatically convert the storage classes of objects.

- Rules for storage class conversion based on the last modified date of objects
  - Locally redundant storage (LRS)

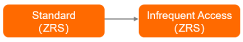


Locally redundant storage

The storage class of LRS objects can be converted based on the following rules:

- Conversions from Standard LRS to IA LRS, Archive LRS, or Cold Archive LRS.
- Conversions from IA LRS to Archive LRS or Cold Archive LRS.
- Conversions from Archive LRS to Cold Archive LRS.

  - Zone-redundant storage (ZRS)



Zone-redundant storage

The storage class of ZRS objects can be converted based on the rule: Only conversions from Standard ZRS to IA ZRS are supported.

For more information, see Lifecycle rules based on the last modified time.

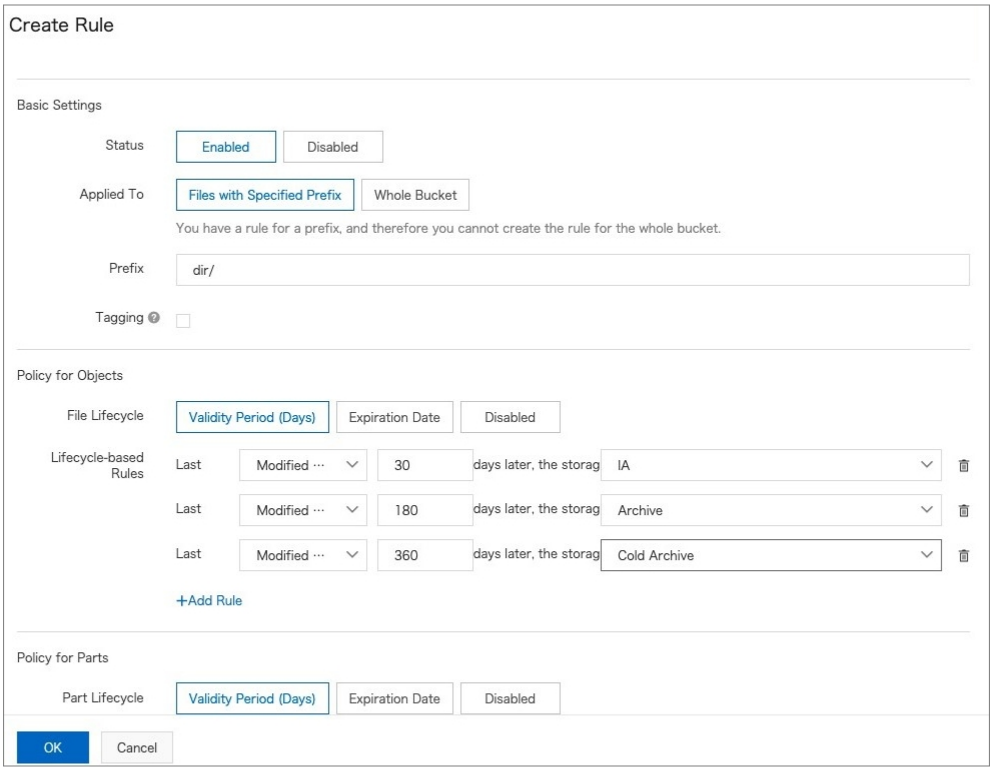
- Rules for storage class conversion based on the last access date of objects

  You can configure a lifecycle rule that automatically converts Standard objects to IA objects a specific number of days after the last access date of the objects. After the objects are converted, you can specify whether to convert the objects back to Standard objects or remain IA objects when the objects are accessed. For more information, see Lifecycle rules based on the last access time.

- Conversion examples

  For example, you want to configure a lifecycle rule for an LRS bucket so that objects whose names contain a specified prefix can be converted to a specified storage class a specific number of days after the last modified date of the objects.

  - Objects are converted to IA objects after the objects are stored for 30 days.
  - Objects are converted to Archive objects after the objects are stored for 180 days.
  - Objects are converted to Cold Archive objects after the objects are stored for 360 days.



Create Rule

Basic Settings

Status: [✓ Enabled] Disabled

Applied To: [✓ Files with Specified Prefix] Whole Bucket

You have a rule for a prefix, and therefore you cannot create the rule for the whole bucket.

Prefix: dir/

Tagging □

Policy for Objects

File Lifecycle: [✓ Validity Period (Days)] Expiration Date Disabled

Lifecycle-based Rules:

Last Modified ··· 30 days later, the storag IA

Last Modified ··· 180 days later, the storag Archive

Last Modified ··· 360 days later, the storag Cold Archive

+Add Rule

Policy for Parts

Part Lifecycle: [✓ Validity Period (Days)] Expiration Date Disabled

When you configure a lifecycle rule that converts objects to IA objects, Archive objects, and Cold Archive objects after a specific number of days, the number of days specified must meet the following requirements:

Days for conversion to IA objects < Days for conversion to Archive objects < Days for conversion to Cold Archive objects

- Implementation methods of configuring lifecycle rules

Implementation method	Description
Console	A user-friendly and intuitive web application.
ossutil	A high-performance command-line tool.
Java SDK	SDK demos for a variety of programming languages.
Python SDK	
PHP SDK	
Go SDK	
C SDK	
.NET SDK	
Node.js SDK	
Ruby SDK	

Call CopyObject to manually convert storage classes

You can call the CopyObject operation to convert the storage class of an object by overwriting the object. If the storage class of the converted object is IA, Archive, or Cold Archive and the object is stored for less than the specified period of time, you are charged based on the minimum storage period. For more information, see Storage fees.

- Rules for storage class conversion by calling CopyObject

- LRS

  Conversions between storage classes are supported.

  > Note Before you can convert the storage class of an Archive object or a Cold Archive object, you must restore the object. For more information about how to restore an object, see Restore objects.

- ZRS

  Only conversions between Standard ZRS and IA ZRS are supported.

- Implementation methods of storage class conversion by calling CopyObject

Implementation method	Description
Console	A user-friendly and intuitive web application.
ossutil	A high-performance command-line tool.
Java SDK	SDK demos for a variety of programming languages.
Python SDK	
Go SDK	
C++ SDK	

Usage notes

After you convert the storage class of an object to IA, Archive, or Cold Archive, take note of the following items:

- Minimum billable size

  You are charged the minimum billable size of 64 KB for objects that are smaller than 64 KB.

- Minimum storage period

  The minimum storage period that you can set for an IA Object is 30 days. The minimum storage period that you can set for an Archive object is 60 days. The minimum storage period that you can set for a Cold Archive object is 180 days. If an object is stored for a period less than the minimum storage period, you are still charged for the minimum storage period.

  - Automatic conversion triggered by lifecycle rules

    If you configure lifecycle rules to automatically convert the storage class of an object, OSS does not recalculate the retention period when the storage class of the object changes. Example: An object named *a.txt* is a Standard object. After the object is stored in OSS for 10 days, its storage class is converted to IA based on lifecycle rules. Then, the object must be stored as an IA object for another 20 days to meet the minimum storage period of 30 days. For more information, see FAQ.

  - Manual conversion

    If you manually convert the storage class of an object, OSS recalculates the retention period of the object. Example: An object named *a.txt* is a Standard object. After the object is stored in OSS for 10 days, its storage class is manually converted to IA. Then, the retention period of the object as an IA object is reset to 0, and the object must be stored for another 30 days to meet the minimum storage period of 30 days.

- Restoration time

  It takes a period of time to restore Archive or Cold Archive objects to the readable state. If your business requires your objects to be read in real time, we recommend that you do not convert the storage classes of your objects to Archive or Cold Archive.

- Data retrieval fees

  When you access IA objects, you are charged additional data retrieval fees based on the amount of accessed data. You are charged data restoration fees when you restore Archive or Cold Archive objects. Data restoration and outbound traffic are two separate billable items. If an object is accessed more than once per month on average, the storage cost of the object may be higher if you convert the storage class of the object from Standard to IA, Archive, or Cold Archive.

- Temporary storage fees

  When you restore a Cold Archive object, a Standard replica of the object is generated for temporary access. You are charged the temporary storage fees of the replica for its duration as a Standard object before the restoration period ends.

5.Buckets

5.1. Overview

Before you upload data such as documents, images, and audio or video files to Object Storage Service (OSS), you must create a bucket in an OSS region. OSS does not impose limits on the number of objects that you can upload to a bucket.

Background information

Buckets and objects are OSS resources. OSS provides API operations to manage these resources. For example, you can call API operations to create a bucket and upload objects to the bucket. You can also perform these operations in the OSS console. When you perform operations in the OSS console, OSS API operations are called to sent requests to OSS.

OSS does not use a hierarchical structure for objects, but instead uses a flat structure. All elements are stored as objects in buckets. However, OSS supports folders as a concept to group objects and simplify management. The name of a bucket is unique within OSS and cannot be modified after the bucket is created. For more information, see bucket naming conventions in Terms.

Related operations

The following table lists supported operations related to buckets. For more information about operations related to objects, see Overview.

Operation	Description
Create buckets	Before you can upload objects to OSS, you must create a bucket. The attributes of a bucket include the region, access control list (ACL), and other metadata. When you create a bucket, you must select a region for the bucket based on your requirements on latency, costs, and compliance. For more information about the regions supported by OSS, see Regions and endpoints.
Configure bucket ACLs	You can configure the ACL of a bucket when you create the bucket or modify the ACL of a created bucket. Only the owner of a bucket can configure or modify the ACL of the bucket.
Query the region of a bucket	You can call the GetBucketLocation operation to obtain the region of a bucket, which indicates the data center in which the bucket is located.
List buckets	You can specify different filters to list all buckets or buckets whose names contain a specified prefix in a region.
Configure bucket inventory	You can use the bucket inventory feature to export the information about specific objects in a bucket, such as the number, sizes, storage classes, and encryption status of the objects. Compared with the GetBucket (ListObjects) operation, we recommend that you use the bucket inventory feature to list a large number of objects.
Configure pay-by-requester	When pay-by-requester is enabled for a bucket, requesters pay the request and traffic fees that are incurred when the requesters access objects in the bucket. The bucket owner is still charged for the storage fees of the objects. You can enable pay-by-requester to share your data in OSS without having to pay for additional fees on your own.
Map custom domain names	After you upload objects to a bucket, OSS automatically generates URLs that include the public endpoint of the bucket for the uploaded objects. You can use these URLs to access the objects. If you want to access the objects by using custom domain names, you must map the custom domain names to the bucket in which the objects are stored and add CNAME records for the custom domain names.
Configure transfer acceleration	OSS uses data centers distributed around the globe to implement transfer acceleration. When a request is sent to your bucket, it is resolved and routed to the data center where the bucket resides over the most optimal network path and protocol. The transfer acceleration feature provides an optimized end-to-end acceleration solution to access OSS over the Internet.
Configure CORS	Cross-origin resource sharing (CORS) is a standard cross-origin solution provided by HTML5 to allow web application servers to control cross-origin access, which ensures the security of data transmission across origins.
Configure bucket tagging	OSS allows you to configure bucket tags to classify and manage buckets. For example, you can list buckets that have specific tags and configure ACL for buckets that have specific tags.
Configure event notification	You can configure event notifications for specific objects in a bucket. When the specified events occur on the objects, you are notified as soon as possible.
Configure lifecycle rules	After you configure lifecycle rules for a bucket, OSS converts the storage class of objects in the bucket to Infrequent Access (IA), Archive, Cold Archive, or deletes expired objects and parts on a regular basis to save storage costs.
Configure real-time log query	After you configure real-time log query for a bucket, you can track requests that are sent to access the bucket. This feature allows you to collect access statistics, audit access to OSS, track exceptions, and troubleshoot problems. Real-time log query can improve your efficiency and help you make better decisions based on real-time data.

Operation	Description
Configure retention policies	You can configure a time-based retention policy for a bucket. After the retention policy is locked, you can upload objects to the objects or read objects in the bucket. However, objects in the bucket and the retention policy cannot be deleted within the retention period. You can delete objects in the bucket only after the retention period expires.
Bucket Policy	You can configure bucket policies to authorize other users to access the specified OSS resources.
Configure data replication	You can replicate objects from a source bucket to a destination bucket. The source bucket and the destination bucket can be in the same region or different regions. • Same-region replication (SRR) allows you to replicate objects across buckets within the same region in an automatic and asynchronous (near real-time) manner. Operations such as the creation, overwriting, and deletion of objects are synchronized from the source bucket to the destination bucket. • Cross-region replication (CRR) allows you to replicate objects across buckets in different regions in an automatic and asynchronous (near real-time) manner. Operations such as the creation, overwriting, and deletion of objects are synchronized from the source bucket to the destination bucket.
Configure versioning	OSS allows you to configure versioning for a bucket to protect objects stored in the bucket. After you enable versioning for a bucket, objects that are overwritten or deleted in the bucket are saved as previous versions. You can use versioning to recover a previous version of an object that is accidentally overwritten or deleted.
Configure static website hosting	Static websites are websites in which all web pages consist only of static content, including scripts such as JavaScript code running on the client. You can use the static website hosting feature to host your static website in an OSS bucket and use the endpoint of the bucket to access the website.
Delete buckets	You can delete a bucket that you no longer need.

5.2. Bucket naming conventions

A bucket is a container that is used to store objects in Object Storage Service (OSS). Every object is contained in a bucket. You can configure a variety of bucket attributes such as the region, access control list (ACL), and storage class. You can create buckets of different storage classes to store data.

Naming conventions

The maximum number of buckets that can be created by using an Alibaba Cloud account within a region is 100. After a bucket is created, its name cannot be modified. OSS supports the following bucket naming conventions:

- The name of a bucket must be unique in OSS in an Alibaba Cloud account.
- The name can contain only lowercase letters, digits, and hyphens (-).
- The name must start and end with a lowercase letter or a digit.
- The name must be 3 to 63 characters in length.

Examples

The following examples of bucket names are valid:

- examplebucket1
- test-bucket-2021
- aliyun-oss-bucket

The following examples show invalid bucket names and the reasons why the names are invalid:

- Examplebucket1 (Uppercase letters are included.)
- test_bucket_2021 (Underscores (_) are included.)
- aliyun-oss-bucket- (The name ends with a hyphen (-).)

5.3. Create buckets

Before you can upload objects to Object Storage Service (OSS), you must create a bucket. You can configure various attributes of a bucket, including the region, access control list (ACL), and storage class. You can create buckets of different storage classes to store your data.

Usage notes

- When you create a bucket, you are charged only for the storage of objects in the bucket and the traffic generated when the objects are accessed. For more information, see Overview.
- The capacity of the bucket is scalable. You do not need to purchase the capacity before you use the bucket.

Limits

- You can use an Alibaba Cloud account to create up to 100 buckets in the same region.
- A bucket name must be globally unique within OSS. For more information about the naming conventions of buckets, see Bucket naming conventions.
- After a bucket is created, you cannot modify its name, region, storage class or redundancy type.
- OSS does not impose limits on the capacity of a bucket.

Use the OSS console

1. Log on to the OSS console.

2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click **Create Bucket**.
3. In the **Create Bucket** panel, configure parameters. The following table describes the parameters.

Parameter	Required	Description
Bucket Name	Yes	Specify the name of the bucket The name must meet the following requirements: ○ The bucket name must be globally unique in Alibaba Cloud OSS. ○ The name can contain only lowercase letters, digits, and hyphens (-). ○ The name must start and end with a lowercase letter or a digit. ○ The name must be 3 to 63 characters in length.
Region	Yes	Select the region for the bucket. To access OSS from an Elastic Compute Service (ECS) instance over the internal network, select the region in which the ECS instance is located. For more information, see OSS domain names. **Note** You must complete real-name registration on the Real-name Registration page before you create a bucket in a region within the Chinese mainland.
Storage Class	Yes	Select the storage class for the bucket. ○ **Standard**: provides highly reliable, highly available, and high-performance object storage services that can handle frequent data access. Standard storage is ideal for storing images for social networking and sharing applications and store data for audio and video applications, large websites, and big data analysis. ○ **IA**: provides high-durability storage services at prices lower than Standard. Objects of the Infrequent Access (IA) storage class have a minimum storage period of 30 days and a minimum billable size of 64 KB. You can access objects of the IA storage class in real time. However, you are charged data retrieval fees when you access IA objects. IA storage is suitable for data that is infrequently accessed, such as once or twice a month. ○ **Archive**: provides high-durability storage services at prices lower than Standard and IA. Objects of the Archive storage class have a minimum storage period of 60 days and a minimum billable size of 64 KB. You must restore an Archive object before you can access it. The restoration takes approximately one minute. You are charged for data retrieval fees. Archive storage is ideal for data that needs to be stored for a long period, such as archival data, medical images, scientific materials, and video footage. ○ **Cold Archive**: provides high-durability storage services at prices lower than Standard, IA, and Archive. Objects of the Cold Archive storage class have a minimum storage period of 180 days and a minimum billable size of 64 KB. You must restore an object of the Cold Archive storage class before you can access it. The amount of time required to restore a Cold Archive object depends on the object size and the restoration mode. You are charged for data retrieval fees when you restore Cold Archive objects. Cold Archive storage is ideal for storing cold data over an ultra-long period of time. Such data includes data that must be retained for an extended period of time due to compliance requirements, raw data that is accumulated over an extended period of time in the big data and AI fields, retained media resources in the film and television industries, and archived videos from the online education industry. **Note** Cold Archive is supported in the following regions: China (Qingdao), China (Beijing), China (Zhangjiakou), China (Hangzhou), China (Shanghai), China (Shenzhen), China (Chengdu), China (Ulanqab), China (Hong Kong), Australia (Sydney), Singapore (Singapore), US (Silicon Valley), Germany (Frankfurt), Malaysia (Kuala Lumpur), Indonesia (Jakarta), India (Mumbai), and UAE (Dubai). To apply for a trial, contact technical support. For more information about storage classes, see Overview.
OSS-HDFS	No	If you want to access OSS by using JindoSDK to build a data lake, enable OSS-HDFS first. **Notice** ○ OSS-HDFS is supported only in the China (Hangzhou), China (Shanghai), China (Shenzhen), China (Beijing), and China (Zhangjiakou) regions.To apply for a trial, contact technical support. OSS-HDFS cannot be disabled after it is enabled. Exercise caution when you enable OSS-HDFS. ○ OSS-HDFS does not support Archive buckets or Cold Archive buckets.
Zone-redundant Storage	No	Specify the redundancy type of the bucket. ○ Enable: After this feature is enabled, OSS data is stored in zone-redundant storage (ZRS) mode. ZRS uses the multi-zone mechanism to distribute user data across three zones within the same region. Even if one zone becomes unavailable due to failures such as power outages and fires, the data is still accessible. **Notice** ZRS is supported in the following regions: China (Shenzhen), China (Beijing), China (Hangzhou), China (Shanghai), China (Hong Kong), Singapore (Singapore), and Indonesia (Jakarta). Extra fees are charged for ZRS. This feature cannot be disabled after it is enabled. Exercise caution when you enable this feature. For more information about ZRS, see ZRS. ○ Disable: After ZRS is disabled, the redundancy type of the objects in the bucket is locally redundant storage (LRS). LRS stores the copies of each object across different devices within the same zone. This way, OSS ensures data reliability and availability if two storage devices are damaged at the same time.

Parameter	Required	Description
Versioning	No	Select whether to enable versioning. ○ **Enable**: If versioning is enabled for a bucket, an object that is overwritten or deleted is saved as a previous version of the object. Versioning allows you to restore objects in a bucket to a previous version, and protects your data from being accidentally overwritten or deleted. For more information, see Overview. ○ **Disable**: If versioning is disabled for a bucket, the overwritten or deleted data is not saved.
Access Control List (ACL)	Yes	Select the bucket ACL. ○ **Private**: Only the bucket owner can perform read and write operations on objects in the bucket. Other users cannot access the objects in the bucket. ○ **Public Read**: Only the bucket owner can perform write operations on objects in the bucket. Other users, including anonymous users, can perform only read operations on the objects in the bucket. **Warning** All users on the Internet can access objects in the bucket. This may result in unexpected access to the data in your bucket and unexpectedly high costs. Exercise caution when you set your bucket ACL to Public Read. ○ **Public Read/Write**: All users, including anonymous users, can perform read and write operations on the objects in the bucket. **Warning** All users on the Internet can access objects in the bucket and write data to the bucket. This may result in unexpected access to the data in your bucket and unexpectedly high fees. If a user uploads prohibited data or information, your legitimate interests and rights may be infringed. Therefore, we recommend that you do not set your bucket ACL to Public Read/Write except in special cases.
Encryption Method	No	Select whether to enable server-side encryption for the bucket. ○ **Encryption Method**: Select an encryption method for the bucket. ■ **None**: Server-side encryption is disabled. ■ **OSS-Managed**: Keys managed by OSS are used to encrypt objects in the bucket. OSS uses data keys to encrypt objects. In addition, OSS uses regularly rotated master keys to encrypt data keys. ■ **KMS**: The default CMK stored in KMS or the specified CMK ID is used to encrypt and decrypt data. Before you use SSE-KMS, you must activate KMS. For more information, see activate KMS. ○ **Encryption Algorithm**:Only 256-bit Advanced Encryption Standard (AES-256) is supported. ○ **CMK**: You can set this parameter if you select **KMS** in the **Encryption Method** section. You can configure the following parameters for a CMK: ■ **alias/acs/oss**: The default CMK stored in KMS is used to encrypt different objects and decrypt the objects when they are downloaded. ■ CMK ID: The keys generated by a specified CMK are used to encrypt different objects, and the specified CMK ID is recorded in the metadata of the encrypted object. Objects are decrypted when they are downloaded by users who are granted decryption permissions. Before you specify a CMK ID, you must create a normal key or an external key in the same region as the bucket in the KMS console For more information, see Import key material.
Real-time Log Query	No	If you want to query OSS access logs of the last seven days free of charge, click **Enable**. For more information about real-time log query, see Real-time log query. If you do not need to query real-time logs, keep the default setting, which is **Disable**.
Scheduled Backup	No	If you want to back up your OSS data on a regular basis, click **Enable**. OSS automatically creates a backup plan to back up data by using Hybrid Backup Recovery (HBR) once a day. The generated backup objects are stored for one week. **Notice** If HBR is not activated or HBR is not authorized to access OSS, scheduled backup plans cannot be created. For more information, see Configure scheduled backup. If you do not need to back up your OSS data on a regular basis, keep the default setting, which is **Disable**.
Hierarchical Namespace	No	If you want to rename a directory or an object, enable the hierarchical namespace feature. **Notice** You can enable the hierarchical namespace feature for a bucket only when you create the bucket. The hierarchical namespace feature cannot be disabled after it is enabled for a bucket. After you enable this feature for a bucket, some OSS features are no longer supported for the bucket. For more information about a list of features that are not supported for a bucket for which the hierarchical namespace feature is enabled, see Hierarchical namespace.

4. Click **OK**.

Use ossbrowser

ossbrowser supports the same operations related to buckets as the OSS console. You can follow the on-screen instructions in ossbrowser to create a bucket. For more information about how to use ossbrowser, see Use ossbrowser.

Use OSS SDKs

The following code provides examples on how to create a bucket by using OSS SDKs for common programming languages. For more information about how to create a bucket by using OSS SDKs for other programming languages, see Overview.

[illegible]

```
// Set yourEndpoint to the endpoint of the region in which the bucket is located. For example, if the bucket is located in the China (Hangzhou) regio
n, set yourEndpoint to https://oss-cn-hangzhou.aliyuncs.com.
String endpoint = "yourEndpoint";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API op
erations. We recommend that you use a RAM user to call API operations or perform routine operations and maintenance. To create a RAM user, log on to
the RAM console.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Create a CreateBucketRequest object.
CreateBucketRequest createBucketRequest = new CreateBucketRequest("examplebucket");
// The following code provides an example on how to specify the storage class, ACL, and redundancy type of the bucket that you want to create.
// In this example, the storage class of the bucket is set to Standard.
/createBucketRequest.setStorageClass(StorageClass.Standard);
// By default, the redundancy type of a bucket is set to DataRedundancyType.LRS, which indicates LRS. To set the redundancy type to zone-redundant st
orage (ZRS), set the value to DataRedundancyType.ZRS.
//createBucketRequest.setDataRedundancyType(DataRedundancyType.ZRS);
// Set the ACL of the bucket to public-read. The default value is private.
//createBucketRequest.setCannedACL(CannedAccessControlList.PublicRead);
// Create the bucket.
ossClient.createBucket(createBucketRequest);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Use ossutil

For more information about how to create a bucket by using ossutil, see mb (create buckets).

Use the RESTful API

If your program requires high customization, you can directly initiate a RESTful API request. In this case, you need to manually write code to calculate the signature. For more information, see PutBucket.

References

- Object management
  - Upload objects

    After you create a bucket, you can upload objects to the bucket. For more information about how to upload objects, see Simple upload.
  - Download objects

    After you upload objects to a bucket, you can download the objects to the default download path of your browser or the specified local path. For more information about how to download objects, see Simple download.
  - Share objects

    You can share the URLs of uploaded objects with third parties for downloads or previews. For more information about how to share objects, see Share objects.
- Access control

  By default, the ACL of OSS resources is private to ensure data security. Only the owners of the resources and authorized users can access these resources. OSS allows you to configure a variety of policies to grant third-party users specific permissions to access or use your OSS resources. For more information about ACLs, see Overview.

5.4. Query the region information of a bucket

You can use the GetBucketLocation operation of OSS to query the location information about the region, which is the data center, where a bucket is located.

> Note For more information about the GetBucketLocation operation, see GetBucketLocation.

The returned Location field indicates the region where the bucket is located. For example, a bucket is located in the China (Hangzhou) region, and the value of the returned Location field is `oss-cn-hangzhou`. For more information about regions, see Regions and endpoints.

Implementation modes

Implementation mode	Description
Console	A web application that displays the region of a bucket on the bucket overview page
ossbrowser	An easy-to-operate graphical tool
ossutil	A high-performance command-line tool
Java SDK	SDK demos for various programming languages
Python SDK	
PHP SDK	
Go SDK	

Implementation mode	Description
C SDK	
.NET SDK	
Node.js SDK	

5.5. List buckets

Buckets are listed in alphabetical order in Object Storage Service (OSS). You can list all buckets that belong to the current Alibaba Cloud account, buckets whose names contain a specified prefix, and a specified number of buckets.

List buckets that meet specific conditions

You can specify the prefix, marker, and max-keys parameters to list buckets that meet specific conditions.

Parameter	Description
prefix	The prefix that must be contained in the names of returned buckets. If you do not specify this parameter, all buckets are returned.
marker	The name of the bucket from which the list operation begins. Buckets whose names are alphabetically after the name are returned. If you do not specify this parameter, all buckets are returned.
max-keys	The maximum number of buckets to return. Valid values: 1 to 1000. Default value: 100

Limits

Accelerate endpoints cannot be used to list buckets. Transfer acceleration is applicable only to third-level domains that contain a specific bucket name, such as *https://BucketName.oss-accelerate.aliyuncs.com*. However, the domains of requests that are used to list buckets do not contain the information about bucket names, such as *https://oss-cn-hangzhou.aliyuncs.com*.

Use the OSS console

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**.

   By default, all buckets that belong to the current Alibaba Cloud account are displayed on the Buckets page. If you want to query the number of buckets and attributes of the buckets, in the upper-right corner, click the Export CSV icon.

Use ossbrowser

ossbrowser supports the same operations related to buckets as the OSS console. You can follow the on-screen instructions in ossbrowser to list buckets. For more information about how to use ossbrowser, see Use ossbrowser.

Use OSS SDKs

The following code provides examples on how to list buckets by using OSS SDKs for common programming languages. For more information about how to list buckets by using OSS SDKs for other programming languages, see Overview.

[illegible]

```
// Set yourEndpoint to the endpoint of the region in which the bucket is located. For example, if the bucket is located in the China (Hangzhou) regio
n, set the endpoint to https://oss-cn-hangzhou.aliyuncs.com.
String endpoint = "yourEndpoint";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API op
erations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List all buckets that belong to the current account.
List<Bucket> buckets = ossClient.listBuckets();
for (Bucket bucket : buckets) {
    System.out.println(" - " + bucket.getName());
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Use ossutil

For more information about how to list buckets by using ossutil, see List buckets.

Use the RESTful API

If your program requires more custom options to list buckets, you can call RESTful API operations. In this case, you need to manually write code to calculate the signature. For more information, see GetService (ListBuckets).

5.6. Hierarchical namespace

Object Storage Service (OSS) provides the hierarchical namespace feature for you to manage objects in a multi-level hierarchical structure based on directories. This feature allows you to rename a directory or an object by performing a single atomic operation. This way, you do not need to list or process all objects whose names contain the same prefix. Traditionally, OSS uses a flat namespace to store objects in buckets and simulates directories by using objects whose names end with a forward slash (/). Compared with this method, the hierarchical namespace feature greatly improves the performance of directory management.

Benefits

- Atomic operations on directories

  If a bucket uses a flat namespace in which real directories are not supported, applications may need to process millions of objects to complete a task at the directory level. In contrast, if the hierarchical namespace feature is enabled for a bucket, applications can update a parent directory in a single atomic operation to complete multiple directory-level tasks at the same time.

- Optimized performance

  Compared with a flat namespace, a hierarchical namespace eliminates the need to replicate or convert data before the data is analyzed. This improves the performance of directory management. The hierarchical namespace feature is especially important for big data analytics frameworks. For example, Hive or Spark writes output results to a temporary directory during task execution and renames the directory after the task is complete. In a flat namespace, the time that is consumed to rename the directory is usually longer than the time that is consumed to perform the task.

Implementation methods

You can enable the hierarchical namespace feature for a bucket only when you create the bucket. The hierarchical namespace feature cannot be disabled after it is enabled for a bucket. For more information about how to enable the hierarchical namespace feature, see Create buckets.

The following table describes the methods that you can use to create, rename, or delete directories in a bucket for which the hierarchical namespace feature is enabled.

Implementation method	Description
In the OSS console • Create directories • Rename a directory or object • Delete directories	A user-friendly and intuitive web application
Java SDK	Simple and easy-to-use SDK demos

Usage notes

- Supported regions

  The hierarchical namespace feature is available only in the following regions: Australia (Sydney), US (Silicon Valley), Japan (Tokyo), India (Mumbai), UK (London), and Malaysia (Kuala Lumpur).

- Create directories

  A directory name cannot contain consecutive forward slashes (/).

  Data cannot be written to directories. The Content-Length of a directory can be set only to 0.

  The Content-Type of a directory can be set only to `application/x-directory`.

- Rename a directory or object

  The name of a renamed directory or object cannot be the same as the name of an existing directory or object in the same bucket.

  The parent directory that is included in the name of a renamed directory or object must exist. For example, if you rename a directory `destfolder/examplefolder/test`, the parent directory `destfolder/examplefolder` must exist in the bucket.

- Delete directories
  - Recursive delete

    Recursive delete is used to delete a directory and all objects and subdirectories within the directory. To use recursive delete, you must have the DeleteObject permission on the directory and all objects and subdirectories within the directory that you want to delete.

    For example, if you want to use recursive delete to delete the dest/testfolder directory and the objects in the directory, you must have the DeleteObject permission on `dest/testfolder` and `dest/testfolder/*`.

    Notice If a concurrent request is sent to write data to a directory when you use recursive delete to delete the directory, the directory may fail to be deleted.

  - Non-recursive delete

    Non-recursive delete is used to delete empty directories. To use non-recursive delete, you must have the DeleteObject permission on the directory that you want to delete.

Unsupported features

The following table describes the features that are not supported by buckets for which the hierarchical namespace feature is enabled.

Category	Description

Category	Description
Bucket features	- Cross-region replication (CRR) - Versioning - Bucket inventory - Cross-origin resource sharing (CORS) - Static website hosting - Lifecycle rules - Retention policies - Transfer acceleration
Object features	- Symbolic links - Append upload - Access control list (ACL) and tagging - Image processing (IMG) - Archive and Cold Archive objects - Callback - The RestoreObject operation that is used to restore Archive and Cold Archive objects - The `x-oss-forbid-overwrite` parameter that is used to prevent an object from being overwritten by another object that has the same name - The `response-content-*` parameter in GetObject requests that are sent to query directories - Operations that are related to LiveChannel - The DeleteMultipleObjects operation that is used to batch delete objects - The SelectObject operation that is used to select content from objects

5.7. Lifecycle

5.7.1. Overview

You can create lifecycle rules for a bucket based on the last modified time and last access time of objects in the bucket. This way, Object Storage Service (OSS) can regularly convert the storage class of the objects or delete expired objects and parts to save storage costs.

The following table describes the differences between lifecycle rules configured based on the last modified time and last access time of objects.

Policy	Based on last modified time	Based on last access time
Scenario	Objects are accessed in specific or predictable patterns.	Objects are accessed in random or unpredictable patterns.
Object deletion	Supported	Not supported
Recovery of deleted objects	**Objects in unversioned buckets cannot be recovered after the objects are deleted.** If you configure a lifecycle rule for an unversioned bucket based on the last modified time of objects in the bucket, objects that are deleted based on the lifecycle rule cannot be recovered. To recover deleted objects, enable versioning for the bucket in which the objects are stored. For more information about how to enable versioning, see Overview. If you configure a lifecycle rule for a versioned bucket based on the last modified time of objects in the bucket, the current version and previous versions of the objects are deleted in different ways based on the lifecycle rule. - If the current version of an object is deleted based on the lifecycle rule, OSS does not delete the current version but converts the current version to a previous version and adds a delete marker to the object. The delete marker becomes the new current version object. - If a previous version of an object is deleted based on the lifecycle rule, OSS deletes the previous version. In addition, if you configure a lifecycle rule to delete previous versions, delete markers that are stored as previous versions are also deleted.	N/A
Reversion of storage class conversion	The object storage class cannot be reverted after it is converted. For example, if the storage class of an object is automatically converted from Standard to IA based on a lifecycle rule, the storage class of the object cannot reverted back to Standard. For more information about the storage class conversion rules supported by lifecycle rules, see Configure lifecycle rules to automatically convert the storage class of an object. Objects whose storage class is converted to IA, Archive, and Cold Archive have minimum storage periods and minimum billable sizes. You are charged data retrieval fees when you access these objects. For more information, see Usage notes.	The object storage class can be reverted after it is converted. If the storage class of an object is automatically converted from Standard to IA based on a lifecycle rule, you can revert the storage class of the object back to Standard when you access the object. Objects whose storage class is converted to IA have a minimum storage period and a minimum billable size. You are charged data retrieval fees when you access these objects. For more information, see Usage notes.

For more information about lifecycle rules based on the last access time, see Lifecycle rules based on the last modified time.

For more information about lifecycle rules based on the last access time, see Lifecycle rules based on the last access time.

5.7.2. Lifecycle rules based on the last modified time

You can call PutBucketLifecycle to configure lifecycle rules for a bucket based on the last modified time of objects in the bucket. After the lifecycle rules are configured, Object Storage Service (OSS) regularly converts the storage class of expired objects to IA, Archive, or Cold Archive, or deletes expired objects and parts to save storage costs.

Scenarios

You can configure a lifecycle rule to regularly convert the storage class of non-hot data to IA, Archive, or Cold Archive, or delete objects that are no longer accessed. This makes data management easier and saves storage costs. For example, you can configure lifecycle rules in the following scenarios:

- A medical institution stores its medical records in OSS. These objects are occasionally accessed within six months after they are uploaded, and almost never after that. In this case, you can configure a lifecycle rule to convert the storage class of these objects to Archive 180 days after they are uploaded.
- A company stores the call records of its customer service hotline in OSS. These objects are frequently accessed within the first two months, occasionally after two months, and almost never after six months. After two years, these objects no longer need to be stored. In this case, you can configure a lifecycle rule to convert the storage class of these objects to IA 60 days after they are uploaded and to Archive 180 days after they are uploaded, and then delete them 730 days after they are uploaded.
- You can manually delete up to 1,000 objects each time. If a bucket contains more than 1,000 objects, you must delete the objects multiple times. In this case, you can configure a lifecycle rule that is used to delete all objects in the bucket the next day. This way, all objects in the bucket can be deleted the next day.

Usage notes

- Billing
  - API operation calling fees

    Successful API operations that are asynchronously called to perform actions triggered by lifecycle rules are recorded in access logs. You are charged for successful API operations. Failed operations are not recorded or charged.
  - Storage fees

    OSS charges you storage fees based on the storage class, size, and storage duration of the objects that you store. You are also charged for the entire minimum storage duration of IA, Archive, and Cold Archive objects that are deleted within the minimum storage duration based on lifecycle rules.
    - Fees charged when the storage class of objects is converted to IA or Archive and the objects are deleted within the minimum storage duration based on lifecycle rules

      Objects stored in the IA storage class have a minimum storage duration period of 30 days, and objects stored in the Archive storage class have a minimum storage duration period of 60 days. The minimum storage duration of an IA or Archive object is calculated from the time specified by the Last-Modified header of the object. For example, OSS converts a Standard object to an IA object 10 days after the object is created, converts the IA object to an Archive object after 20 days, and then deletes the Archive object after five days based on a lifecycle rule. In this case, you are also charged for the storage duration of the remaining 25 days.
    - Fees charged when the storage class of objects is converted to Cold Archive and the objects are deleted within the minimum storage duration based on lifecycle rules

      Objects stored in the Cold Archive storage class have a minimum storage duration period of 180 days. The minimum storage duration of a Cold Archive object is calculated from the time when the storage class of the object is converted to Cold Archive. For example, OSS converts a Standard object to a Cold Archive object 10 days after the object is created and then deletes the Cold Archive object the next day based on a lifecycle rule. In this case, you are also charged for the storage duration of the remaining 179 days.

    For more information about storage fees for IA, Archive, and Cold Archive objects, see Storage fees.
- Number of lifecycle rules

  You can configure up to 100 lifecycle rules for each bucket in the OSS console and up to 1,000 lifecycle rules for each bucket by using the command-line tool ossutil.
- Effective Time

  After a lifecycle rule is created, OSS loads the rule within 24 hours. After the lifecycle rule is loaded, OSS runs the rule every day at 08:00:00 (UTC+8) and completes the actions triggered by the rule within 24 hours. The interval between the last modified time of an object and the time when the lifecycle rule is run must be longer than 24 hours. For example, if you configure a lifecycle rule for a bucket to delete objects one day after they are uploaded, objects that are uploaded on July 20, 2020 are deleted on a different date based on the specific time when the objects are uploaded.
  - Objects uploaded before 08:00:00 (UTC+8) are deleted from 08:00:00 (UTC+8) on July 21, 2020 to 08:00:00 (UTC+8) on July 22, 2020.
  - Objects uploaded after 08:00:00 (UTC+8) are deleted from 08:00:00 (UTC+8) on July 22, 2020 to 08:00:00 (UTC+8) on July 23, 2020.

> **Notice** When you update a lifecycle rule, tasks performed based on the rule on the current day are suspended. We recommend that you do not frequently update lifecycle rules.

Component elements

A lifecycle rule consists of the following elements:

- Policy: specifies the objects and parts that match the rule.
  - Match by prefix: specifies that the rule matches objects and parts by prefix. You can create multiple lifecycle rules to configure different prefixes. Each prefix must be unique. The naming conventions for prefixes are the same as those for objects. For more information, see object.
  - Match by tag: specifies that the rule matches objects by tag key and tag value. You can specify multiple tags in a single lifecycle rule. The lifecycle rule applies to all objects that have the specified tags. Lifecycle rules cannot match parts by tag.

    > **Note** For more information, see Object tagging.
  - Match by prefix and tag: indicates that the rule matches objects by using specified prefixes and tags.
  - Match by bucket: indicates that the rule matches all objects and parts stored in the bucket. After you configure a lifecycle rule for a bucket to match all objects and parts in a bucket, other lifecycle rules cannot be configured for the bucket.
- Object lifecycle policy: specifies the validity period or the expiration date of objects and the operation to perform on expired objects.

- Validity period: specifies the validity period of objects in unversioned buckets and the current versions of objects in versioned buckets and the operation to perform on these objects after they expire. Objects that match the lifecycle rule are retained for the specified validity period after the objects are last modified. The specified operation is performed on these objects after they expire.
- Expiration date: specifies the expiration date of objects in unversioned buckets and the current versions of objects in versioned buckets and the operation to perform on these objects after they expire. All objects that are last modified before this date expire, and the specified operation is performed on these objects.
- Validity period for the previous versions of objects: specifies the validity period for the previous versions of objects and the operation to perform on these previous versions. Objects that match the lifecycle rule are retained for the specified validity period after the objects become previous versions. The specified operation is performed on these objects after they expire.

You can configure lifecycle rules to convert the storage class of expired objects or delete expired objects. For more information, see Configuration elements.

- Part lifecycle policy: specifies the validity period or the expiration date of parts and the operation to perform on these expired parts.
  - Validity period: specifies the validity period for parts. Parts that match the lifecycle rule are retained within the validity period and are deleted after they expire.
  - Expiration date: specifies the expiration date of parts. Parts that are last modified before this date expire and are deleted.

Matching logic

- A lifecycle rule takes effect

  For example, the following objects are stored in a bucket:

```
logs/program.log.1
logs/program.log.2
logs/program.log.3
doc/readme.txt
```

  If the prefix specified by a rule is *logs/*, the rule applies to the first three objects whose names are prefixed with *logs/*. If the prefix specified by a rule is *doc/readme.txt*, the rule applies only to the object *doc/readme.txt*.

  When GetObject or HeadObject operations are performed on an object based on a lifecycle rule, the `x-oss-expiration` header is contained in the response. This header contains two parameters: `expiry-date` that specifies the expiration date of the object, and `rule-id` that specifies the ID of the matched lifecycle rule.

- Multiple lifecycle rules conflict
  - Same prefix and tags are specified in multiple lifecycle rules

    When objects that have the same prefix and tags match multiple lifecycle rules at the same time, lifecycle rules that are configured to delete objects take precedence over lifecycle rules that are configured to convert the storage class of objects. For example, both rule1 and rule2 described in the following table apply to objects that have the abc prefix in their names and have the a=1 tag. rule1 is configured to delete matched objects 20 days after they are last modified. rule2 is configured to convert the storage class of matched objects to Archive 20 days after they are last modified. If rule1 and rule2 are configured for a bucket at the same time, rule2 does not take effect.

Rule	Prefix	Tag	Action
rule1	abc	a=1	Delete matched objects 20 days after they are last modified.
rule2	abc	a=1	Convert the storage class of matched objects to Archive 20 days after they are last modified.

  - Overlapped prefixes and the same tags are specified in multiple lifecycle rules

    For example, rule1 described in the following table applies to all objects that have the a=1 tag and is configured to convert the storage class of matched objects to IA 10 days after they are last modified. rule2 described in the following table applies to all objects that have the abc prefix in their names and have the a=1 tag. rule2 is configured to delete matched objects 120 days after they are last modified.

Rule	Prefix	Tag	Action
rule1	-	a=1	Convert the storage class of matched objects to IA 10 days after they are last modified.
rule2	abc	a=1	Delete matched objects 120 days after they are last modified.

    rule3 described in the following table applies to all objects that have the a=1 tag and is configured to convert the storage class of matched objects to Archive 20 days after they are last modified. rule4 described in the following table applies to objects that have the abc prefix in their names and have the a=1 tag and is configured to convert the storage class of matched objects to IA 30 days after they are last modified. If rule3 and rule4 are configured for a bucket at the same time, the storage class of objects that have the abc prefix in their names and have the a=1 tag is converted to Archive 20 days after they are last modified based on rule3 first. Archive objects cannot be converted to IA objects. Therefore, rule4 does not take effect.

Rule	Prefix	Tag	Action
rule3	-	a=1	Convert the storage class of matched objects to Archive 20 days after they are last modified.
rule4	abc	a=1	Convert the storage class of matched objects to IA 30 days after they are last modified.

Use the OSS console

1. Log on to the OSS console.
2. Click **Buckets**, and then click the name of the target bucket.
3. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

4. If you want to create lifecycle rules based on the last access time of objects, turn on **Enable access tracking** on the **Lifecycle** page.
5. On the page that appears, click **Create Rule**. In the **Create Rule** panel, configure the parameters. The following table describes the parameters.
   - Parameters for unversioned buckets

Section	Parameter	Description
Basic settings	**Status**	Specify the state of the lifecycle rule. Valid values: **Enabled** and **Disabled**.
	Applied To	Specify the objects to which the lifecycle rule applies. If you select **Files with Specified Prefix**, you can configure multiple lifecycle rules for objects whose names contain different prefixes. If you select **Whole Bucket**, you can configure only one lifecycle rule for the bucket.
	Prefix	Specify the prefix of objects to which the lifecycle rule applies. For example, if you want the rule to apply to objects whose names start with img, enter *img* in the field.
	Tagging	Configure tags. The rule applies only to objects that have specified tags. Example: If you select **Files with Specified Prefix** and set Prefix to img, Key to a, and Value to 1, the rule applies to all objects that have the img prefix in their names and have the tag a=1. For more information about object tagging, see Object tagging.
Policy for Objects	**File Lifecycle**	Configure rules for objects to specify when the objects expire. You can set File Lifecycle to **Validity Period (Days)**, **Expiration Date**, or **Disabled**. If you select **Disabled**, the configurations of File Lifecycle do not take effect.
	Lifecycle-based Rules	Configure the policy to convert the storage class of objects or delete expired objects. Example 1: Select Last **Access Time**, set **Validity Period (Days)** to 30, and specify that the storage class of the objects is converted to **IA** after the validity period ends. In this case, the storage class of objects whose last access time is September 1, 2021 is converted to Infrequent Access (IA) on October 1, 2021. Example 2: Select Last **Modified Time**, set **Expiration Date** to September 24, 2021, and specify that objects that are last modified before this date are deleted. In this case, objects that are last modified before September 24, 2021 are automatically deleted. The deleted objects cannot be recovered.
Policy for Parts	**Part Lifecycle**	Specify the operations to perform on expired parts. If you select **Tagging**, this option is unavailable. You can select **Validity Period (Days)**, **Expiration Date**, or **Disabled**. If you select **Disabled**, the configurations of Part Lifecycle do not take effect. **Notice** Each lifecycle rule must contain at least one of object expiration policies and part expiration policies.
	Delete Parts	Specify when parts expire based on the value of Part Lifecycle. Expired parts are automatically deleted and cannot be recovered.

   - Parameters for versioned buckets

     Configure the parameters in the **Basic Settings** and **Policy for Parts** sections in the same way as the parameters configured for unversioned buckets. The following table describes only the parameters that are different from the parameters that you can configure for unversioned buckets.

Section	Parameter	Description
Policy for Current Versions	**Clean Up Delete Marker**	If you enable versioning for the bucket, you can configure the **Clean Up Delete Marker** parameter. Other parameters are the same as those you can configure for unversioned buckets. After you select Clean Up Delete Marker, if an object has only one version and the version is a delete marker, OSS considers the delete marker expired and cleans up the delete marker. If an object has multiple versions and the current version of the object is a delete marker, OSS retains the delete marker. For more information about delete markers, see Delete marker.
Policy for Previous Versions	**File Lifecycle**	Specify when previous versions expire. Valid values: **Validity Period (Days)** or **Disabled**. If you select **Disabled**, File Lifecycle does not take effect.
	Lifecycle-based Rules	Specify the number of days within which objects can be retained after they become previous versions. After they expire, the specified operations are performed on the previous versions the next day. For example, if you set Validity Period (Days) to 30, the storage class of the objects that become previous versions on September 1, 2021 are converted to the specified storage class or deleted on October 1, 2021. **Notice** You can determine when an object becomes a previous version based on the time when the later version is generated.

6. Click **OK**.

   After a lifecycle rule is saved, you can view the rule in the lifecycle rule list.

Use ossbrowser

ossbrowser supports the same operations related to buckets as the OSS console. You can follow the on-screen instructions in ossbrowser to configure a lifecycle rule for a bucket. For more information about how to use ossbrowser, see Use ossbrowser.

Use OSS SDKs

The following code provides examples on how to use OSS SDKs for common programming languages to configure lifecycle rules. For more information about the sample code to configure lifecycle rules by using OSS SDKs for other programming languages, see Overview.

[illegible]

```
OSS ossClient = null;
try {
    // Set yourEndpoint to the endpoint of the region in which the bucket is located. For example, if the bucket is located in the China (Hangzhou) region, set yourEndpoint to https://oss-cn-hangzhou.aliyuncs.com.
    String endpoint = "yourEndpoint";
    // Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API operations. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
    String accessKeyId = "yourAccessKeyId";
    String accessKeySecret = "yourAccessKeySecret";
    // Specify the name of the bucket for which you want to configure a lifecycle rule. Example: examplebucket.
    String bucketName = "examplebucket";
    // Create an OSSClient instance.
    ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
    // Create a request by using SetBucketLifecycleRequest.
    SetBucketLifecycleRequest request = new SetBucketLifecycleRequest(bucketName);
    // Specify the ID for the lifecycle rule.
    String ruleId0 = "rule0";
    // Specify the prefix that you want the lifecycle rule to match.
    String matchPrefix0 = "A0/";
    // Specify the tag that you want the lifecycle rule to match.
    Map<String, String> matchTags0 = new HashMap<String, String>();
    // Specify the key and value of the object tag. Example: the key is owner, and the value is John.
    matchTags0.put("owner", "John");
    String ruleId1 = "rule1";
    String matchPrefix1 = "A1/";
    Map<String, String> matchTags1 = new HashMap<String, String>();
    matchTags1.put("type", "document");
    String ruleId2 = "rule2";
    String matchPrefix2 = "A2/";
    String ruleId3 = "rule3";
    String matchPrefix3 = "A3/";
    String ruleId4 = "rule4";
    String matchPrefix4 = "A4/";
    String ruleId5 = "rule5";
    String matchPrefix5 = "A5/";
    String ruleId6 = "rule6";
    String matchPrefix6 = "A6/";
    // Set the expiration time to three days after the last modified time.
    LifecycleRule rule = new LifecycleRule(ruleId0, matchPrefix0, LifecycleRule.RuleStatus.Enabled, 3);
    rule.setTags(matchTags0);
    request.AddLifecycleRule(rule);
    // Specify that objects that are created before the expiration date expire.
    rule = new LifecycleRule(ruleId1, matchPrefix1, LifecycleRule.RuleStatus.Enabled);
    rule.setCreatedBeforeDate(DateUtil.parseIso8601Date("2022-10-12T00:00:00.000Z"));
    rule.setTags(matchTags1);
    request.AddLifecycleRule(rule);
    // Set the expiration time to three days for parts of an object.
    rule = new LifecycleRule(ruleId2, matchPrefix2, LifecycleRule.RuleStatus.Enabled);
    LifecycleRule.AbortMultipartUpload abortMultipartUpload = new LifecycleRule.AbortMultipartUpload();
    abortMultipartUpload.setExpirationDays(3);
    rule.setAbortMultipartUpload(abortMultipartUpload);
    request.AddLifecycleRule(rule);
    // Specify that parts that are created before the expiration date expire.
    rule = new LifecycleRule(ruleId3, matchPrefix3, LifecycleRule.RuleStatus.Enabled);
    abortMultipartUpload = new LifecycleRule.AbortMultipartUpload();
    abortMultipartUpload.setCreatedBeforeDate(DateUtil.parseIso8601Date("2022-10-12T00:00:00.000Z"));
    rule.setAbortMultipartUpload(abortMultipartUpload);
    request.AddLifecycleRule(rule);
    // Specify that the storage class of objects is converted to IA 10 days after they are last modified, and to Archive 30 days after they are last modified.
    rule = new LifecycleRule(ruleId4, matchPrefix4, LifecycleRule.RuleStatus.Enabled);
    List<LifecycleRule.StorageTransition> storageTransitions = new ArrayList<LifecycleRule.StorageTransition>();
    LifecycleRule.StorageTransition storageTransition = new LifecycleRule.StorageTransition();
    storageTransition.setStorageClass(StorageClass.IA);
    storageTransition.setExpirationDays(10);
    storageTransitions.add(storageTransition);
    storageTransition = new LifecycleRule.StorageTransition();
    storageTransition.setStorageClass(StorageClass.Archive);
    storageTransition.setExpirationDays(30);
    storageTransitions.add(storageTransition);
    rule.setStorageTransition(storageTransitions);
    request.AddLifecycleRule(rule);
    // Specify that the storage class of objects that are last modified before October 12, 2022 is converted to Archive.
    rule = new LifecycleRule(ruleId5, matchPrefix5, LifecycleRule.RuleStatus.Enabled);
    storageTransitions = new ArrayList<LifecycleRule.StorageTransition>();
    storageTransition = new LifecycleRule.StorageTransition();
    storageTransition.setCreatedBeforeDate(DateUtil.parseIso8601Date("2022-10-12T00:00:00.000Z"));
    storageTransition.setStorageClass(StorageClass.Archive);
    storageTransitions.add(storageTransition);
    rule.setStorageTransition(storageTransitions);
    request.AddLifecycleRule(rule);
    // Specify that rule6 is configured for versioned buckets.
    rule = new LifecycleRule(ruleId6, matchPrefix6, LifecycleRule.RuleStatus.Enabled);
    // Specify that the storage class of objects is converted to Archive 365 days after the objects are last modified.
    storageTransitions = new ArrayList<LifecycleRule.StorageTransition>();
    storageTransition = new LifecycleRule.StorageTransition();
    storageTransition.setStorageClass(StorageClass.Archive);
    storageTransition.setExpirationDays(365);
    storageTransitions.add(storageTransition);
    rule.setStorageTransition(storageTransitions);
    // Configure the lifecycle rule to automatically delete expired delete markers.
    rule.setExpiredDeleteMarker(true);
    // Configure the lifecycle rule to convert the previous versions of objects to the IA storage class 10 days after the objects are last modified.
    LifecycleRule.NoncurrentVersionStorageTransition noncurrentVersionStorageTransition =
            new LifecycleRule.NoncurrentVersionStorageTransition().withNoncurrentDays(10).withStrorageClass(StorageClass.IA);
    // Specify that the storage class of the previous versions of objects is converted to Archive 20 days after the objects are last modified.
    LifecycleRule.NoncurrentVersionStorageTransition noncurrentVersionStorageTransition2 =
            new LifecycleRule.NoncurrentVersionStorageTransition().withNoncurrentDays(20).withStrorageClass(StorageClass.Archive);
    // Specify that the previous versions of objects are deleted 30 days after the objects become previous versions.
    LifecycleRule.NoncurrentVersionExpiration noncurrentVersionExpiration = new LifecycleRule.NoncurrentVersionExpiration().withNoncurrentDays(30);
    List<LifecycleRule.NoncurrentVersionStorageTransition> noncurrentVersionStorageTransitions = new ArrayList<LifecycleRule.NoncurrentVersionStorage
Transition>();
    noncurrentVersionStorageTransitions.add(noncurrentVersionStorageTransition2);
    rule.setStorageTransition(storageTransitions);
    rule.setNoncurrentVersionExpiration(noncurrentVersionExpiration);
    rule.setNoncurrentVersionStorageTransitions(noncurrentVersionStorageTransitions);
    request.AddLifecycleRule(rule);
    // Initiate a request to configure lifecycle rules.
    ossClient.setBucketLifecycle(request);
    // Query the lifecycle rules configured for the bucket.
    List<LifecycleRule> listRules = ossClient.getBucketLifecycle(bucketName);
    for(LifecycleRule rules : listRules){
        System.out.println("ruleId="+rules.getId()+", matchPrefix="+rules.getPrefix());
    }
} catch (Exception e) {
    e.printStackTrace();
} finally {
    if(ossClient != null){
        // Shut down the OSSClient instance.
        ossClient.shutdown();
    }
}
```

Use ossutil

For more information about how to configure lifecycle rules by using ossutil, see Add or modify lifecycle rules.

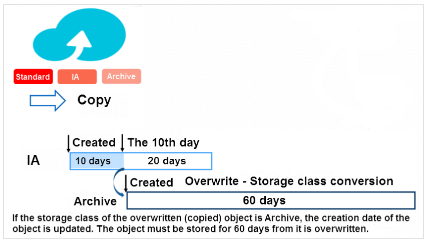
Use the RESTful API

If your program requires more custom options in the configuration of lifecycle rules based on the last modified time, you can call RESTful API operations. In this case, you need to manually write code to calculate the signature. For more information, see PutBucketLifecycle.

FAQ

- Does OSS impose limits on the minimum storage duration of objects whose storage class is converted by using CopyObject?

  Yes, OSS imposes limits on the minimum storage duration of IA, Archive, and Cold Archive objects. For example, if you use CopyObject to convert the storage class of an IA object to Archive 10 days after the object is created, you are charged for the entire minimum storage duration of the IA object. In addition, the Last-Modified value of the object is updated to the time when the storage class of the object is converted. The converted Archive object must be stored for at least 60 days. Otherwise, you are charged for the minimum storage duration of Archive objects.

  If the storage class of the overwritten (copied) object is Archive, the creation date of the object is updated. The object must be stored for 60 days from it is overwritten.

- Does OSS record operations when the storage class of objects is converted or expired objects are deleted based on lifecycle rules?

  Yes, OSS records operations when the storage class of objects is converted or expired objects are deleted based on lifecycle rules. The logs include the following fields:

  - Operation
    - CommitTransition: indicates the storage class to which the object is converted based on lifecycle rules. Example: IA, Archive, or Cold Archive.
    - ExpireObject: indicates the expired objects that are deleted based on lifecycle rules.
  - Sync Request

    lifecycle: indicates the operations that are triggered by lifecycle rules, such as deleting expired objects and converting the storage class of objects.

5.7.3. Lifecycle rules based on the last access time

You can configure lifecycle rules for a bucket based on the last access time of objects in the bucket. After you configure a lifecycle rule based on the last access time for a bucket, Object Storage Service (OSS) monitors the access patterns of objects in the bucket, identifies cold data, and converts the storage class of cold data. This way, cold data is stored by using storage classes that are different from the storage classes of hot data, which helps you reduce storage costs.

Scenarios

- Multimedia data storage

  If you store the videos and images of your website in an OSS bucket, the data may become infrequently accessed over time. For data that becomes infrequently accessed, you may need to convert the storage class to Infrequent Access (IA). For data that has been uploaded for a long time but is still frequently accessed, you need to retain the Standard storage class. In this case, you can configure a lifecycle rule based on the last access time for the bucket. This way, cold and hot data is stored by using different storage classes and storage costs are reduced.

- Albums or network disks

  You can configure a lifecycle rule based on the last access time for a bucket that is used to store albums or used as a network disk. Then, the storage class of cold data is automatically converted to IA after the lifecycle rule is triggered, and the data can be accessed in real time.

- Life science data storage

  A large amount of data is generated in gene sequencing, and users need to determine whether the data is frequently accessed based on the last access time but not last modified time of the data. In the past, users must manually analyze logs to identify cold and hot data and manage cold and hot data by using different methods. In this case, you can configure a lifecycle rule based on the last access time. This way, OSS automatically identifies cold and hot objects based on the last access time of the objects and stores the objects by using different storage classes. In addition, you can specify policies based on the last access time and last modified time in the same lifecycle rule to manage data in a more flexible manner.

Implementation methods

If you want to create a lifecycle rule based on the last access time for a bucket, you must enable access tracking for the bucket in the OSS console. For more information, see Configure lifecycle rules.

Usage notes

- Supported regions

  You can configure lifecycle rules based on the last access time only for buckets in the China (Qingdao), China (Hohhot), Germany (Frankfurt), and Australia (Sydney) regions.

- Policies based on the last access time

  After you enable access tracking for a bucket, the last access time of all objects in the bucket is set to the time when access tracking is enabled and is updated based on the actual access time of the objects. If multiple GetObject requests are sent to access the same object within 24 hours, the last access time of the object is set to the time when the object is accessed by the first GetObject request within the period.

  In addition, if an object is accessed by using a symbolic link that points to the object, the last access time of the object is not updated.

- Supported storage classes

  In a lifecycle rule based on the last access time, you can specify policies to convert the storage class of objects from Standard to IA. You can also specify whether to convert the storage class of the IA objects back to Standard when the objects are accessed. You cannot convert the storage classes of objects to Archive or Cold Archive by using the lifecycle rules based on the last access time.

- Billing
  - Object monitoring and management fees

    After you enable access tracking for a bucket, object monitoring and management fees are generated. However, you are not charged such fees temporarily.

  - Storage fees

    You can configure lifecycle rules based on the last access time for objects of all sizes. You are charged storage fees for the objects based on the sizes and storage classes of the objects. If the storage class of the objects is Standard, you are charged storage fees based on the sizes of the objects and the unit price of Standard objects. If the storage class of the objects is IA, you are charged storage fees based on the sizes of the objects and the unit price of IA objects. Objects have a minimum billable size of 64 KB. Objects that are equal to or larger than 64 KB in size are billed based on their actual sizes.

  - Storage fees for IA objects that are stored for less than the minimum storage duration

    IA objects has a minimum storage duration of 30 days. You are charged for the entire minimum storage duration of IA objects if the IA objects are stored for less than the minimum storage duration. The following examples show how IA objects are charged when lifecycle rules based on the last access time are configured:

    Example 1: OSS converts a Standard object to an IA object 10 days after the object is created, and then converts the IA object back to a Standard object after 5 days based on a lifecycle rule. In this case, you are also charged for the storage duration of the remaining 15 days.

    Example 2: OSS converts a Standard object to an IA object 10 days after the object is created, and then deletes the IA object after 15 days based on a lifecycle rule. In this case, you are also charged for the storage duration of the remaining 5 days.

    If you call the CopyObject operation to overwrite a Standard object with an IA object, the IA object also has a minimum storage duration. For example, if you call the CopyObject operation to convert the storage class of a Standard object to IA 10 days after the Standard object is created, and then delete the IA object after 10 days, you are also charged for the storage duration of the remaining 20 days.

  - Retrieval fees for IA objects

    When you access IA objects, you are charged data retrieval fees based on the size of the retrieved IA objects.

FAQ

- Why is a lifecycle rule unable to immediately take effect after the rule is created?

  After a lifecycle rule is created, OSS loads the rule within 24 hours. After the lifecycle rule is loaded, OSS runs the rule every day at 08:00:00 (UTC+8) and completes the actions triggered by the rule within 24 hours. Therefore, a lifecycle rule completely takes effect up to 48 hours after the rule is created.

- What happens if I configure a lifecycle rule based on the last modified time and a lifecycle rule based on the last access time at the same time for objects that have the same prefix in the same bucket?

  For example, you configure two lifecycle rules for a bucket named examplebucket at the same time. The first rule specifies that all objects whose names are prefixed with doc in examplebucket are deleted 30 days after the objects are last modified. The second rule specifies that the storage class of all objects whose names are prefixed with doc in examplebucket is converted to IA 30 days after the objects are last accessed.

In this case, only the first lifecycle rule takes effect because OSS tends to implement lifecycle rules that involve lower fees. If the first rule is implemented, you are not charged after the specified objects are deleted based on the rule. If the second rule is implemented, you are still charged storage fees and data retrieval fees after the storage class of the specified objects is converted based on the rule.

- When does a lifecycle rule take effect after I modify the rule? What happens to the objects to which the original rule applies?

  For example, you configure a lifecycle rule for objects whose names contain the `er` prefix. Based on the lifecycle rule, the storage class of the objects is converted to IA 30 days after the objects are last accessed, and the storage class of the objects can be converted back to Standard when the IA objects are accessed after 30 days. In this case, if you change the prefix that you specify in the lifecycle rule from `er` to `re` 35 days after the objects whose names are prefixed with er are last accessed, the storage class of these objects has already been converted to IA and cannot be converted back to Standard based on the original lifecycle rule. After the lifecycle rule is modified, the last access time of the objects whose names contain the re prefix is set to the time when access tracking is enabled for the bucket.

- How are objects stored if I configure lifecycle rules based on the last access time for a versioned bucket?

  In a versioned bucket, each object has a unique version ID. Objects whose version IDs are different are separately stored. Therefore, after you configure lifecycle rules based on the last access time for a versioned bucket, the storage class of the current version of an object may be different from the storage class of a previous version of the same object.

- Can I disable access tracking?

  Yes, you can disable access tracking. Before you disable access tracking for a bucket, you must make sure that no lifecycle rules based on the last access time are configured for the bucket. After you disable access tracking for a bucket, OSS stops tracking the last access time of objects in the bucket. If you enable access tracking for the bucket again, the last access time of objects in the bucket is reset.

5.7.4. Configuration elements

This topic describes the elements that you can configure in lifecycle rules for objects.

The configuration file of lifecycle rules for objects is in the XML format. Example:

```
<LifecycleConfiguration>
<Rule>
 <ID>rule1</ID>
 <Prefix>logs/</Prefix>
 <Status>Enabled</Status>
 <Expiration>
   <Days>10</Days>
  </Expiration>
</Rule>
<Rule>
 <ID>rule2</ID>
 <Prefix>doc/</Prefix>
 <Status>Disabled</Status>
 <Expiration>
   <CreatedBeforeDate>2017-12-31T00:00:00.000Z</CreatedBeforeDate>
 </Expiration>
</Rule>
<Rule>
 <ID>rule3</ID>
 <Tag><Key>xx</Key><Value>1</Value></Tag>
 <Status>Enabled</Status>
 <Transition>
   <Days>60</Days>
   <StorageClass>Archive</StorageClass>
</Transition>
</Rule>
</LifecycleConfiguration>
```

The following section describes the three lifecycle rules in the preceding example:

- The first rule is used to delete the objects whose names contain the logs/ prefix and whose last modified date is 10 days ago.
- The second rule is used to delete the objects whose names contain the doc/ prefix and whose last modified date is before December 31, 2017. This rule does not take effect because the rule is in the Disabled state.
- The third rule is used to convert the storage class of the objects whose setting is xx=1 and whose storage class was converted to Archive 60 days ago.

The following section describes the elements such as ID and operation elements that you can configure in lifecycle rules.

ID

The ID element specifies the ID of the lifecycle rule that is configured for the bucket. The ID of a lifecycle rule can be up to 255 bytes in length. If you do not configure this element or you leave the value empty, Object Storage Service (OSS) automatically generates a unique ID for the lifecycle rule.

Status

The Status element specifies the status of the lifecycle rule, which can be Enabled or Disabled. Only rules in which the value of Status is Enabled take effect.

Prefix

The Prefix element specifies the object name prefix based on which the lifecycle rule applies to all or some objects in the bucket.

Time

- Date

  The <CreatedBeforeDate> child element specifies an absolute date. All objects that were last modified before the date expire, and the specified operation is performed on the objects.

- Days

The <Days> child element specifies the validity period during which objects are retained after the objects were last modified. After the validity period ends, the specified operation is performed on the objects.

Operation elements

You can configure one or more operation elements in a lifecycle rule for an object. Then, OSS performs the specified operations on the object during the lifecycle. The effect of the operations varies based on the versioning status of the bucket. The following section describes how the versioning status of the bucket in which the object is stored affects the operation specified in the lifecycle rule for the object.

- Buckets that have versioning disabled

Operation	Description
Transition	Specifies the date or the validity period after which the storage class of the object is converted to the specified storage class. For more information about the storage classes that you can specify in lifecycle rules, see Configure lifecycle rules to automatically convert the storage class of an object.
Expiration	Specifies the date or the validity period after which objects that match the lifecycle rule are permanently deleted.

- Buckets that have versioning enabled

  The following section describes the elements that are specified in lifecycle rules for objects in versioned buckets. Versioned buckets can be versioning-enabled or versioning-suspended.

  - Delete or storage class conversion operation on the current version of an object

Operation	Description
Expiration	The operation performed on an expired object varies based on whether the current version of the object is a delete marker. ■ If the current version of the object is not a delete marker, OSS processes the version based on the following rules: ■ If versioning is enabled for the bucket that contains the object, OSS inserts a delete marker that has a unique version ID. Then, the current version of the object is replaced by the inserted delete marker. ■ If versioning is suspended for the bucket that contains the object, OSS inserts a delete marker whose version ID is null. Then, the inserted delete marker overwrites the current version of the object to make sure that only one version of the object whose version ID is null exists. ■ If the current version of the object is a delete marker, OSS processes the version based on the following rules: ■ If one or more previous versions of the object exist, no operation that is specified in the lifecycle rule is triggered. ■ If only one version of the object exists, the delete marker is removed after the delete marker expires based on the date specified in the lifecycle rule or when the ExpiredObjectDeleteMarker child element is set to true in the lifecycle rule. **Notice** ■ If the Expiration operation of a lifecycle rule configured for an object is triggered or a delete operation is initiated with no object version specified, the current version of the object becomes the previous version of the object. If the Expiration operation of a lifecycle rule configured for an object is triggered or a delete operation is initiated on a specified version of the object, all previous versions of the object are permanently deleted and only the expired delete marker of the object is retained. ■ You cannot configure the ExpiredObjectDeleteMarker child element and specify a tag in a lifecycle rule at the same time.
Transition	Specifies the date or the validity period after which the storage class of the current version of the object is converted to the specified storage class.

  - Delete or storage class conversion operation on the previous versions of an object

Operation	Description	Associated child element
NoncurrentVersionExpiration	Specifies the date or the validity period after which the previous versions of the object are deleted.	The <NoncurrentDays> child element specifies the validity period during which the previous versions of the object are retained. After the validity period ends, the previous versions are permanently deleted. **Note** For example, a version was the current version of an object. On May 1, 2019, the PutObject operation was called, and this version became a previous version. In the NoncurrentVersionExpiration element, <NoncurrentDays> is set to 3. On May 4, 2019, the version was permanently deleted. Each time the PutObject operation is called for an object, the current version of the object becomes the most recent previous version. The uploaded version becomes the current version of the object. OSS determines when a version becomes a previous version based on the time when the next version is created.
NoncurrentVersionTransition	Specifies the date or validity period after which the storage class of previous versions is converted.	■ The <NoncurrentDays> child element specifies the validity period during which previous versions are retained. After the validity period ends, the storage class of previous versions is converted. ■ The <StorageClass> child element specifies the storage class to which the storage class of matched objects is converted.

5.7.5. Configuration examples

This topic provides several common examples of lifecycle configurations for you to better manage objects in your bucket.

Specify filter conditions

Each lifecycle rule contains at least one filter condition, which determines whether the lifecycle rule applies to a part of or all objects in a bucket. The following examples describe how to specify a filter condition in lifecycle configurations:

- In the lifecycle rule configurations, the `doc/` prefix is specified as a filter condition. This condition indicates that the lifecycle rule applies only to objects whose names are prefixed with `doc/`, such as `doc/test1.txt` and `doc/test2.jpg`. In addition, the rule specifies that the storage class of matched objects is converted to Infrequent Access (IA) 180 days after the last modified time, and that the matched objects are deleted 365 days after the last modified time.

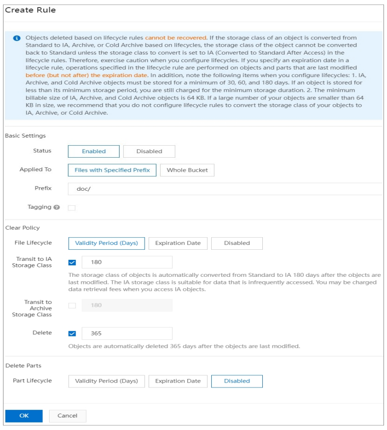
  The following examples show the XML and console configurations of the lifecycle rule:

  - XML

    ```
    <LifecycleConfiguration>
      <Rule>
        <ID>test-rule0</ID>
        <Prefix>doc/</Prefix>
        <Status>Enabled</Status>
        <Expiration>
          <Days>365</Days>
        </Expiration>
        <Transition>
          <Days>180</Days>
          <StorageClass>IA</StorageClass>
        </Transition>
      </Rule>
    </LifecycleConfiguration>
    ```

  - Console configurations

    > **Note** The following figure shows the configurations of the preceding lifecycle rule in the OSS console: For more information, see Configure lifecycle rules.



Create Rule

Objects deleted based on lifecycle rules cannot be recovered. If the storage class of an object is converted from Standard to IA, Archive, or Cold Archive based on lifecycles, the storage class of the object cannot be converted back to Standard unless the storage class to convert is set to IA (Converted to Standard After Access) in the lifecycle rules. Therefore, exercise caution when you configure lifecycles. If you specify an expiration date in a lifecycle rule, operations specified in the lifecycle rule are performed on objects and parts that are last modified before (but not after) the expiration date. In addition, note the following items when you configure lifecycles: 1. IA, Archive, and Cold Archive objects must be stored for a minimum of 30, 60, and 180 days. If an object is stored for less than its minimum storage period, you are still charged for the minimum storage duration. 2. The minimum billable size of IA, Archive, and Cold Archive objects is 64 KB. If a large number of your objects are smaller than 64 KB in size, we recommend that you do not configure lifecycle rules to convert the storage class of your objects to IA, Archive, or Cold Archive.

Basic Settings

Status: [✓ Enabled] Disabled

Applied To: [✓ Files with Specified Prefix] Whole Bucket

Prefix: doc/

Tagging □

Clear Policy

File Lifecycle: [✓ Validity Period (Days)] Expiration Date Disabled

Transit to IA Storage Class ☑ 180

The storage class of objects is automatically converted from Standard to IA 180 days after the objects are last modified. The IA storage class is suitable for data that is infrequently accessed. You may be charged data retrieval fees when you access IA objects.

Transit to Archive Storage Class □ 180

Delete ☑ 365

Objects are automatically deleted 365 days after the objects are last modified.

Delete Parts

Part Lifecycle: Validity Period (Days) Expiration Date [✓ Disabled]

Example 1

- In the following lifecycle configuration example, the filter condition indicates that the lifecycle rule applies to all objects in the bucket. All objects in the bucket expire 300 days after the objects are last modified based on the lifecycle rule.

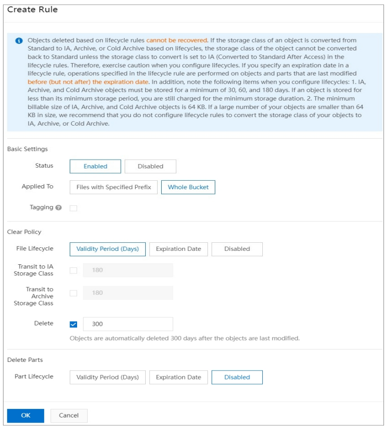
  The following examples show the XML and console configurations of the lifecycle rule:

  - XML

    ```
    <LifecycleConfiguration>
      <Rule>
        <ID>test-rule1</ID>
        <Prefix></Prefix>
        <Status>Enabled</Status>
        <Expiration>
          <Days>300</Days>
        </Expiration>
      </Rule>
    </LifecycleConfiguration>
    ```

- Console configurations

  Note The following figure shows the configurations of the preceding lifecycle rule in the OSS console: For more information, see Configure lifecycle rules.



Create Rule

Objects deleted based on lifecycle rules cannot be recovered. If the storage class of an object is converted from Standard to IA, Archive, or Cold Archive based on lifecycles, the storage class of the object cannot be converted back to Standard unless the storage class to convert is set to IA (Converted to Standard After Access) in the lifecycle rules. Therefore, exercise caution when you configure lifecycles. If you specify an expiration date in a lifecycle rule, operations specified in the lifecycle rule are performed on objects and parts that are last modified before (but not after) the expiration date. In addition, note the following items when you configure lifecycles: 1. IA, Archive, and Cold Archive objects must be stored for a minimum of 30, 60, and 180 days. If an object is stored for less than its minimum storage period, you are still charged for the minimum storage duration. 2. The minimum billable size of IA, Archive, and Cold Archive objects is 64 KB. If a large number of your objects are smaller than 64 KB in size, we recommend that you do not configure lifecycle rules to convert the storage class of your objects to IA, Archive, or Cold Archive.

Basic Settings

Status: Enabled

Applied To: Whole Bucket

Tagging: □

Clear Policy

File Lifecycle: Validity Period (Days)

Transit to IA Storage Class: □ 180

Transit to Archive Storage Class: □ 180

Delete: ☑ 300

Objects are automatically deleted 300 days after the objects are last modified.

Delete Parts

Part Lifecycle: Disabled

Example 2

- In the following lifecycle configuration example, the Prefix parameter is not configured, which indicates that the lifecycle rule applies to all objects in the bucket. Objects that are modified before December 30, 2021 expire based on the lifecycle rule.

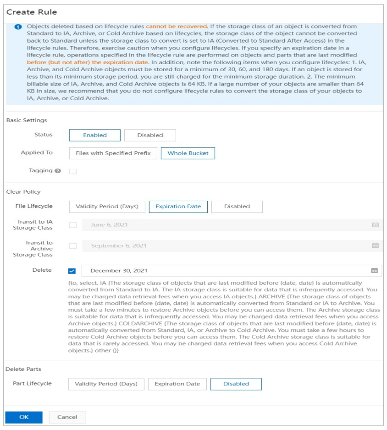
  The following examples show the XML and console configurations of the lifecycle rule:

  - XML

```
<LifecycleConfiguration>
  <Rule>
    <ID>test-rule0</ID>
    <Prefix></Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <CreatedBeforeDate>2021-12-30T00:00:00.000Z</CreatedBeforeDate>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

  - Console configurations

  Note The following figure shows the configurations of the preceding lifecycle rule in the OSS console: For more information, see Configure lifecycle rules.



Create Rule

Objects deleted based on lifecycle rules cannot be recovered. If the storage class of an object is converted from Standard to IA, Archive, or Cold Archive based on lifecycles, the storage class of the object cannot be converted back to Standard unless the storage class to convert is set to IA (Converted to Standard After Access) in the lifecycle rules. Therefore, exercise caution when you configure lifecycles. If you specify an expiration date in a lifecycle rule, operations specified in the lifecycle rule are performed on objects and parts that are last modified before (but not after) the expiration date. In addition, note the following items when you configure lifecycles: 1. IA, Archive, and Cold Archive objects must be stored for a minimum of 30, 60, and 180 days. If an object is stored for less than its minimum storage period, you are still charged for the minimum storage duration. 2. The minimum billable size of IA, Archive, and Cold Archive objects is 64 KB. If a large number of your objects are smaller than 64 KB in size, we recommend that you do not configure lifecycle rules to convert the storage class of your objects to IA, Archive, or Cold Archive.

Basic Settings

Status: Enabled

Applied To: Whole Bucket

Tagging: □

Clear Policy

File Lifecycle: Expiration Date

Transit to IA Storage Class: □ June 6, 2021

Transit to Archive Storage Class: □ September 6, 2021

Delete: ☑ December 30, 2021

{to, select, IA {The storage class of objects that are last modified before {date, date} is automatically converted from Standard to IA. The IA storage class is suitable for data that is infrequently accessed. You may be charged data retrieval fees when you access IA objects.} ARCHIVE {The storage class of objects that are last modified before {date, date} is automatically converted from Standard or IA to Archive. You must take a few minutes to restore Archive objects before you can access them. The Archive storage class is suitable for data that is infrequently accessed. You may be charged data retrieval fees when you access Archive objects.} COLDARCHIVE {The storage class of objects that are last modified before {date, date} is automatically converted from Standard, IA, or Archive to Cold Archive. You must take a few hours to restore Cold Archive objects before you can access them. The Cold Archive storage class is suitable for data that is rarely accessed. You may be charged data retrieval fees when you access Cold Archive objects.} other {}}

Delete Parts

Part Lifecycle: Disabled

Example 3

Overlapping filter conditions

The following examples show whether the operations specified in lifecycle rules conflict when the filter conditions of the lifecycle rules overlap.

- Assume that you specify the following lifecycle rules:
  - Rule 1 specifies a filter condition based on tag A, and specifies that the storage class of objects with the tag are converted to IA 180 days after the objects are last modified.
  - Rule 2 specifies a filter condition based on tag B, and specifies that the objects with the tag expire 10 days after the objects are last modified.

  If an object has the two tags specified in the two lifecycle rules, both lifecycle rules apply to the object. In this case, the object expires 10 days after it is last modified based on the second lifecycle rule. After the object is deleted, the storage class conversion operation specified in the first lifecycle rule cannot be performed. Therefore, only the expiration operation specified in the second lifecycle takes effect in this example.

The following examples show the XML and console configurations of the lifecycle rule:

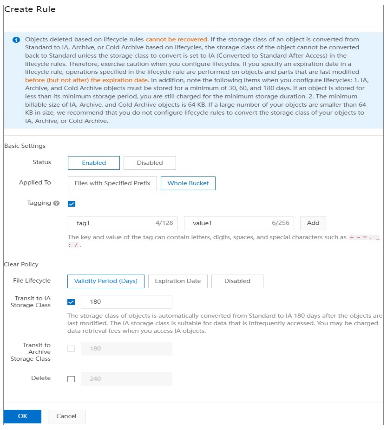
- XML

```
<LifecycleConfiguration>
  <Rule>
    <ID>test-rule1</ID>
    <Prefix></Prefix>
    <Tag>
      <Key>tag1</Key>
      <Value>value1</Value>
    </Tag>
    <Status>Enabled</Status>
    <Transition>
      <Days>180</Days>
      <StorageClass>IA</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>test-rule2</ID>
    <Prefix></Prefix>
    <Tag>
      <Key>tag2</Key>
      <Value>value2</Value>
    </Tag>
    <Status>Enabled</Status>
    <Expiration>
      <Days>10</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

- Console configurations

> **Note** The following figure shows the configurations of the preceding lifecycle rule in the OSS console: For more information, see Configure lifecycle rules.

- The following figure shows the configurations of the preceding rule 1 in the console:



Create Rule

Objects deleted based on lifecycle rules cannot be recovered. If the storage class of an object is converted from Standard to IA, Archive, or Cold Archive based on lifecycles, the storage class of the object cannot be converted back to Standard unless the storage class to convert is set to IA (Converted to Standard After Access) in the lifecycle rules. Therefore, exercise caution when you configure lifecycles. If you specify an expiration date in a lifecycle rule, operations specified in the lifecycle rule are performed on objects and parts that are last modified before (but not after) the expiration date. In addition, note the following items when you configure lifecycles: 1. IA, Archive, and Cold Archive objects must be stored for a minimum of 30, 60, and 180 days. If an object is stored for less than its minimum storage period, you are still charged for the minimum storage duration. 2. The minimum billable size of IA, Archive, and Cold Archive objects is 64 KB. If a large number of your objects are smaller than 64 KB in size, we recommend that you do not configure lifecycle rules to convert the storage class of your objects to IA, Archive, or Cold Archive.

Basic Settings

Status: [✓ Enabled] Disabled

Applied To: Files with Specified Prefix [✓ Whole Bucket]

Tagging ☑

tag1 4/128　value1 6/256　Add

The key and value of the tag can contain letters, digits, spaces, and special characters such as + - = . _ : /.

Clear Policy

File Lifecycle: [✓ Validity Period (Days)] Expiration Date Disabled

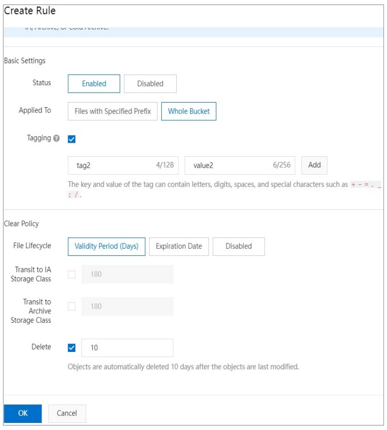
Transit to IA Storage Class ☑ 180

The storage class of objects is automatically converted from Standard to IA 180 days after the objects are last modified. The IA storage class is suitable for data that is infrequently accessed. You may be charged data retrieval fees when you access IA objects.

Transit to Archive Storage Class □ 180

Delete □ 240

- The following figure shows the configurations of the preceding rule 2 in the console:



Create Rule

Basic Settings

Status: [✓ Enabled] Disabled

Applied To: Files with Specified Prefix [✓ Whole Bucket]

Tagging ☑

tag2 4/128　value2 6/256　Add

The key and value of the tag can contain letters, digits, spaces, and special characters such as + - = . _ : /.

Clear Policy

File Lifecycle: [✓ Validity Period (Days)] Expiration Date Disabled

Transit to IA Storage Class □ 180

Transit to Archive Storage Class □ 180

Delete ☑ 10

Objects are automatically deleted 10 days after the objects are last modified.

Example 1 shows that the operations specified in lifecycle rules conflict

- Assume that you specify the following lifecycle rules whose specified prefixes overlap:
  - Rule 1 leaves Prefix empty, which indicates that the rule applies to all objects in the bucket. All objects in the bucket are deleted 365 days after the objects are last modified.

- Rule 2 specifies Prefix as test/, and specifies that the storage class of the objects whose names contain the prefix is converted to Archive 30 days after the objects are last modified.

Therefore, the operations specified in the lifecycle rules take effect.

The following examples show the XML and console configurations of the lifecycle rule:

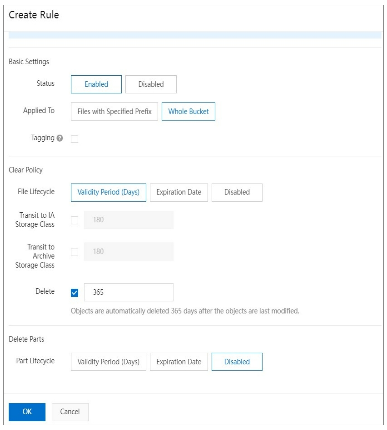
- XML

```
<LifecycleConfiguration>
  <Rule>
    <ID>test-rule1</ID>
    <Prefix></Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>test-rule2</ID>
    <Prefix>test/</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>Archive</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

- Console configurations

> **Note** The following figure shows the configurations of the preceding lifecycle rule in the OSS console: For more information, see Configure lifecycle rules.

  - The following figure shows the configurations of the preceding rule 1 in the console:



Create Rule

Basic Settings

Status: [✓ Enabled] Disabled

Applied To: Files with Specified Prefix [✓ Whole Bucket]

Tagging: □

Clear Policy

File Lifecycle: [✓ Validity Period (Days)] Expiration Date Disabled

Transit to IA Storage Class: □ 180

Transit to Archive Storage Class: □ 180

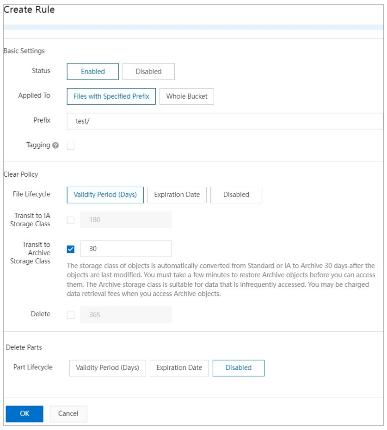
Delete: ☑ 365

Objects are automatically deleted 365 days after the objects are last modified.

Delete Parts

Part Lifecycle: Validity Period (Days) Expiration Date [✓ Disabled]

  - The following figure shows the configurations of the preceding rule 2 in the console:



Create Rule

Basic Settings

Status: [✓ Enabled] Disabled

Applied To: [✓ Files with Specified Prefix] Whole Bucket

Prefix: test/

Tagging: □

Clear Policy

File Lifecycle: [✓ Validity Period (Days)] Expiration Date Disabled

Transit to IA Storage Class: □ 180

Transit to Archive Storage Class: ☑ 30

The storage class of objects is automatically converted from Standard or IA to Archive 30 days after the objects are last modified. You must take a few minutes to restore Archive objects before you can access them. The Archive storage class is suitable for data that is infrequently accessed. You may be charged data retrieval fees when you access Archive objects.

Delete: □ 365

Delete Parts

Part Lifecycle: Validity Period (Days) Expiration Date [✓ Disabled]

Example 2 shows that the operations specified in lifecycle rules do not conflict

Disable lifecycle rules

In this example, two lifecycle rules are configured. The first rule specifies that the storage class of objects whose names contain the `logs/` prefix are converted to IA 100 days after the objects are created. The second rule specifies that the storage class of objects whose names contain the `documents/` prefix are converted to Archive 50 days after the objects are created. Then, disable the first rule and enable the second rule.

After the lifecycle configurations take effect, only the second lifecycle rule in which the value of <Status> is Enabled takes effect. In this case, the storage class of objects whose names contain the `documents/` prefix are converted to Archive 50 days after they are created.

The following examples show the XML and console configurations of the lifecycle rule:

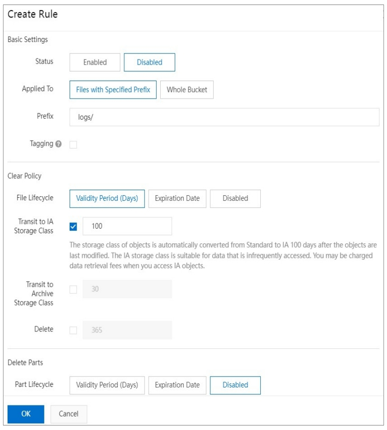
- XML

```
<LifecycleConfiguration>
  <Rule>
    <ID>test-rule1</ID>
    <Prefix>logs/</Prefix>
    <Status>Disabled</Status>
    <Transition>
      <Days>100</Days>
      <StorageClass>IA</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>test-rule2</ID>
    <Prefix>documents/</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>50</Days>
      <StorageClass>Archive</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

- Console configurations

> **Note** The following figure shows the configurations of the preceding lifecycle rule in the OSS console: For more information, see Configure lifecycle rules.

  - The following figure shows the configurations of the preceding rule 1 in the OSS console:



Create Rule

Basic Settings

Status: Enabled [✓ Disabled]

Applied To: [✓ Files with Specified Prefix] Whole Bucket

Prefix: logs/

Tagging □

Clear Policy

File Lifecycle: [✓ Validity Period (Days)] Expiration Date Disabled

Transit to IA Storage Class ☑ 100

The storage class of objects is automatically converted from Standard to IA 100 days after the objects are last modified. The IA storage class is suitable for data that is infrequently accessed. You may be charged data retrieval fees when you access IA objects.

Transit to Archive Storage Class □ 30

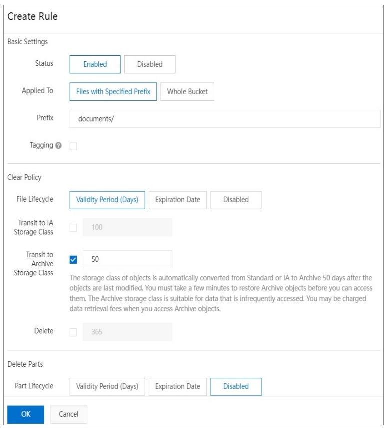
Delete □ 365

Delete Parts

Part Lifecycle: Validity Period (Days) Expiration Date [✓ Disabled]

OK Cancel

  - The following figure shows the configurations of the preceding rule 2 in the OSS console:



Create Rule

Basic Settings

Status: [✓ Enabled] Disabled

Applied To: [✓ Files with Specified Prefix] Whole Bucket

Prefix: documents/

Tagging □

Clear Policy

File Lifecycle: [✓ Validity Period (Days)] Expiration Date Disabled

Transit to IA Storage Class □ 100

Transit to Archive Storage Class ☑ 50

The storage class of objects is automatically converted from Standard or IA to Archive 50 days after the objects are last modified. You must take a few minutes to restore Archive objects before you can access them. The Archive storage class is suitable for data that is infrequently accessed. You may be charged data retrieval fees when you access Archive objects.

Delete □ 365

Delete Parts

Part Lifecycle: Validity Period (Days) Expiration Date [✓ Disabled]

OK Cancel

Configure lifecycle rules for versioning-enabled buckets

In a versioning-enabled bucket, each object has a current version and may have previous versions. For more information about versioning, see Overview.

- The lifecycle rule configured in this example specifies that the storage class of all objects in the bucket are converted to IA 10 days after they are last modified and to Archive 60 days after they become previous versions. In addition, objects are deleted 90 days after they become previous versions.

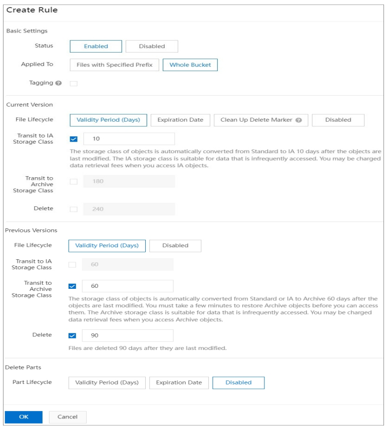
  The following examples show the XML and console configurations of the lifecycle rule:

- XML

```
<LifecycleConfiguration>
  <Rule>
    <ID>test-rule0</ID>
    <Prefix></Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>10</Days>
      <StorageClass>IA</StorageClass>
    </Transition>
    <NoncurrentVersionTransition>
      <NoncurrentDays>60</NoncurrentDays>
      <StorageClass>Archive</StorageClass>
    </NoncurrentVersionTransition>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>90</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

- Console configurations

> **Note** The following figure shows the configurations of the preceding lifecycle rule in the OSS console: For more information, see Configure lifecycle rules.



Create Rule

Basic Settings

Status: [✓ Enabled] Disabled

Applied To: Files with Specified Prefix [✓ Whole Bucket]

Tagging □

Current Version

File Lifecycle: [✓ Validity Period (Days)] Expiration Date Clean Up Delete Marker Disabled

Transit to IA Storage Class ☑ 10

The storage class of objects is automatically converted from Standard to IA 10 days after the objects are last modified. The IA storage class is suitable for data that is infrequently accessed. You may be charged data retrieval fees when you access IA objects.

Transit to Archive Storage Class □ 180

Delete □ 240

Previous Versions

File Lifecycle: [✓ Validity Period (Days)] Disabled

Transit to IA Storage Class □ 60

Transit to Archive Storage Class ☑ 60

The storage class of objects is automatically converted from Standard or IA to Archive 60 days after the objects are last modified. You must take a few minutes to restore Archive objects before you can access them. The Archive storage class is suitable for data that is infrequently accessed. You may be charged data retrieval fees when you access Archive objects.

Delete ☑ 90

Files are deleted 90 days after they are last modified.

Delete Parts

Part Lifecycle: Validity Period (Days) Expiration Date [✓ Disabled]

Example 1

- If the current version of an object is a delete marker and other versions of the object are deleted, the delete marker expires. You can configure the following lifecycle rule to delete expired delete markers:

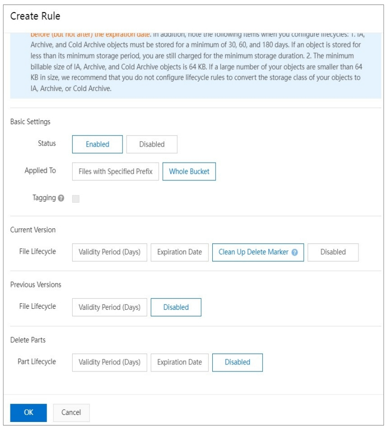
  The following examples show the XML and console configurations of the lifecycle rule:

  - XML

```
<LifecycleConfiguration>
  <Rule>
    <ID>test-rule0</ID>
    <Prefix></Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <ExpiredObjectDeleteMarker>true</ExpiredObjectDeleteMarker>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

- Console configurations

  Note The following figure shows the configurations of the preceding lifecycle rule in the OSS console: For more information, see Configure lifecycle rules.



Create Rule

before (but not after) the expiration date. In addition, note the following items when you configure lifecycles: 1. IA, Archive, and Cold Archive objects must be stored for a minimum of 30, 60, and 180 days. If an object is stored for less than its minimum storage period, you are still charged for the minimum storage duration. 2. The minimum billable size of IA, Archive, and Cold Archive objects is 64 KB. If a large number of your objects are smaller than 64 KB in size, we recommend that you do not configure lifecycle rules to convert the storage class of your objects to IA, Archive, or Cold Archive.

Basic Settings

Status: [✓ Enabled] Disabled

Applied To: Files with Specified Prefix [✓ Whole Bucket]

Tagging: □

Current Version

File Lifecycle: Validity Period (Days) Expiration Date [✓ Clean Up Delete Marker] Disabled

Previous Versions

File Lifecycle: Validity Period (Days) [✓ Disabled]

Delete Parts

Part Lifecycle: Validity Period (Days) Expiration Date [✓ Disabled]

Example 2

Configure lifecycle rules to delete expired parts

Assume that you use multipart upload to upload an object and do not call the CompleteMultipartUpload operation to complete the multipart upload task. In this case, you can configure a lifecycle rule that specifies that parts whose names contain the prefix logs expire 5 days after they are uploaded.

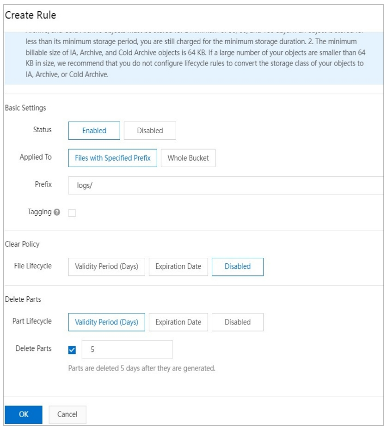
The following examples show the XML and console configurations of the lifecycle rule:

- XML

```
<LifecycleConfiguration>
  <Rule>
    <ID>lifecyclerule1</ID>
    <Prefix>logs/</Prefix>
    <Status>Enabled</Status>
    <AbortMultipartUpload>
      <Days>5</Days>
    </AbortMultipartUpload>
  </Rule>
</LifecycleConfiguration>
```

- Console configurations

  Note The following figure shows the configurations of the preceding lifecycle rule in the OSS console: For more information, see Configure lifecycle rules.



Create Rule

less than its minimum storage period, you are still charged for the minimum storage duration. 2. The minimum billable size of IA, Archive, and Cold Archive objects is 64 KB. If a large number of your objects are smaller than 64 KB in size, we recommend that you do not configure lifecycle rules to convert the storage class of your objects to IA, Archive, or Cold Archive.

Basic Settings

Status: [✓ Enabled] Disabled

Applied To: [✓ Files with Specified Prefix] Whole Bucket

Prefix: logs/

Tagging: □

Clear Policy

File Lifecycle: Validity Period (Days) Expiration Date [✓ Disabled]

Delete Parts

Part Lifecycle: [✓ Validity Period (Days)] Expiration Date Disabled

Delete Parts: ☑ 5

Parts are deleted 5 days after they are generated.

5.8. Bucket inventory

You can use the bucket inventory feature to export the information about specified objects in a bucket, such as the number, sizes, storage classes, and encryption status of the objects. Compared with the GetBucket (ListObjects) operation, we recommend that you use the bucket inventory feature to list a large number of objects.

Overview

After an inventory is configured for a bucket, Object Storage Service (OSS) generates inventory lists at the specified time interval. The following structure shows the directories in which generated inventory lists are stored.

```
dest_bucket
 └─destination-prefix/
     └─src_bucket/
         └─inventory_id/
             ├─YYYY-MM-DDTHH-MMZ/
             │   ├─manifest.json
             │   └─manifest.checksum
             └─data/
                 └─745a29e3-bfaa-490d-9109-47086afcc8f2.csv.gz
```

Directory	Description
destination-prefix/	This directory is generated based on the prefix specified for inventory lists. If no prefix is specified for inventory lists, this directory is omitted.
src_bucket/	This directory is generated based on the name of the source bucket for which inventory lists are generated.
inventory_id/	The directory is generated based on the name of the inventory.
YYY-MM-DDTHH-MMZ/	This directory indicates the start time when the bucket is scanned. The name of this directory is a timestamp in GMT. Example: 2020-05-17T16-00Z. The *manifest.json* and *manifest.checksum* objects are stored in this directory.
data/	Inventory lists that include the metadata of exported objects in the source bucket are stored in this directory. Inventory lists are CSV files that are compressed by using Gzip. Notice When large inventory lists are exported, they are split into multiple CSV files. The names of these CSV files are generated in the following format in sequence: *uuid.csv.gz*, *uuid-1.csv.gz*, and *uuid-2.csv.gz*. You can obtain the list of these CSV files from the *manifest.json* file. Then, you can decompress the files based on the preceding sequence to read the inventory lists.

After an inventory is configured for a bucket, the following objects are generated based on the inventory:

- Manifest objects

  Manifest objects include *manifest.json* and *manifest.checksum*.

  - *manifest.json*: stores the metadata of inventory lists and related information.

```
{
    "creationTimestamp": "1642994594",
    "destinationBucket": "destbucket",
    "fileFormat": "CSV",
    "fileSchema": "Bucket, Key, VersionId, IsLatest, IsDeleteMarker, Size, StorageClass, LastModifiedDate, ETag, IsMultipartUploaded, EncryptionS
tatus",
    "files": [{
            "MD5checksum": "F77449179760C3B13F1E76110F07****",
            "key": "destbucket/inventory0124/data/a1574226-b5e5-40ee-91df-356845777c04.csv.gz",
            "size": 2046}],
    "sourceBucket": "srcbucket",
    "version": "2019-09-01"}
```

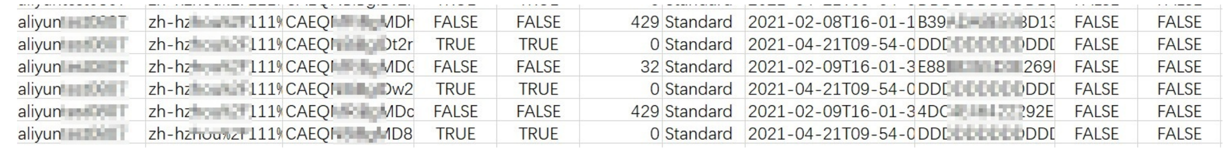
  The following table describes the fields included in the manifest.json file:

Field	Description
creationTimestamp	The start time when the source bucket is scanned. The value of this field is a UNIX timestamp.
destinationBucket	The destination bucket in which the inventory lists are stored.
fileFormat	The format of the inventory lists.
fileSchema	The fields contained in each inventory list.
files	This field contains the name, size, and MD5 hash of each inventory list.
sourceBucket	The source bucket for which the inventory lists are generated.
version	The version of the inventory list.

  - *manifest.checksum*: stores the MD5 hash of the *manifest.json* file. Example: `8420A430CBD6B659A1C0DFC1C11A****`.

- Inventory lists

  Inventory lists contain the exported object information and are stored in the *data/* directory. The following figure shows an example of an inventory list:



aliyun[redacted]T	zh-hz[redacted]111%	CAEQ[redacted]MDh	FALSE	FALSE	429	Standard	2021-02-08T16-01-1	B39[redacted]3D13	FALSE	FALSE
aliyun[redacted]T	zh-hz[redacted]111%	CAEQ[redacted]Dt2r	TRUE	TRUE	0	Standard	2021-04-21T09-54-0	DDD[redacted]DDD	FALSE	FALSE
aliyun[redacted]T	zh-hz[redacted]111%	CAEQ[redacted]MDG	FALSE	FALSE	32	Standard	2021-02-09T16-01-3	E88[redacted]269	FALSE	FALSE
aliyun[redacted]T	zh-hz[redacted]111%	CAEQ[redacted]Dw2	TRUE	TRUE	0	Standard	2021-04-21T09-54-0	DDD[redacted]DDD	FALSE	FALSE
aliyun[redacted]T	zh-hz[redacted]111%	CAEQ[redacted]MDc	FALSE	FALSE	429	Standard	2021-02-09T16-01-3	4DC[redacted]292E	FALSE	FALSE
aliyun[redacted]T	zh-hz[redacted]111%	CAEQ[redacted]MD8	TRUE	TRUE	0	Standard	2021-04-21T09-54-0	DDD[redacted]DDD	FALSE	FALSE

  The following table describes the fields in the preceding figure from left to right:

Field	Description
Bucket	The name of the bucket for which the inventory is created.

Field	Description
Key	The name of the object in the bucket. The object name is URL-encoded. You must decode the object name before you can view it.
VersionId	The version ID of the object. This field exists only when versioning is enabled for the bucket and the inventory specifies that all versions of data are exported.
IsLatest	This field indicates whether the version is the latest version. If the version is the latest version, the value is *True*. Otherwise, the value is *False*. This field exists only when versioning is enabled for the bucket and the inventory specifies that all versions of data are exported.
IsDeleteMarker	This field indicates whether the version is a delete marker. If the version is a delete marker, the value is *True*. Otherwise, the value is *False*. This field exists only when versioning is enabled for the bucket and the inventory specifies that all versions of data are exported.
Size	The size of the object.
StorageClass	The storage class of the object.
LastModifiedDate	The time when the object is last modified.
ETag	The ETag of the object. An ETag is generated when an object is created. ETags are used to identify the content of the objects. ○ If an object is created by calling PutObject, the ETag of the object is the MD5 hash value of the object content. ○ If an object is created by using other methods, the ETag of the object is not the MD5 hash of the object content but a unique value calculated based on the object.
IsMultipartUploaded	This field indicates whether the object is created by using multipart upload. If the object is created by using multipart upload, the value is *True*. Otherwise, the value is *False*.
EncryptionStatus	This field indicates whether the object is encrypted. If the object is encrypted, the value is *True*. Otherwise, the value is *False*.

Usage notes

- Permissions

  Resource Access Management (RAM) users must have permissions to configure inventories. If RAM users do not have the permissions in the following scenarios, you can perform the following operations to grant them permissions.

  - For a RAM user that has no permissions

    If a RAM user wants to use the bucket inventory feature when they have no permissions, perform the following steps to grant them permissions:

    a. Create the following custom policy by using the **JSON** method. For more information, see Create a custom policy.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "oss:PutBucketInventory",
                "oss:GetBucketInventory",
                "oss:DeleteBucketInventory",
                "oss:ListBuckets",
                "ram:CreateRole",
                "ram:AttachPolicyToRole",
                "ram:GetRole",
                "ram:ListPoliciesForRole"
            ],
            "Resource": "*"
        }
    ],
    "Version": "1"
}
```

    b. Attach the created custom policy to the RAM user. For more information, see Grant permissions to a RAM user.

- For a RAM user that has the `AliyunOSSFullAccess` permission
    - Attach a custom policy to the RAM user

      If a RAM user has the `AliyunOSSFullAccess` permission, you can use the **JSON** method to create the following custom policy and attach the custom policy to the RAM user:

      ```
      {
          "Statement": [
              {
                  "Effect": "Allow",
                  "Action": [
                      "ram:CreateRole",
                      "ram:AttachPolicyToRole",
                      "ram:GetRole",
                      "ram:ListPoliciesForRole"
                  ],
                  "Resource": "*"
              }
          ],
          "Version": "1"
      }
      ```

    - Grant the `AliyunRAMFullAccess` permission

      If a RAM user has the `AliyunOSSFullAccess` permission and does not want to use the bucket inventory feature by attaching custom policies, you can grant the `AliyunRAMFullAccess` permission to the RAM user. For more information about how to grant system permissions, see Grant permissions to a RAM user.

After you perform the preceding operations, use the OSS console to configure an inventory. After you configure the inventory, the RAM console automatically creates a RAM role named `AliyunOSSRole`. By default, the RAM role has permissions to read all objects from the source bucket and write objects to the destination bucket. If you want to use SDKs to use the bucket inventory feature, we recommend that you directly use the `AliyunOSSRole` RAM role created in the OSS console. This prevents other roles from being used to grant permissions.

- Recommended configurations

  We recommend that you configure inventories based on the number of objects in the source bucket:

    - If the number of objects in the source bucket is smaller than 1 billion, you can configure inventories to export inventory lists on a daily basis.
    - If the number of objects in the source bucket is between 1 billion and 10 billion, you can configure inventories to export inventory lists on a weekly basis.
    - If the number of objects in the source bucket is greater than 10 billion, we recommend that you configure different inventories based on object prefixes to generate inventory lists on a weekly basis and make sure the number of objects scanned based on each inventory does not exceed 10 billion.

- Traffic and bandwidth

  To increase the export speed of inventory lists, bucket- and user-level bandwidth may be occupied when the inventory lists are exported to the required bucket. If the destination bucket in which the exported inventory lists are stored is the source bucket for which the inventory is configured, and the source bucket is frequently accessed and the available bandwidth of the source bucket is limited, we recommend that you create a separate bucket to store the inventory lists.

- Exceptions
    - If no objects are stored in the bucket for which the inventory is configured or the specified prefix does not apply to objects in the inventory, inventory lists are not generated.
    - When you export the inventory lists, the exported lists may not contain all objects in the source bucket due to operations such as creation, deletion, or overwriting. If the last modified time of an object is earlier than the time specified by the createTimeStamp field in the *manifest.json* object, the inventory list contains information about the object. Otherwise, the inventory list may not contain information about the object. We recommend that you check the object attributes by calling HeadObject before you export the information about an object. For more information, see HeadObject.

Limits

- You can configure up to 1,000 inventories for a bucket by using OSS SDKs or ossutil. You can configure a maximum of 10 inventories in the OSS console.
- The source bucket for which an inventory is configured and the destination bucket in which the inventory lists are stored do not have to be the same bucket, but they must belong to the same Alibaba Cloud account and reside within the same region.

Billing

- You are charged for the bucket inventory feature. However, only storage fees for inventory lists and API calling fees are charged during public preview.
- OSS generates inventory lists based on the inventory until you delete the inventory. Fees are incurred for the storage of the inventory lists. To avoid unnecessary costs, delete inventory lists that are no longer needed.

Use the OSS console

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket that you want to manage.
3. In the left-side navigation pane, choose **Basic Settings > Bucket Inventory**. In the Bucket Inventory section, click **Configure**.
4. Click **Create Inventory**. In the **Create Inventory** panel, configure the parameters in the following table.

Parameter	Description
Status	Set the status of the inventory. You can select **Enabled** or **Disabled**.
Rule Name	Set the name of the inventory. The name can contain only lowercase letters, digits, and hyphens (-) and cannot start or end with a hyphen (-).
Destination Bucket	Select the bucket in which generated inventory lists are stored. The source bucket for which an inventory is configured and the destination bucket in which the inventory lists are stored do not have to be the same bucket, but they must belong to the same account and reside within the same region.

Parameter	Description
Inventory List Path	Configure the directory in which generated inventory lists are stored. - If you want to store the inventory lists in the root directory of the destination bucket, leave the parameter empty. - Otherwise, specify the parameter as a full path of a directory, excluding the destination bucket name. For example, when you want to store the inventory lists in the exampledir1 path of the destination bucket named examplebucket, set the parameter to *exampledir1*. When you want to store the inventory lists in the exampledir2 subdirectory of the exampledir1 directory, set the parameter to *exampledir1/exampledir2*.
Frequency	Configure the frequency at which inventory lists are generated. You can select **Weekly** or **Daily**. We recommend that you configure inventories based on the number of objects in the source bucket: - If the number of objects in the source bucket is smaller than 1 billion, we recommend that you configure inventories to export inventory lists on a daily basis. - If the number of objects in the source bucket is between 1 billion and 10 billion, we recommend that you configure inventories to export inventory lists on a weekly basis. - If the number of objects in the source bucket is greater than 10 billion, we recommend that you configure different inventories based on object prefixes to generate inventory lists on a weekly basis and make sure the number of objects scanned based on each inventory does not exceed 10 billion.
Encryption Method	Specify whether to encrypt inventory lists. - **None**: Inventory lists are not encrypted. - **AES-256**: Inventory lists are encrypted by using AES-256. - **KMS**: Inventory lists are encrypted by using a customer master key (CMK) managed by Key Management Service (KMS). To use a CMK to encrypt inventory lists, you must create a CMK in KMS in the same region as the destination bucket. For more information about how to configure CMKs, see Create a CMK. **Note** You are charged for calling API operations when you use CMKs to encrypt or decrypt data.
Object Versions	Select the object version to which the inventory is applied. If versioning is enabled for the bucket, you can select **Current Version** or **All Versions** to generate inventory lists for the current version or all versions of objects in the bucket. For more information, see Overview. By default, inventory lists are generated for all objects in the bucket if versioning is not enabled for the bucket.
Object Prefix	Set the prefix based on which to scan objects. - To scan all objects in the bucket, do not specify this parameter. - To scan all objects in a directory of the bucket, set this parameter to the full path that does not include the bucket name. To scan all objects in the exampledir1 root directory of the examplebucket bucket, set this parameter to *exampledir1/*. To scan all objects in the exampledir2 subdirectory of the exampledir1 root directory, set this parameter to *exampledir1/exampledir2/*. **Note** If no objects in the bucket match the specified prefix, no inventory lists are generated.
Optional Fields	Select the object information that you want to export to inventory lists. You can select the following fields: **Object Size**, **Storage Class**, **Last Update Time**, **ETag**, **Multipart Upload**, and **Encryption Status**.

5. Read and select **I understand the terms and agree to authorize Alibaba Cloud OSS to access the resources in my buckets.** Then, click **OK**.
   It may take a while to generate an inventory list for a large number of objects. If you want to be notified when the inventory list is generated for the objects, we recommend that you configure an event notification for the destination bucket in which the inventory list is stored and set the event to PutObject. When an inventory list is generated, a notification is sent to you. For more information about how to configure event notifications, see Configure event notification rules.

Use ossbrowser

ossbrowser supports the same operations related to buckets as the OSS console. You can follow the on-screen instructions in ossbrowser to configure an inventory for a bucket. For more information about how to use ossbrowser, see Use ossbrowser.

Use OSS SDKs

The following sample code provides examples on how to configure an inventory for a bucket by using OSS SDKs for common programming languages. For more information about how to configure an inventory for a bucket by using OSS SDKs for other programming languages, see Overview.

Java Python Go C++

```
// Specify the endpoint of the region in which the bucket is located. For example, if the bucket is located in the China (Hangzhou) region, set the endpoint to https://oss-cn-hangzhou.aliyuncs.com.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access Object Storage Service (OSS) because the account has permissions on all API operations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Specify the name of the bucket for which you want to add inventories.
String bucketName = "yourBucketName";
// Specify the name of the bucket in which you want to store the generated inventory lists.
String destBucketName ="yourDestinationBucketName";
// Specify the account ID granted by the bucket owner to add inventories for the bucket.
String accountId ="yourDestinationBucketAccountId";
// Specify the name of the RAM role. The RAM role must have the permissions to read the source bucket for which you want to configure the inventory and the permissions to write data to the destination bucket in which you want to store the generated inventory lists.
String roleArn ="yourDestinationBucketRoleArn";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Create an inventory configuration.
InventoryConfiguration inventoryConfiguration = new InventoryConfiguration();
// Specify the inventory name.
String inventoryId = "testid";
inventoryConfiguration.setInventoryId(inventoryId);
// Specify the object attributes to include in the inventory list.
List<String> fields = new ArrayList<String>();
fields.add(InventoryOptionalFields.Size);
fields.add(InventoryOptionalFields.LastModifiedDate);
fields.add(InventoryOptionalFields.IsMultipartUploaded);
fields.add(InventoryOptionalFields.StorageClass);
fields.add(InventoryOptionalFields.ETag);
fields.add(InventoryOptionalFields.EncryptionStatus);
inventoryConfiguration.setOptionalFields(fields);
// Specify whether to generate the inventory lists on a daily or weekly basis. The following code provides an example on how to generate inventory lists on a weekly basis. Weekly indicates once a week. Daily indicates once a day.
inventoryConfiguration.setSchedule(new InventorySchedule().withFrequency(InventoryFrequency.Weekly));
// Specify that the inventory lists include only the current version of objects. If you set the InventoryIncludedObjectVersions parameter to All, all versions of objects are included in the inventory lists. This configuration takes effect only when versioning is enabled for the bucket.
inventoryConfiguration.setIncludedObjectVersions(InventoryIncludedObjectVersions.Current);
// Specify whether inventory is enabled for the bucket. Valid values: true and false. If you set this parameter to true, the inventory is enabled.
inventoryConfiguration.setEnabled(true);
// Specify the rule used to filter the objects to include in the inventory lists. The following code provides an example on how to filter the objects by prefix.
InventoryFilter inventoryFilter = new InventoryFilter().withPrefix("obj-prefix");
inventoryConfiguration.setInventoryFilter(inventoryFilter);
// Configure the destination bucket.
InventoryOSSBucketDestination ossInvDest = new InventoryOSSBucketDestination();
// Specify the prefix of the path in which you want to store the generated inventory lists.
ossInvDest.setPrefix("destination-prefix");
// Specify the format of the inventory lists.
ossInvDest.setFormat(InventoryFormat.CSV);
// Specify the ID of the Alibaba Cloud account that owns the destination bucket.
ossInvDest.setAccountId(accountId);
// Specify the role ARN of the destination bucket.
ossInvDest.setRoleArn(roleArn);
// Specify the name of the destination bucket in which you want to store the generated inventory lists.
ossInvDest.setBucket(destBucketName);
// The following code provides an example on how to encrypt the inventory lists by using customer master keys (CMKs) hosted in Key Management System (KMS).
// InventoryEncryption inventoryEncryption = new InventoryEncryption();
// InventoryServerSideEncryptionKMS serverSideKmsEncryption = new InventoryServerSideEncryptionKMS().withKeyId("test-kms-id");
// inventoryEncryption.setServerSideKmsEncryption(serverSideKmsEncryption);
// ossInvDest.setEncryption(inventoryEncryption);
// The following code provides an example on how to encrypt the inventory lists on the OSS server.
// InventoryEncryption inventoryEncryption = new InventoryEncryption();
// inventoryEncryption.setServerSideOssEncryption(new InventoryServerSideEncryptionOSS());
// ossInvDest.setEncryption(inventoryEncryption);
// Specify the destination for the generated inventory lists.
InventoryDestination destination = new InventoryDestination();
destination.setOssBucketDestination(ossInvDest);
inventoryConfiguration.setDestination(destination);
// Apply the inventory configuration to the bucket.
ossClient.setBucketInventoryConfiguration(bucketName, inventoryConfiguration);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Use ossutil

For more information about how to configure an inventory for a bucket by using ossutil, see inventory (configure bucket inventories).

Use the RESTful API

If your program requires high customization, you can directly initiate a RESTful API request. In this case, you need to manually write code to calculate the signature. For more information, see PutBucketInventory.

FAQ

How do I know whether an inventory list is generated?

It may take a while to generate an inventory list for a large number of objects. If you want to be notified when the inventory list is generated for objects, we recommend that you configure an event notification for the destination bucket in which the inventory list is stored and set the event to PutObject. When an inventory list is generated, a notification is sent to you. For more information about how to configure an event notification, see Configure event notification rules.

5.9. Enable pay-by-requester

When pay-by-requester is enabled for a bucket, requesters pay the request and traffic fees that are incurred when the requesters access objects in the bucket. The bucket owner is charged only the storage fees of the objects. You can enable pay-by-requester to share your data in Object Storage Service (OSS) without having to pay for additional fees on your own.

Scenarios

- Share large datasets. For example, a research institute wants to share with its customers a public dataset that includes postal code directories, references data, geospatial information, or web crawling data. In addition, the research institute wants requesters to pay the incurred request and traffic fees.

  In this case, you can perform the following steps to configure pay-by-requester:

  i. Set the access control list (ACL) of the bucket in which the public dataset is stored to public read. For more information, see Configure the ACL of a bucket.

  ii. Enable pay-by-requester for the bucket.

- Deliver production data to your customers or partners. For example, a company must deliver production data to its partners and wants its partners to pay for the incurred request and traffic fees generated when they download the data.

  In this case, you can perform the following steps to configure pay-by-requester:

  i. Set the ACL of the bucket in which the production data is stored to private. For more information, see Configure the ACL of a bucket.

  ii. Enable pay-by-requester for the bucket.

  iii. Use bucket policies to grant your partners permissions to access the production data in the bucket. For more information, see Tutorial: Authorize a RAM user under another Alibaba Cloud account by adding a bucket policy.

  > **Notice** Make sure that you grant the Resource Access Management (RAM) users of your partners permissions to access the production data in the bucket. Do not share the AccessKey pair of RAM users under your Alibaba Cloud account to your partners. Otherwise, you are charged for the request and traffic fees because the requesters use the RAM users under your Alibaba Cloud account to access the data.

Request methods

- Requests from anonymous users are not allowed

  If you enable pay-by-requester for a bucket, anonymous users are not allowed to access the bucket. Requesters must provide authentication information so that OSS can identify and charge requesters for request and traffic fees.

  If a requester assumes a RAM role of an Alibaba Cloud account to request data, OSS charges the Alibaba Cloud account for the requests sent by the requester and the generated traffic.

- Requests must contain the `x-oss-request-payer` header

  If you enable pay-by-requester for a bucket, requesters must specify the `x-oss-request-payer` header in POST, GET, or HEAD requests. The value of the header is *requester*. This indicates that requesters understand that they are charged for the requests and downloaded data. Otherwise, the requests cannot be authenticated.

  - POST, GET, and HEAD requests must contain the `x-oss-request-payer:requester` header.
  - Requests that use signed URLs must contain the `x-oss-request-payer=requester` header.

  Bucket owners do not need to contain the `x-oss-request-payer` header in requests sent to access their buckets. The bucket owner is charged for their own requests and the generated traffic.

Billing

When pay-by-requester is enabled, requesters are charged for one or more of the following billing items based on the requested content: outbound traffic over the Internet, Content Delivery Network (CDN) back-to-origin traffic, and traffic generated when requesters call API operations, perform Image Processing (IMG), take video snapshots, and retrieve Infrequent Access (IA) objects or Archive objects. The bucket owner pays other fees such as storage, object tagging, and transfer acceleration fees. In the following scenarios, a request to a bucket with pay-by-requester enabled fails. In these cases, HTTP status code 403 is returned and the bucket owner is charged for the request.

- The POST, GET, or HEAD request does not contain the `x-oss-request-payer` header.
- The request fails to be authenticated.
- The request is anonymous.

Use the OSS console

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket that you want to manage.
3. In the left-side navigation bar, choose **Basic Settings > Pay by Requester**. In the **Pay by Requester** section, click **Configure** to enable or disable the pay-by-requester mode.
4. Click **Save**.

Use OSS SDKs

The following code provides examples on how to configure pay-by-requester for a bucket by using OSS SDKs for common programming languages. For more information about how to configure pay-by-requester for a bucket by using OSS SDKs for other programming languages, see Overview.

Python

```
import com.aliyun.oss.OSS;
import com.aliyun.oss.OSSClientBuilder;
import com.aliyun.oss.OSSException;
import com.aliyun.oss.model.*;
public class Demo {
    public static void main(String[] args) {
        // In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint. For more information about the endpoints of other regions, see Regions and endpoints.
        String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
        // Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API operations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
        String accessKeyId = "yourAccessKeyId";
        String accessKeySecret = "yourAccessKeySecret";
        // Specify the name of the bucket. Example: examplebucket.
        String bucketName = "examplebucket";
        // Create an OSSClient instance.
        OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
        try {
            // Enable pay-by-requester for the bucket.
            Payer payer = Payer.Requester;
            ossClient.setBucketRequestPayment(bucketName, payer);
        } catch (OSSException oe) {
            System.out.println("Caught an OSSException, which means your request made it to OSS, "
                    + "but was rejected with an error response for some reason.");
            System.out.println("Error Message:" + oe.getErrorMessage());
            System.out.println("Error Code:" + oe.getErrorCode());
            System.out.println("Request ID:" + oe.getRequestId());
            System.out.println("Host ID:" + oe.getHostId());
        } catch (Throwable ce) {
            System.out.println("Caught an ClientException, which means the client encountered "
                    + "a serious internal problem while trying to communicate with OSS, "
                    + "such as not being able to access the network.");
            System.out.println("Error Message:" + ce.getMessage());
        } finally {
            // Shut down the OSSClient instance.
            if (ossClient != null) {
                ossClient.shutdown();
            }
        }
    }
}
```

Use ossutil

For more information about how to use ossutil to configure pay-by-requester for a bucket, see Enable pay-by-requester.

Use the RESTful API

If your program requires more custom options to configure pay-by-requester, you can call RESTful API operations. In this case, you need to manually write code to calculate the signature. For more information, see PutBucketRequestPayment.

5.10. Map custom domain names

After you upload objects to a bucket, Object Storage Service (OSS) automatically generates URLs for the uploaded objects. You can use these URLs to access the objects. If you want to access the objects by using custom domain names, you must map the custom domain names to the bucket in which the objects are stored and add CNAME records for the custom domain names.

Usage notes

- In accordance with the administrative regulations of the People's Republic of China, you must file the custom domain names that you want to map to OSS buckets in mainland China regions at the Ministry of Industry and Information Technology (MIIT) in advance.

  For more information, see What is an ICP filing?.
- Up to 100 domain names can be mapped to a bucket. Each domain name can be mapped to only one bucket. OSS does not impose limits on the number of domain names that can be mapped to an Alibaba Cloud account.
- When you map a custom domain name to a bucket in the OSS console, the domain name cannot contain wildcards. When you map an accelerated domain name to a bucket, the domain name can contain wildcards but is not displayed in the OSS console.

Scenarios

- For image objects in a bucket created in mainland China regions after September 23, 2019, if you want to make sure that the image objects are previewed instead of downloaded when the objects are accessed by using a browser, map a custom domain name to the bucket.
- If you want to make sure that a bucket for which static website hosting is configured is not downloaded when the bucket is accessed, map a custom domain name to the bucket.

Implementation methods

You can map custom domain names to a bucket in the OSS console. For more information, see Map custom domain names.

Examples

For example, a public read object named *exampleobject.jpg* is stored in the root directory of the bucket named *examplebucket* in the China (Hangzhou) region. The custom domain name `www.example.com` is mapped to the bucket. Then, the URL of the bucket changes after the custom domain name is mapped.

- Before the custom domain name is mapped to the bucket

You can use the following URL that contains the default domain name of the bucket to access the *exampleobject.jpg* object: `https://examplebucket.oss-cn-hangzhou.aliyuncs.com/exampleobject.jpg`.

- After the custom domain name is mapped to the bucket

  You can use the following URL that contains the custom domain name mapped to the bucket to access the *exampleobject.jpg* object: `https://www.example.com/exampleobject.jpg`.

  > ⓘ **Note** After a custom domain name is mapped to a bucket, you can preview an image object in the bucket when you access the image object, and the object is not downloaded as an attachment. If the object cannot be previewed but is downloaded as an attachment, the cause may be that the browser does not support previews for some image formats. To solve this issue, you must install a plug-in that allows your browser to preview objects of the corresponding format.

References

- For more information about how to map an accelerate endpoint, see Bind accelerate endpoints.
- For more information about how to access OSS resources as a static website, see Configure static website hosting and Tutorial: Configure static website hosting by using a custom domain name.
- For more information about how to access OSS over HTTPS, see Host SSL certificates.

5.11. Transfer acceleration

Object Storage Service (OSS) uses data centers distributed around the globe to implement transfer acceleration. When a request is sent to your bucket, it is parsed and routed to the data center where the bucket is located over the optimal network path and protocol. The transfer acceleration feature provides an optimized end-to-end acceleration solution to accessing OSS over the Internet.

Prerequisites

You must complete real-name registration on the Real-name Registration page before you can enable transfer acceleration for a bucket.

Scenarios

- Accelerate remote data transfer

  For example, forums and online collaboration tools that provide services to users across the globe store data in OSS. However, upload and download speeds vary from region to region, which delivers an inconsistent user experience. OSS transfer acceleration allows users from different regions to access data in OSS over the optimal network path. This accelerates data transfer and improves user experience.

- Accelerate the uploads and downloads of gigabyte- and terabyte-grade objects

  When large objects are uploaded or downloaded over long geographical distances, transmission failures may occur due to high network latencies. Transfer acceleration combines optimal route selection, protocol stack tuning, and transmission algorithm optimization to reduce timeouts during remote transfer of large objects over the Internet. You can combine transfer acceleration with Multipart upload and Resumable download to develop a solution to uploading and downloading large objects over long distances.

- Accelerate the downloads of non-static and non-hot data

  User experience is a driving factor for product competitiveness and customer retention in applications that require high data download speeds, such as photo management applications, games, e-commerce applications, enterprise portal websites, and financial applications. High download speeds are also required to obtain comments on social networking applications. OSS transfer acceleration is a feature designed to accelerate uploads to and downloads from OSS. You can enable transfer acceleration to maximize bandwidth utilization and accelerate data transfer.

Usage notes

- Transfer acceleration takes effect within 30 minutes after it is enabled.
- Access is accelerated only when you use an accelerate endpoint to access a bucket that has transfer acceleration enabled.
- When you use an accelerate endpoint, you can manage only buckets that have transfer acceleration enabled.
- After you enable transfer acceleration for a bucket, other endpoints of the bucket remain available. In scenarios where transfer acceleration is not required, you can use the default endpoint to avoid unnecessary charges.
- Accelerate endpoints can be accessed only by API requests that use the HTTP or HTTPS protocol. Access by API requests that use other protocols such as RTMP is not supported.
- To ensure data security, the protocol used in transmission may be changed from HTTP to HTTPS after the peer end receives the request. Therefore, when the client uses an accelerate endpoint to access OSS over HTTP, the protocol recorded in access logs may be HTTPS.
- If you enable transfer acceleration and use an accelerate endpoint to access your bucket, OSS charges transfer acceleration fees. For more information about the billing methods of transfer acceleration, see Transfer acceleration fees.

Enable transfer acceleration

After you enable transfer acceleration for a bucket by using one of the following methods, you can access the bucket by using the following two endpoints in addition to the default endpoint.

- Global accelerate endpoint: *oss-accelerate.aliyuncs.com*. Transfer acceleration access points are distributed across the world. You can use this endpoint to accelerate data transfer for buckets in all regions.
- Accelerate endpoint of regions outside mainland China: *oss-accelerate-overseas.aliyuncs.com*. Transfer acceleration access points are distributed across regions outside mainland China. You can use these accelerate endpoints to map a custom domain name without an ICP filing to a bucket in the China (Hong Kong) region or a region outside mainland China.

Use the OSS console

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure a mirroring-based back-to-origin rule.
3. In the left-side navigation pane, choose **Transmission > Transfer Acceleration**.
4. Click **Configure**, turn on Transfer Acceleration, and then click **Save**.

Use OSS SDKs

Python Java

```
import com.aliyun.oss.ClientException;
import com.aliyun.oss.OSS;
import com.aliyun.oss.OSSClientBuilder;
import com.aliyun.oss.OSSException;
public class Demo {
    public static void main(String[] args) throws Exception {
        // In this example, the endpoint of the China (Hangzhou) region is used. Specify the endpoint based on your business requirements.
        String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
        // Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on al
l API operations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console
.
        String accessKeyId = "yourAccessKeyId";
        String accessKeySecret = "yourAccessKeySecret";
        // Specify the name of the bucket. Example: examplebucket.
        String bucketName = "examplebucket";
        // Create an OSSClient instance.
        OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
        try {
            // Configure transfer acceleration for the bucket.
            // If enabled is set to true, transfer acceleration is enabled. If enabled is set to false, transfer acceleration is disabled.
            boolean enabled = true;
            ossClient.setBucketTransferAcceleration(bucketName, enabled);
        } catch (OSSException oe) {
            System.out.println("Caught an OSSException, which means your request made it to OSS, "
                    + "but was rejected with an error response for some reason.");
            System.out.println("Error Message:" + oe.getErrorMessage());
            System.out.println("Error Code:" + oe.getErrorCode());
            System.out.println("Request ID:" + oe.getRequestId());
            System.out.println("Host ID:" + oe.getHostId());
        } catch (ClientException ce) {
            System.out.println("Caught an ClientException, which means the client encountered "
                    + "a serious internal problem while trying to communicate with OSS, "
                    + "such as not being able to access the network.");
            System.out.println("Error Message:" + ce.getMessage());
        } finally {
            if (ossClient != null) {
                ossClient.shutdown();
            }
        }
    }
}
```

Use the RESTful API

If your program requires high customization, you can directly initiate a RESTful API request. In this case, you need to manually write code to calculate the signature. For more information, see PutBucketTransferAcceleration.

Implementation methods

Use a browser

When you use a browser to access data stored in OSS, replace the endpoint in the object URL with an accelerate endpoint. For example, replace `https://test.oss-cn-shenzhen.aliyuncs.com/myphoto.jpg` with `https://test.oss-accelerate.aliyuncs.com/myphoto.jpg` . For more information about how to obtain the URL of an object, see How do I obtain the URL of an uploaded object?. If the access control list (ACL) of the object you want to access is private, you must sign the access request.

> **Note** To accelerate access to a bucket with transfer acceleration enabled and a custom domain name mapped, configure CNAME to map your custom domain name to an accelerate endpoint. For more information about how to configure mirroring-based back-to-origin rules, see Bind accelerate endpoints.

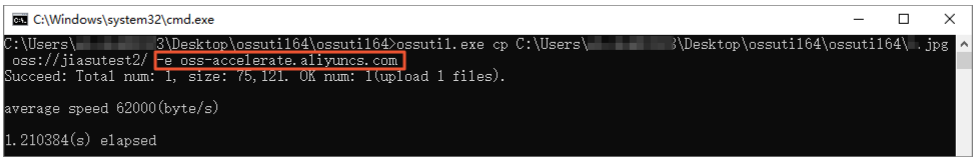
Use ossutil

- Replace the endpoint specified in the configuration file of ossutil with an accelerate endpoint

  When you use ossutil to access data stored in OSS, replace the endpoint specified in the configuration file with an accelerate endpoint. For more information, see ossutil.

- Add `-e oss-accelerate.aliyuncs.com` to commands

  When you run commands in ossutil, add the `-e oss-accelerate.aliyuncs.com` option to the commands. The following figure shows how to specify an accelerate endpoint when you run the cp command to upload an object by using ossutil.



```
C:\Users\[redacted]3\Desktop\ossutil64\ossutil64>ossutil.exe cp C:\Users\[redacted]3\Desktop\ossutil64\ossutil64\[redacted].jpg
 oss://jiasutest2/ -e oss-accelerate.aliyuncs.com
Succeed: Total num: 1, size: 75,121. OK num: 1(upload 1 files).

average speed 62000(byte/s)

1.210384(s) elapsed
```

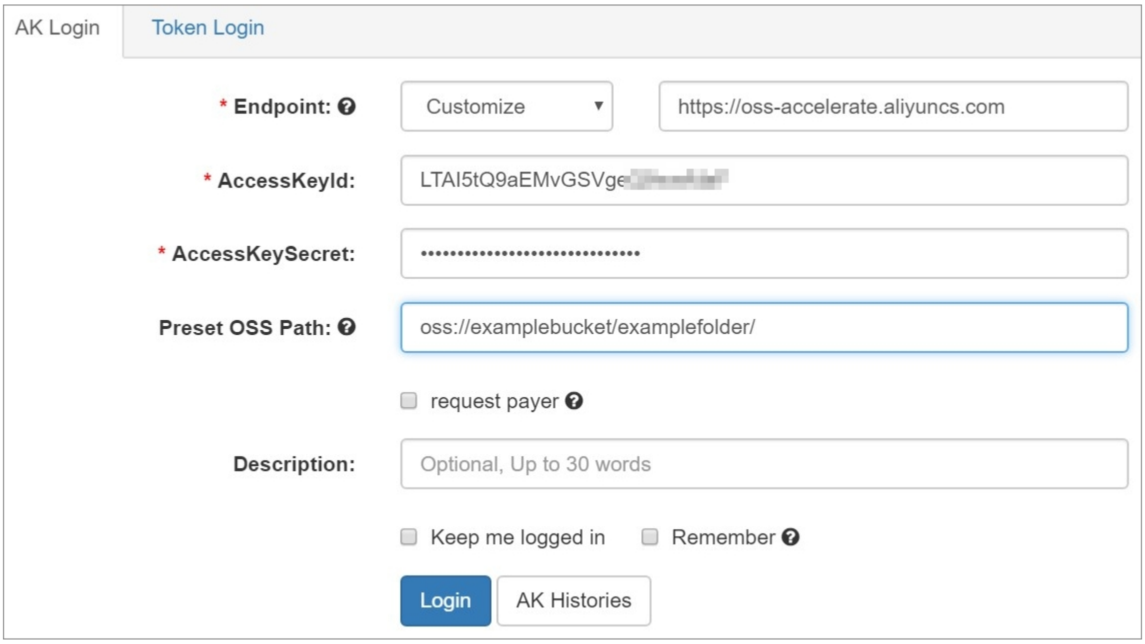
Use ossbrowser

> **Notice** When you access a bucket stored in OSS by using ossbrowser, you must specify your AccessKey pair and the preset path of the bucket in OSS.

The following table describes the parameters that you must configure when you use ossbrowser to access data stored in OSS.

Parameter	Description
Endpoint	Select **Customize** and enter *https://oss-accelerate.aliyuncs.com*.
AccessKeyId and AccessKeySecret	Enter the AccessKey pair of your account. For more information about how to obtain the AccessKey pair, see Obtain an AccessKey pair. **Notice** To ensure data security, we recommend that you log on to ossbrowser by using the AccessKey pair of a Resource Access Management (RAM) user. Before you use the AccessKey pair of a RAM user to log on to ossbrowser, you must grant the following permissions to the RAM user: `AliyunOSSFullAccess` , `AliyunRAMFullAccess` , and `AliyunSTSAssumeRoleAccess` . For more information, see Permission management.
Preset OSS Path	Specify permissions on a bucket or resources stored in the bucket. *oss://bucketname/path*. For example, if you are authorized to access only objects or subdirectories in the examplefolder directory of a bucket named examplebucket, enter *oss://examplebucket/examplefolder/*.

Sample configurations:



* Endpoint: Customize https://oss-accelerate.aliyuncs.com

* AccessKeyId: LTAI5tQ9aEMvGSVge[redacted]

* AccessKeySecret: ••••••••••••••••••••••••••••••

Preset OSS Path: oss://examplebucket/examplefolder/

□ request payer

Description: Optional, Up to 30 words

□ Keep me logged in □ Remember

Use OSS SDKs

When you use OSS SDKs for different programming languages to access OSS, set the endpoint parameter to an accelerate endpoint. The following code provides examples on how to specify an accelerate endpoint when you use OSS SDK for Java to perform simple upload and download.

- Simple upload

```
import com.aliyun.oss.ClientException;
import com.aliyun.oss.OSS;
import com.aliyun.oss.OSSClientBuilder;
import com.aliyun.oss.OSSException;
import com.aliyun.oss.model.PutObjectRequest;
import java.io.File;
public class Demo {
    public static void main(String[] args) throws Exception {
        // Specify an accelerate endpoint. In this example, the global accelerate endpoint is used.
        String endpoint = "https://oss-accelerate.aliyuncs.com";
        // Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API operations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
        String accessKeyId = "yourAccessKeyId";
        String accessKeySecret = "yourAccessKeySecret";
        // Specify the name of the bucket. Example: examplebucket.
        String bucketName = "examplebucket";
        // Specify the full path of the object. Example: exampledir/exampleobject.txt. The full path of the object cannot contain the bucket name.
        String objectName = "exampledir/exampleobject.txt";
        // Specify the full path of the local file. Example: D:\\localpath\\examplefile.txt.
        // By default, if the path of the local file is not specified, the local file is uploaded from the path of the project to which the sample program belongs.
        String filePath= "D:\\localpath\\examplefile.txt";
        // Create an OSSClient instance.
        OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
        try {
            // Create a PutObjectRequest object.
            PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, objectName, filePath);
            // Optional. Specify the storage class and the access control list (ACL) of the object.
            // ObjectMetadata metadata = new ObjectMetadata();
            // metadata.setHeader(OSSHeaders.OSS_STORAGE_CLASS, StorageClass.Standard.toString());
            // metadata.setObjectAcl(CannedAccessControlList.Private);
            // putObjectRequest.setMetadata(metadata);
            // Upload the object.
            ossClient.putObject(putObjectRequest);
        } catch (OSSException oe) {
            System.out.println("Caught an OSSException, which means your request made it to OSS, "
                    + "but was rejected with an error response for some reason.");
            System.out.println("Error Message:" + oe.getErrorMessage());
            System.out.println("Error Code:" + oe.getErrorCode());
            System.out.println("Request ID:" + oe.getRequestId());
            System.out.println("Host ID:" + oe.getHostId());
        } catch (ClientException ce) {
            System.out.println("Caught an ClientException, which means the client encountered "
                    + "a serious internal problem while trying to communicate with OSS, "
                    + "such as not being able to access the network.");
            System.out.println("Error Message:" + ce.getMessage());
        } finally {
            if (ossClient != null) {
                ossClient.shutdown();
            }
        }
    }
}
```

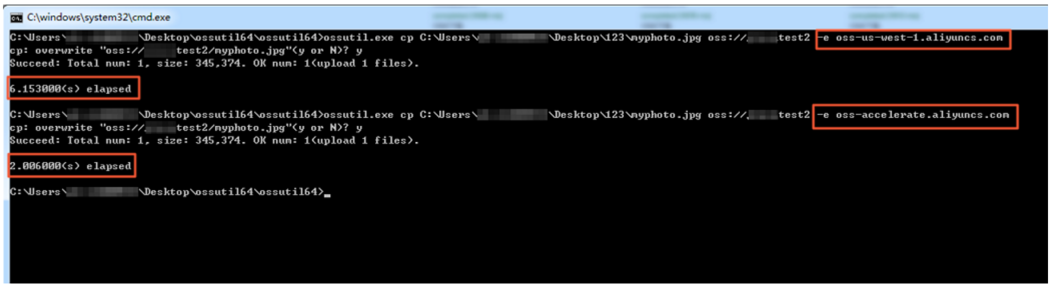
- Simple download

```
import com.aliyun.oss.ClientException;
import com.aliyun.oss.OSS;
import com.aliyun.oss.OSSClientBuilder;
import com.aliyun.oss.OSSException;
import com.aliyun.oss.model.GetObjectRequest;
import java.io.File;
public class Demo {
    public static void main(String[] args) throws Exception {
        // Specify an accelerate endpoint. In this example, the global accelerate endpoint is used.
        String endpoint = "https://oss-accelerate.aliyuncs.com";
        // Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on 
all API operations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM con
sole.
        String accessKeyId = "yourAccessKeyId";
        String accessKeySecret = "yourAccessKeySecret";
        // Specify the name of the bucket. Example: examplebucket.
        String bucketName = "examplebucket";
        // Specify the full path of the object. The full path of the object cannot contain bucket names. Example: testfolder/exampleobject.txt.
        String objectName = "testfolder/exampleobject.txt";
        String filePath = "D:\\localpath\\examplefile.txt";
        // Create an OSSClient instance.
        OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
        try {
            // Download the object to a local file in the specified path. If the specified local file exists, the object to download replaces the f
ile. Otherwise, the file is created.
            // If the path for the object is not specified, the downloaded object is saved to the path of the project to which the sample program b
elongs.
            ossClient.getObject(new GetObjectRequest(bucketName, objectName), new File(filePath));
        } catch (OSSException oe) {
            System.out.println("Caught an OSSException, which means your request made it to OSS, "
                    + "but was rejected with an error response for some reason.");
            System.out.println("Error Message:" + oe.getErrorMessage());
            System.out.println("Error Code:" + oe.getErrorCode());
            System.out.println("Request ID:" + oe.getRequestId());
            System.out.println("Host ID:" + oe.getHostId());
        } catch (ClientException ce) {
            System.out.println("Caught an ClientException, which means the client encountered "
                    + "a serious internal problem while trying to communicate with OSS, "
                    + "such as not being able to access the network.");
            System.out.println("Error Message:" + ce.getMessage());
        } finally {
            if (ossClient != null) {
                ossClient.shutdown();
            }
        }
    }
}
```

Test the effect of transfer acceleration

Use ossutil

You can run commands in ossutil to respectively upload an object to OSS by using the default endpoint and the accelerate endpoint. Then, compare the time that is used to upload the object to verify the effect of transfer acceleration. In the sample commands shown in the following figure, the `-e oss-us-west-1.aliyuncs.com` option is used to upload the object by using the default endpoint, and the `-e oss-accelerate.aliyuncs.com` option is used to upload the object by using the accelerate endpoint.



```
C:\Users\[redacted]\Desktop\ossutil64\ossutil64>ossutil.exe cp C:\Users\[redacted]\Desktop\123\myphoto.jpg oss://[redacted]test2 -e oss-us-west-1.aliyuncs.com
cp: overwrite "oss://[redacted]test2/myphoto.jpg"(y or N)? y
Succeed: Total num: 1, size: 345,374. OK num: 1(upload 1 files).

6.153000(s) elapsed

C:\Users\[redacted]\Desktop\ossutil64\ossutil64>ossutil.exe cp C:\Users\[redacted]\Desktop\123\myphoto.jpg oss://[redacted]test2 -e oss-accelerate.aliyuncs.com
cp: overwrite "oss://[redacted]test2/myphoto.jpg"(y or N)? y
Succeed: Total num: 1, size: 345,374. OK num: 1(upload 1 files).

2.006000(s) elapsed

C:\Users\[redacted]\Desktop\ossutil64\ossutil64>_
```

Use online tools

You can compare the access speeds when you use the accelerate and default endpoints to access OSS in different regions. For more information, see The Comparison of OSS Direct Data Transfer and Accelerated Data Transfer in Different Regions.

FAQ

- What are the differences between transfer acceleration and Alibaba Cloud Content Delivery Network (CDN)?

Feature	How it works	Scenario

Feature	How it works	Scenario
Transfer acceleration	Transfer acceleration provides an end-to-end acceleration solution by combining smart scheduling, protocol stack tuning, optimal route selection, and transmission algorithm optimization with OSS server-side configurations.	○ Accelerate object uploads. ○ Accelerate remote object uploads and downloads. ○ Accelerate the uploads and downloads of large objects. ○ Accelerate the downloads of dynamic objects and non-hot objects.
Alibaba Cloud CDN	Alibaba Cloud CDN uses OSS buckets as origin servers and distributes the content to edge nodes. When users request content, Alibaba Cloud CDN delivers content to the users from the nodes that are nearest to them. This way, content delivery is accelerated.	Accelerate content delivery when a large number of users in the same region concurrently request the same static content.

- How do I implement transfer acceleration by using a custom domain name?

  After you map a custom domain name to a bucket, you can map the CNAME to an accelerate endpoint. For more information, see Bind accelerate endpoints.

- Why am I unable to list buckets by using an accelerate endpoint?

  Transfer acceleration is applicable only to third-level domains that contain a specific bucket name, such as `https://BucketName.oss-accelerate.aliyuncs.com` . The domain name in a ListBuckets request does not contain a bucket name. Therefore, accelerate endpoints cannot be used to list buckets. To list buckets in a specific region, we recommend that you use the default endpoint of the region, such as `https://oss-cn-hangzhou.aliyuncs.com` .

5.12. Configure CORS

Cross-origin resource sharing (CORS) is a standard cross-origin solution provided by HTML5 to allow web application servers to control cross-origin access. This way, the security of data transmission across origins is ensured.

Background information

Browsers check cross-origin requests based on the same-origin policy to keep the website content secure. When a request is sent from Website A by using JavaScript to access Website B of another origin, the browser rejects the request. In this case, you can configure CORS rules to allow cross-origin requests.

Origins that use the same protocol, domain name or IP address, and port number are considered the same origin. The following table describes examples and checks whether the examples are from the same origin as *http://www.aliyun.com/org/test.html*.

URL	Access result	Cause
http://www.aliyun.com/org/other.html	Successful	Same protocol, domain name, and port number
http://www.aliyun.com/org/internal/page.html	Successful	Same protocol, domain name, and port
https://www.aliyun.com/page.html	Failed	Different protocols (HTTP and HTTPS)
http://www.aliyun.com:22/dir/page.html	Failed	Different port numbers (22 and 80)
http://www.alibabacloud.com/help/other.html	Failed	Different domain names

The preceding table shows that the browser denies requests whose protocols, domain names, or port numbers are different from those of the accessed origin. If you want to allow access from the origins, you must configure CORS rules.

CORS rules

OSS allows you to configure CORS rules to allow or deny cross-origin requests based on your requirements. CORS rules are configured only to decide whether to add CORS-related headers to requests. The browser decides whether to reject cross-origin requests. For more information, see PutBucketCORS.

You must set the `Vary: Origin` header to true in the following scenarios to avoid the issue that resources cannot be accessed due to different values of the Origin headers in the request and the local cache.

> **Notice** If you set the `Vary: Origin` header to true, access by using the browser or Content Delivery Network (CDN) back-to-origin requests may increase.

- CORS and non-CORS requests are sent at the same time

  For example, in the following code, a non-CORS request is created in the <img> field and a CORS request is created by using the fetch method:

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>CORS Test</title>
</head>
<body>
// Create a non-CORS request.
<img src="https://examplebucket.oss-cn-beijing.aliyuncs.com/exampleobject.txt" alt="">
<script>
  // Create a CORS request.
  fetch("https://examplebucket.oss-cn-beijing.aliyuncs.com/exampleobject.txt").then(console.log)
</script>
</body>
</html>
```

- The Origin header has multiple possible values

  For example, you can set the Origin header to `http://www.example.com` and `https://www.example.org` to allow CORS requests from these origins.

Implementation methods

Implementation method	Description
Console	A user-friendly and intuitive web application
Java SDK	SDK demos for various programming languages
Python SDK	
PHP SDK	
Go SDK	
C SDK	
.NET SDK	

FAQ

- What do I do if CORS errors occur when I use an accelerated domain name to access OSS?

  If a CORS error occurs when you use an accelerated domain name to access OSS, you must configure CORS rules in the CDN console. For more information, visit Configure CORS for Alibaba Cloud CDN.

- What do I do if the `Response to preflight request doesn't pass access control check: The value of the 'Access-Control-Allow-Origin' header in the response must not be the wildcard '*' when the request's credentials mode is 'include'.` error occurs when I send a cross-origin request?

  We recommend that you set `xhr.withCredentials` to false to resolve this error.

5.13. Bucket tagging

Object Storage Service (OSS) allows you to classify and manage buckets by using bucket tags. You can use this feature to list buckets that have specific tags and configure access control lists (ACLs) for buckets that have specific tags.

The bucket tagging feature uses a key-value pair to identify a bucket. You can add tags to buckets that are used for different purposes and manage the buckets by tags.

- Only the bucket owner or authorized Resource Access Management (RAM) users can configure tagging for the bucket. Otherwise, 403 Forbidden is returned with the error code AccessDenied.
- You can configure up to 20 tags for a bucket.
- Each tag must have a key. The key of a tag can be up to 64 characters in length and cannot start with `http://` , `https://` , or `Aliyun` . The key of a tag cannot be empty.
- The value of a tag can be up to 128 characters in length and can be empty.
- The key and value of a tag must be UTF-8-encoded.

Implementation methods

Implementation method	Description
Console	User-friendly and intuitive web application
ossutil	A high-performance command-line tool
Java SDK	SDK demos for various programming languages
Python SDK	
Go SDK	
C++ SDK	
Node.js SDK	
.NET SDK	

Instructions

After you add tags to buckets, you can manage multiple buckets that have the same tag. For example, you can list buckets that have the same tag and authorize RAM users to manage buckets that have the same tag.

- List buckets that have specific tags

  You can list buckets that have specific tags. For more information, see the following SDK demos:

  - Java SDK
  - Python SDK
  - Go SDK

- Authorize RAM users to manage buckets that have specific tags

  When you have a large number of buckets, you can classify buckets based on tags and configure RAM policy to authorize specific users to manage buckets that have specific tags. For example, you can configure the following RAM policy to authorize User A to list buckets that have the tagging configuration of keytest=valuetest tag:

```
{
    "Version": "1",
    "Statement": [
        {
            "Action": [
                "oss:ListBuckets"
            ],
            "Resource": [
                "acs:oss:*:193248792425xxxx:*"
            ],
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "oss:BucketTag/keytest": "valuetest"
                }
            }
        }
    ]
}
```

5.14. Back-to-origin types

5.14.1. Manage back-to-origin configurations

If you access data in a bucket for which no back-to-origin rule is configured and the data does not exist, 404 Not Found is returned. However, if you configure back-to-origin rules that contain the correct origin URLs, you can obtain the data from the origin based on the back-to-origin rules.

You can configure mirroring-based or redirection-based back-to-origin rules for hot migration and specific request redirection. For more information about API operations that are used to configure back-to-origin rules, see PutBucketWebsite.

Usage notes

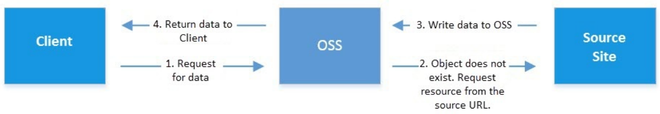
- You can configure up to 20 back-to-origin rules for a bucket. The rules are used to match a request in a sequence of their RuleNumber values. If a request matches a rule, subsequent rules are not used to match the request. Object Storage Service (OSS) determines whether a request matches a back-to-origin rule based on whether the request meets the conditions specified in the rule. OSS does not check whether the requested object can be obtained from the origin.
- You cannot set the origin URL in a back-to-origin rule to an address of the internal network.
- In regions within China, the default queries per second (QPS) for mirroring-based back-to-origin is 2,000, and the default bandwidth is 2 Gbit/s. In regions outside China, the default QPS for mirroring-based back-to-origin is 1,000, and the default bandwidth is 1 Gbit/s.
- You can add Image Processing (IMG) parameters to origin URLs in mirroring-based back-to-origin rules. However, snapshot parameters cannot be added to origin URLs in mirroring-based back-to-origin rules.
- By default, the timeout period of a mirroring-based back-to-origin request is 10 seconds.

Implementation methods

Implementation method	Description
Console	A user-friendly and intuitive web application
ossutil	A high-performance command-line tool

Mirroring-based back-to-origin

When you call the GetObject operation to query an object that does not exist in a bucket, OSS sends a request to the origin URL that is specified in the back-to-origin rule to retrieve the object. After OSS retrieves the object, OSS stores the object in the bucket and returns the object to you. The following figure shows the detailed process of mirroring-based back-to-origin.



- Scenarios

  Mirroring-based back-to-origin is used to seamlessly migrate data to OSS. This feature allows you to migrate services from a self-managed origin or from another cloud service to OSS without interrupting services. You can use mirroring-based back-to-origin to obtain the data that is not migrated to OSS during migration. This ensures service continuity. For detailed examples, see Seamlessly migrate data of a web-based service provider to OSS.

- Detail analysis
  - Trigger conditions

    OSS performs mirroring-based back-to-origin to retrieve an object from the origin only when 404 is returned for the GetObject operation.

  - Naming conventions

    The URL used by OSS to obtain an object from the origin is in the following format: `http(s)://MirrorURL+ObjectName` . ObjectName specifies the name of the requested object. For example, the origin URL configured for a bucket is `https://aliyun.com` , and the requested object named *example.jpg* does not exist in the bucket. OSS obtains the object by using `https://aliyun.com/example.jpg` and stores the object as *example.jpg*.

  - Rules for failed back-to-origin requests

    If the object that you request is not found in the origin, the origin returns HTTP status code 404 to OSS. Then, OSS returns the same HTTP status code to you. If the origin returns a non-200 HTTP status code to OSS to indicate an error such as an object retrieval failure due to network errors, OSS returns HTTP status code `424 MirrorFailed` to you.

- x-oss-tag response header

  When OSS returns an object that is obtained from the origin, OSS adds the `x-oss-tag` header to the response and sets the value of the header to `MIRROR`. The header is in the following format: `x-oss-tag:MIRROR`.

  After OSS obtains an object from the origin, this header is added to the response when the object is downloaded unless the object is overwritten. This header indicates that the object is obtained by using mirroring-based back-to-origin.

- Update rules of objects obtained by using mirroring-based back-to-origin

  After OSS obtains an object by using mirroring-based back-to-origin and the object is modified in the origin, OSS does not update the obtained object.

- Object metadata obtained by using mirroring-based back-to-origin

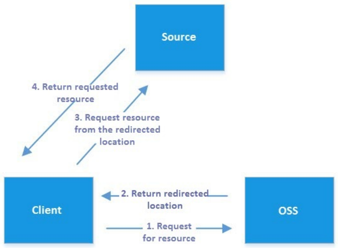
  OSS stores the following HTTP headers that are returned from the origin as the object metadata:

  ```
  Content-Type
  Content-Encoding
  Content-Disposition
  Cache-Control
  Expires
  Content-Language
  Access-Control-Allow-Origin
  ```

- HTTP request rules
  - Headers that are contained in the request sent to OSS are not contained in the request sent by OSS to the origin. OSS determines whether to send the QueryString information to the origin based on the back-to-origin rules that you configure in the OSS console.
  - If the origin returns chunked-encoded data, OSS also returns chunked-encoded data to you.

Redirection-based back-to-origin

Redirection-based back-to-origin enables OSS to return a redirect response and HTTP status code 3xx based on the specified back-to-origin conditions and corresponding redirection configurations. The following figure shows the detailed process of redirection-based back-to-origin.


Source
4. Return requested resource
3. Request resource from the redirected location
Client
2. Return redirected location
1. Request for resource
OSS

Scenarios:

- Seamless data migration from other data sources to OSS

  When you use redirection-based back-to-origin to asynchronously migrate data from your data source to OSS, OSS returns a 302 redirect request by using URL rewrite for the data that is not migrated to OSS. Then, your client can read the data from the data source based on the Location value in the 302 redirect request.

- Page redirection

  For example, you can use redirection-based back-to-origin to hide objects whose names contain a specified prefix and return a specific page to visitors.

- Page redirection for 404 or 500 errors

  You can use redirection-based back-to-origin to redirect users to a preset error page if a 404 or 500 error occurs. This method can prevent OSS from exposing system errors to users.

5.15. Delete buckets

You can delete a bucket that you no longer use.

ⓘ **Note** For more information about the API operation called to delete a bucket, see DeleteBucket.

Prerequisites

- All objects in the bucket are deleted. For more information about how to delete a small number of objects, see Delete objects. To delete a large number of objects, we recommend you configure lifecycle rules to batch delete the objects. For more information about how to delete a large number of objects, see Configure lifecycle rules.

  If versioning is enabled for the bucket you want to delete, make sure that all versions of objects in the bucket are deleted. For more information about how to delete all versions of objects in a bucket, see Configure versioning.

- Parts generated by multipart upload or resumable upload tasks in the bucket are deleted. For more information about how to delete parts, see Manage parts.
- All LiveChannels in the bucket are deleted. For more information about how to delete LiveChannel, see DeleteLiveChannel .

Warning You cannot recover deleted buckets. Exercise caution when you delete a bucket.

Implementation methods

Implementation method	Description
Console	A user-friendly and intuitive web application

Implementation method	Description
ossbrowser	An easy-to use graphical tool
ossutil	A high-performance command-line tool
Java SDK	SDK demos for various programming languages
Python SDK	
PHP SDK	
Go SDK	
C SDK	
.NET SDK	
Android SDK	
iOS SDK	
Node.js SDK	
Ruby SDK	

FAQ

- What do I do if I am unable to delete a bucket?

  The bucket cannot be deleted because the bucket is not empty, or you are not authorized to delete the bucket. Troubleshoot this issue based on following causes:

  - The bucket is not empty.

    Make sure that all objects, parts and LiveChannels are deleted from the bucket.

  - You are not authorized to delete the bucket.

    The bucket is empty, but you are not authorized to delete the bucket as a Resource Access Management (RAM) user. Obtain permissions based on the following causes:

    - You do not have the `oss:DeleteBucket` permission: If you cannot delete the bucket as a RAM user, request a RAM user who has the administrator permissions to add the `oss:DeleteBucket` permission to the RAM policy for you.
    - The Deny statement is configured for the `oss:DeleteBucket` permission in the bucket policy: If the `oss:DeleteBucket` permission is added to the RAM policy but the bucket still cannot be deleted, the bucket policy contains the Deny statement for the `oss:DeleteBucket` permission. You must change Deny to Allow or directly delete the bucket policy. Then, you can delete the bucket.

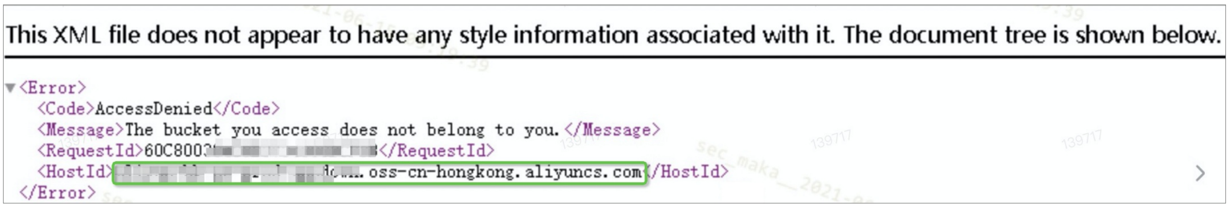
- Why am I unable to create a bucket that has the same name as that of an existing bucket after I delete the existing bucket?

  You are allowed to create a bucket that has the same name as the deleted bucket only about 30 minutes after you delete the object.

5.16. FAQ

5.16.1. How do I hide the error information returned from OSS when I retrieve an object from OSS by using CDN?

When you use Alibaba Cloud CDN or a third party CDN to retrieve an object from Object Storage Service (OSS), a similar error is returned if the endpoint mapped to the specified bucket is incorrect. The error information contains the public endpoint of OSS. The public endpoint of OSS is exposed, which causes information leakage and security risks.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<Error>
   <Code>AccessDenied</Code>
   <Message>The bucket you access does not belong to you.</Message>
   <RequestId>60C8003[redacted]</RequestId>
   <HostId>[redacted]... oss-cn-hongkong.aliyuncs.com</HostId>
 </Error>
```

To hide the public endpoint of OSS, use the following methods:

- Use static website hosting to configure the default 404 page

  Use static website hosting to configure the default 404 page. Example of the 404 page: `https://help.error.html` . After you configure the 404 page, OSS returns the default 404 page if the OSS resource you want to access by using CDN does not exist. For more information, see Configure static website hosting.

- Use CDN EdgeScript to rewrite or redirect requests

  For example, when you send a request to OSS and OSS returns the HTTP 403 status code, you can use the following EdgeScript rule to redirect the request to the back-to-origin and cached URL `https://www.error.html` :

  ```
  $status = get_status()
  if eq($status,403){
    rewrite('https://www.error.html', 'redirect')
  }
  ```

  You can replace the status code and the URL to which the request is redirected in the preceding EdgeScript rule with actual values.

  For more information, see Customize rewrites or redirects.

5.16.2. Why am I unable to access the default homepage of a bucket when I retrieve an object from a private bucket by using CDN?

Cause: After CDN is enabled for a private bucket, signature information is included by default for non-anonymous access when you retrieve an object from OSS by using CDN. However, to access the default homepage configured by using static website hosting, the request must be anonymous.

Solution: Use the CDN console to configure the rewrite rule. In the Create Rewrite Rule dialog box, set Original URI to `^/$`, Final URI to `/index.html`, and Flag to **Redirect**. After the rule is configured, the CDN node returns 302 and `www.example.com/index.html` when the client requests `www.example.com/`.

For more information about how to configure URI rewrite rules, see Create a URI rewrite rule.

6.Objects

6.1. Overview

Objects are the basic unit for data storage in Object Storage Service (OSS). Objects are also known as files. OSS stores all elements as objects in buckets, and does not organize the objects in a hierarchical structure.

Object types

Objects can be classified into the following three types based on how they are created:

- Normal: Objects of this type are created by using simple upload.
- Multipart: Objects of this type are created by using multipart upload.
- Appendable: Objects of this type are created by using append upload. You can append content to objects only of the Appendable type.

Object information

An object consists of the following information:

- Key: the name of the object. You can use the object key to query the object.
- Data: the content that is stored in the object. The content is a sequence of bytes.
- Version ID: After you upload an object to a versioned bucket, OSS generates a version ID for the object.
- Object metadata: the metadata of the object. Object metadata is a set of key-value pairs that define the attributes of the object, such as the time when the object is last modified and the object size. You can add custom information to object metadata.

Access control

OSS allows you to use the following access control methods to manage access to objects in buckets: bucket policies, access control lists (ACLs), Resource Access Management (RAM) policies, temporary access authorization based on Security Token Service (STS), and hotlink protection. Bucket policies and ACLs are implemented based on resources. RAM policies are implemented based on users. Hotlink protection is implemented by using whitelists. For more information about access control, see Overview.

Authorized access

By default, the ACLs of OSS resources such as buckets and objects are private. To allow unauthorized users to access these resources, you must grant permissions on these resources to the users. For example, if you store image and video resources of your websites in OSS buckets, you can use one of the following methods to authorize third-party users to access the resources:

- Set the ACLs of the resources to public-read. For more information, see Configure ACL for objects.
- Sign the URLs that are used to access the resources. For more information, see Authorize third-party users to download objects.

6.2. Object naming conventions

Object Storage Service (OSS) uses a flat structure instead of a hierarchical structure used by traditional file systems to store objects. All elements in OSS are stored as objects in buckets. Objects are the basic unit for data operations in OSS. Objects are also known as files. OSS uses a key (name) to uniquely identify an object.

Naming conventions

The name of an object must comply with the following conventions:

- The name can contain only UTF-8 characters.
- The name must be 1 to 1,023 bytes in length.
- The name cannot start with a forward slash (/) or a backslash (\).

Examples

The key of an object varies with the location in which the object is stored. The following table describes the keys of two objects that are stored in different locations of a bucket.

Object	Key
An object that is named exampleobject.txt and is stored in the root directory of a bucket named examplebucket	exampleobject.txt
An object that is named exampleobject.jpg and is stored in the destdir directory within the root directory of a bucket named examplebucket	destdir/exampleobject.jpg

6.3. Upload files

6.3.1. Simple upload

You can call the PutObject operation to upload a single object smaller than 5 GB. This is referred to as simple upload. Simple upload is suitable for scenarios in which an object can be uploaded by sending a single HTTP request.

Prerequisites

A bucket is created. For more information, see Create buckets.

Precautions

- Object size

  The size of the object that you can upload by means of simple upload cannot exceed 5 GB. To upload an object larger than 5 GB, use multipart upload. For more information, see Multipart upload and resumable upload.

- Naming conventions
    - The name must be encoded in UTF-8.
    - The name must be 1 to 1,023 characters in length.
    - The name cannot start with a forward slash (/) or a backslash (\).
- Upload security and authorization

  To prevent unauthorized third-party users from uploading data to your bucket, Object Storage Service (OSS) provides bucket-level and object-level access control. For more information, see Overview.

  To authorize third-party users to upload objects to your buckets, OSS also provides account-level authorization. For more information, see Authorized third-party upload.
- Prevent existing objects from being overwritten by objects with the same names

  By default, when you upload an object to OSS, an existing object with the same name is overwritten by the uploaded object. You can use the following methods to prevent your objects from being unexpectedly removed:
    - Enable versioning

      If versioning is enabled, overwritten objects are saved as previous versions. You can restore an object to a previous version at any time. For more information, see Overview.
    - Include a specific parameter in the upload request

      Include the x-oss-forbid-overwrite parameter in the upload request header and set the parameter to *true*. This way, if you upload an object that has the same name as an existing object, the upload fails and OSS returns the `FileAlreadyExists` error. If this parameter is not included in the request header, or this parameter is set to *false*, the object that has the same name is overwritten.
- Optimize object upload performance

  If you upload a large number of objects with sequential prefixes such as timestamps and letters in the object names, many object indexes may be stored in a single partition. In this case, if you send a large number of requests to query these objects, the latency may increase. We recommend that you use random prefixes but not sequential prefixes for object names when you upload a large number of objects. For more information, see OSS performance and scalability best practices.

Use the OSS console

Note In Alibaba Finance Cloud, OSS does not have a region connected to the Internet. Therefore, objects cannot be uploaded by using the OSS console. To upload objects, you must use OSS SDKs, ossutil, or ossbrowser.

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket to which you want to upload objects.
3. In the left-side navigation pane, click the **Files** tab. On the page that appears, click **Upload**.
4. In the **Upload** panel, configure the parameters described in the following table.

i. The following table describes the basic settings.

Parameter	Description
Upload To	Set the path in which to store an object after the object is uploaded to the bucket. ■ **Current**: Objects are uploaded to the current directory. ■ **Specified**: Objects are uploaded to the specified directory. You must enter the directory name. If the directory whose name you entered does not exist, OSS automatically creates the directory and uploads the object to the directory. The directory must meet the following naming conventions: ■ The name can contain only UTF-8 characters. The name must be 1 to 254 characters in length. ■ The name cannot start with a forward slash (/) or backslash (\). ■ The name cannot contain consecutive forward slashes (/). ■ The name of the directory cannot be `..`.
File ACL	Set the access control list (ACL) for the object. ■ **Inherited from Bucket**: The ACL of the object is the same as that of the bucket. ■ **Private**: Only the object owner or authorized users can read and write the objects to upload. Other users, including anonymous users, cannot access the objects without authorization. We recommend that you set the File ACL parameter to this value. ■ **Public Read**: Only the owner or authorized users of this bucket can write the objects to upload. Other users, including anonymous users, can only read the objects. If you set the File ACL parameter to this value, the objects may be unexpectedly accessed, which results in out-of-control costs. ■ **Public Read/Write**: All users, including anonymous users, can read and write the objects to upload. If you set the File ACL parameter to public read/write, the objects may be unexpectedly accessed, which results in out-of-control costs. If a user uploads prohibited data or information, your legitimate interests and rights may be infringed. Therefore, we recommend that you do not set the File ACL parameter to public read/write except in special cases. For more information about object ACLs, see Object ACL.
Upload Acceleration	After transfer acceleration is enabled for the bucket that contains the object, you can turn on **Upload Acceleration** if you want to accelerate the upload of the object. For more information about transfer acceleration, see Transfer acceleration.
Files to Upload	Select the files or directories that you want to upload. You can click **Select Files** to select a local file or click **Select Folders** to select a directory. You can also drag the required local file or directory to the Files to Upload section. If you select an unnecessary object, click **Remove** in the Actions column that corresponds to the object to remove the object. Notice ■ When you upload an object that has the same name as an existing object in OSS to an unversioned bucket, the existing object is overwritten. ■ When you upload an object that has the same name as an existing object in OSS to a versioned bucket, the existing object becomes a previous version, and the uploaded object becomes the latest version.

ii. (Optional) Configure advanced settings such as Storage Class and Encryption Method.

Parameter	Description
Storage Class	Set the storage class of the object. ■ **Inherited from Bucket**: The storage class of the object is the same as that of the bucket. ■ **Standard**: Standard is suitable for objects that are frequently accessed. ■ **IA**: Infrequent Access (IA) is suitable for objects that are less frequently accessed. On average, objects that are accessed less than once to twice a month fall into this category. IA objects have a minimum storage duration of 30 days. You are charged for data retrieval when you access these objects. ■ **Archive**: Archive is suitable for objects that are infrequently accessed. Archive objects have a minimum storage duration of 60 days. Before you can access an object of the Archive storage class, you must restore the object. The restoration takes about one minute, and data retrieval fees are incurred during the restoration process. ■ **Cold Archive**: Cold Archive is suitable for long-term storage of backup objects and raw data. Cold Archive objects have a minimum storage duration of 180 days. Before you can access an object of the Cold Archive storage class, you must restore the object. The amount of time required to restore a Cold Archive object depends on the data size and the restore mode. You are charged for the data retrieval when you restore a Cold Archive object. For more information, see Overview.
Encryption Method	Configure server-end encryption method for an object. ■ **Inherited from Bucket**: The encryption method of the object is the same as that of the bucket. ■ **OSS-Managed**: Keys managed by OSS are used to encrypt objects in the bucket. OSS uses data keys to encrypt objects. In addition, OSS uses regularly rotated master keys to encrypt data keys. ■ **KMS**: The default CMK stored in Key Management Service (KMS) or the specified CMK ID is used to encrypt and decrypt data. Descriptions of **CMK**: ■ **alias/acs/oss**: The default customer master key (CMK) stored in KMS is used to encrypt different objects and decrypt the objects when the objects are downloaded. ■ CMK ID: The keys generated by a specified CMK are used to encrypt different objects and the specified CMK ID is recorded in the metadata of the encrypted object. Objects are decrypted when they are downloaded by users who are granted decryption permissions. Before you specify a CMK ID, you must create a normal key or an external key in the same region as the bucket in the KMS console. ■ **Encryption algorithm**:Only AES-256 is supported. For more information about object ACLs, see Object ACL.
User Metadata	Add the descriptive information for the object. You can add multiple pieces of user metadata as custom headers. However, the total size of the user metadata cannot exceed 8 KB. When you add user metadata, you must prefix parameters with `x-oss-meta-` and specify a value such as **x-oss-meta-location:hangzhou** for the parameters.

iii. Click **Upload**.
You can view the upload progress of objects on the **Task List** tab.

Use ossbrowser

Operations related to buckets supported by ossbrowser are similar to those supported by the OSS console. You can follow the guidelines listed on the ossbrowser interface to complete simple upload operations. For more information about how to use ossbrowser, see Use ossbrowser.

Use OSS SDKs

The following code provides an example on how to perform simple upload by using OSS SDKs for common programming languages. For more information about how to perform simple upload by using OSS SDKs for other programming languages, see Overview.

[illegible]

```
import com.aliyun.oss.ClientException;
import com.aliyun.oss.OSS;
import com.aliyun.oss.OSSClientBuilder;
import com.aliyun.oss.OSSException;
import com.aliyun.oss.model.PutObjectRequest;
import java.io.File;
public class Demo {
    public static void main(String[] args) throws Exception {
        // In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
        String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
        // Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API operations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
        String accessKeyId = "yourAccessKeyId";
        String accessKeySecret = "yourAccessKeySecret";
        // Specify the name of the bucket. Example: examplebucket.
        String bucketName = "examplebucket";
        // Specify the full path of the object. Example: exampledir/exampleobject.txt. The full path of the object cannot contain the bucket name.
        String objectName = "exampledir/exampleobject.txt";
        // Specify the full path of the local file. Example: D:\\localpath\\examplefile.txt.
        // By default, if the path of the local file is not specified, the local file is uploaded from the path of the project to which the sample program belongs.
        String filePath= "D:\\localpath\\examplefile.txt";
        // Create an OSSClient instance.
        OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
        try {
            // Create a PutObjectRequest object.
            PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, objectName, filePath);
            // Optional. Specify the storage class and the ACL of the object.
            // ObjectMetadata metadata = new ObjectMetadata();
            // metadata.setHeader(OSSHeaders.OSS_STORAGE_CLASS, StorageClass.Standard.toString());
            // metadata.setObjectAcl(CannedAccessControlList.Private);
            // putObjectRequest.setMetadata(metadata);
            // Upload the object.
            ossClient.putObject(putObjectRequest);
        } catch (OSSException oe) {
            System.out.println("Caught an OSSException, which means your request made it to OSS, "
                    + "but was rejected with an error response for some reason.");
            System.out.println("Error Message:" + oe.getErrorMessage());
            System.out.println("Error Code:" + oe.getErrorCode());
            System.out.println("Request ID:" + oe.getRequestId());
            System.out.println("Host ID:" + oe.getHostId());
        } catch (ClientException ce) {
            System.out.println("Caught an ClientException, which means the client encountered "
                    + "a serious internal problem while trying to communicate with OSS, "
                    + "such as not being able to access the network.");
            System.out.println("Error Message:" + ce.getMessage());
        } finally {
            if (ossClient != null) {
                ossClient.shutdown();
            }
        }
    }
}
```

Use ossutil

For more information about how to perform simple upload by using ossutil, see Simple upload.

Use the RESTful API

If your program requires more custom options to configure pay-by-requester, you can call RESTful API operations. In this case, you must write code to calculate the signature. For more information, see PutObject.

References

- When you use simple upload, you can configure object metadata to describe an object. For example, you can set standard HTTP headers such as Content-Type. You can also configure user metadata. For more information about object metadata, see Manage object metadata.
- After an object is uploaded to OSS, you can use upload callback to send a callback request to a specified application server. For more information, see Upload callback.
- After an image object is uploaded, you can also compress the image object and configure custom styles for the image object. For more information, see IMG implementation modes.

6.3.2. Multipart upload

Alibaba Cloud Object Storage Service (OSS) provides multipart upload so that you can split up large objects into multiple parts and upload the parts separately. After the parts are uploaded, you can call the CompleteMultipartUpload operation to combine these parts into an object.

Prerequisites

A bucket is created. For more information, see Create buckets.

Scenarios

- Accelerated upload of large objects

When the object that you want to upload is larger than 5 GB, you can use multipart upload to split the object into multiple parts and concurrently upload the parts to accelerate the upload.

- Poor network environments

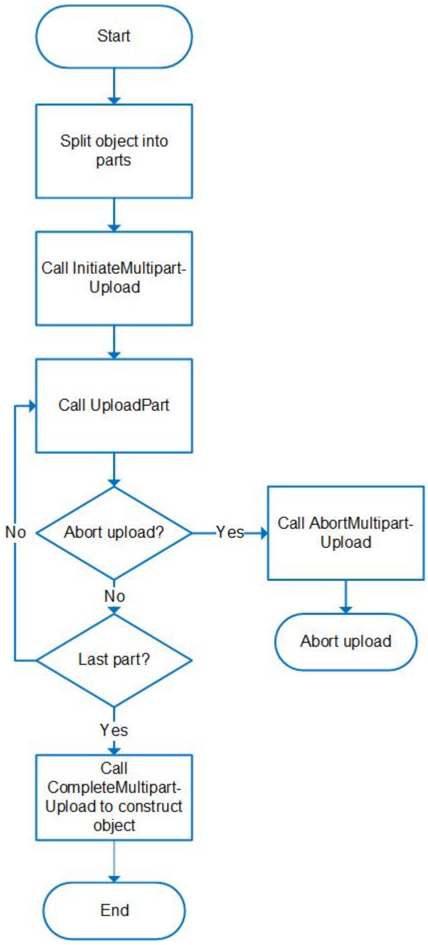
  We recommend that you use multipart upload when network conditions are unstable. When specific parts fail to be uploaded, you need only to upload these parts.

- Uncertain object size

  If you are uncertain of the size of objects to be uploaded, you can use multipart upload. This case is common in industry applications such as video surveillance.

Multipart upload process

The following flowchart shows the basic process of multipart upload.


Start
Split object into parts
Call InitiateMultipart-Upload
Call UploadPart
Abort upload?
Yes
No
Call AbortMultipart-Upload
Abort upload
No
Last part?
Yes
Call CompleteMultipart-Upload to construct object
End

The preceding process consists of the following steps:

1. Split the object that you want to upload into parts based on a specific size.
2. Call the InitiateMultipartUpload operation to initiate a multipart upload task.
3. Call the UploadPart operation to upload the parts.

   After the object is split into parts, a `partNumber` is specified for each part to indicate the sequence of the parts. Therefore, you can concurrently upload the parts in sequence. More concurrent uploads do not necessarily result in faster upload speeds. Therefore, we recommend that you specify the number of concurrent uploads based on your network conditions and the workload of your devices.

   If you want to cancel a multipart upload task, you can call the AbortMultipartUpload operation. After the multipart upload task is canceled, parts that are uploaded by the task are also deleted.

4. Call the CompleteMultipartUpload operation to combine the uploaded parts into an object.

Limits

Item	Limit
Object size	Multipart upload supports objects up to 48.8 TB in size.
Number of parts	You can set the number of parts to a value that ranges from 1 to 10,000.
Part size	Each part can be 100 KB to 5 GB in size. The size of the last part is not limited.
Maximum number of parts that can be returned for a single ListParts request	Up to 1,000 parts can be returned for a single ListParts request.
Maximum number of multipart upload tasks that can be returned for a single ListMultipartUploads request	Up to 1,000 tasks can be returned for a single ListParts request.

Precautions

- Optimize object upload performance

  If you upload a large number of objects whose names have sequential prefixes such as timestamps and letters, multiple object indexes may be stored in a single partition. If a large number of requests are sent to query these objects, the latency increases. We recommend that you do not upload a large number of objects that have sequential prefixes. For more information, see OSS performance and scalability best practices.

- Overwrite objects

  If you upload an object whose name is the same as an existing object in OSS, the existing object is overwritten. You can use the following methods to prevent objects from being unexpectedly overwritten:

  - Enable versioning

    When versioning is enabled for a bucket, overwritten objects are saved as previous versions. You can restore an object to a previous version at any time. For more information, see Overview.

  - Include a specific parameter in the upload request

    Include the x-oss-forbid-overwrite parameter in the upload request header and set the parameter to `true`. This way, if you upload an object whose name is the same as an existing object, the upload fails and OSS returns the `FileAlreadyExists` error. For more information, see InitiateMultipartUpload.

- Delete parts

  When a multipart upload task is interrupted, parts that are uploaded by the task are stored in the specified bucket. To avoid additional storage fees, we recommend that you use the following methods to delete these parts if you no longer use these parts:

  - Manually delete parts. For more information, see Manage parts.
  - Configure lifecycle rules to automatically delete parts. For more information, see Configure lifecycle rules.

Use OSS SDKs

The following code provides examples on how to perform multipart upload by using OSS SDKs for common programming languages. For more information about how to perform multipart upload by using OSS SDKs for other programming languages, see Overview.

[illegible]

```
import com.aliyun.oss.ClientException;
import com.aliyun.oss.OSS;
import com.aliyun.oss.OSSClientBuilder;
import com.aliyun.oss.OSSException;
import com.aliyun.oss.model.*;
import java.io.File;
import java.io.FileInputStream;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;
public class Demo {
    public static void main(String[] args) throws Exception {
        // In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
        String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
        // Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on al
l API operations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console
.
        String accessKeyId = "yourAccessKeyId";
        String accessKeySecret = "yourAccessKeySecret";
        // Specify the name of the bucket. Example: examplebucket.
        String bucketName = "examplebucket";
        // Specify the full path of the object. Example: exampledir/exampleobject.txt. The full path of the object cannot contain the bucket name.
        String objectName = "exampledir/exampleobject.txt";
        // Create an OSSClient instance.
        OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
        try {
            // Create an InitiateMultipartUploadRequest object.
            InitiateMultipartUploadRequest request = new InitiateMultipartUploadRequest(bucketName, objectName);
            // The following code provides an example on how to specify the request headers when you initiate a multipart upload task.
            // ObjectMetadata metadata = new ObjectMetadata();
            // metadata.setHeader(OSSHeaders.OSS_STORAGE_CLASS, StorageClass.Standard.toString());
            // Specify the caching behavior of the web page for the object.
            // metadata.setCacheControl("no-cache");
            // Specify the name of the object when the object is downloaded.
            // metadata.setContentDisposition("attachment;filename=oss_MultipartUpload.txt");
            // Specify the encoding format for the content of the object.
            // metadata.setContentEncoding(OSSConstants.DEFAULT_CHARSET_NAME);
            // Specify whether existing objects are overwritten by objects with the same names when the multipart upload task is initiated. In this e
xample, this parameter is set to true, which indicates that the object with the same name cannot be overwritten.
            // metadata.setHeader("x-oss-forbid-overwrite", "true");
            // Specify the server-side encryption method that is used to encrypt each part of the object to upload.
            // metadata.setHeader(OSSHeaders.OSS_SERVER_SIDE_ENCRYPTION, ObjectMetadata.KMS_SERVER_SIDE_ENCRYPTION);
            // Specify the encryption algorithm that is used to encrypt the object. If you do not configure this parameter, objects are encrypted by
using AES-256.
            // metadata.setHeader(OSSHeaders.OSS_SERVER_SIDE_DATA_ENCRYPTION, ObjectMetadata.KMS_SERVER_SIDE_ENCRYPTION);
            // Specify the ID of the customer master key (CMK) that is managed by Key Management Service (KMS).
            // metadata.setHeader(OSSHeaders.OSS_SERVER_SIDE_ENCRYPTION_KEY_ID, "9468da86-3509-4f8d-a61e-6eab1eac****");
            // Specify the storage class of the object.
            // metadata.setHeader(OSSHeaders.OSS_STORAGE_CLASS, StorageClass.Standard);
            // Configure tagging for the object. You can specify multiple tags for the object at the same time.
            // metadata.setHeader(OSSHeaders.OSS_TAGGING, "a:1");
            // request.setObjectMetadata(metadata);
            // Initiate a multipart copy task
            // Initiate a multipart copy task.
            InitiateMultipartUploadResult upresult = ossClient.initiateMultipartUpload(request);
            // Obtain the upload ID, which uniquely identifies the multipart upload task. You can use the upload ID to cancel or query the multipart
upload task.
            String uploadId = upresult.getUploadId();
            // partETags is the set of PartETags. A PartETag consists of the part number and ETag of an uploaded part
            List<PartETag> partETags =  new ArrayList<PartETag>();
            // The size of each part, which is used to calculate the number of parts of the object. Unit: bytes.
            final long partSize = 1 * 1024 * 1024L;   // Set the part size to 1 MB.
            // Specify the full path of the local file that you want to upload. By default, if you do not specify the full path of the local file, th
e local file is uploaded from the path of the project to which the sample program belongs.
            final File sampleFile = new File("D:\\localpath\\examplefile.txt");
            long fileLength = sampleFile.length();
            int partCount = (int) (fileLength / partSize);
            if (fileLength % partSize != 0) {
                partCount++;
            }
            // Upload each part until all parts are uploaded.
            for (int i = 0; i < partCount; i++) {
                long startPos = i * partSize;
                long curPartSize = (i + 1 == partCount) ? (fileLength - startPos) : partSize;
                InputStream instream = new FileInputStream(sampleFile);
                // Skip the parts that have been uploaded.
                instream.skip(startPos);
                UploadPartRequest uploadPartRequest = new UploadPartRequest();
                uploadPartRequest.setBucketName(bucketName);
                uploadPartRequest.setKey(objectName);
                uploadPartRequest.setUploadId(uploadId);
                uploadPartRequest.setInputStream(instream);
                // Configure the size available for each part. Each part except for the last part must be larger than 100 KB in size.
                uploadPartRequest.setPartSize(curPartSize);
                // Set part numbers. Each part has a part number. The number ranges from 1 to 10000. If the specified number is beyond the range, OSS
returns an InvalidArgument error code.
                uploadPartRequest.setPartNumber( i + 1);
                // Parts are not necessarily uploaded in sequence. They can be uploaded from different OSS clients. OSS sorts the parts based on thei
r part numbers and combines them into a complete object.
                UploadPartResult uploadPartResult = ossClient.uploadPart(uploadPartRequest);
                // Each time a part is uploaded, OSS returns a result that contains a PartETag. The PartETags are stored in partETags.
                partETags.add(uploadPartResult.getPartETag());
            }
            // Create a CompleteMultipartUploadRequest object.
            // When the multipart upload task is completed, you must provide all valid PartETags. After receiving the PartETags, OSS verifies the val
idity of all parts one by one. After all parts are verified, OSS combines these parts into a complete object.
            CompleteMultipartUploadRequest completeMultipartUploadRequest =
                    new CompleteMultipartUploadRequest(bucketName, objectName, uploadId, partETags);
            // Optional. The following code provides an example on how to set the access control list (ACL) of the object.
            // completeMultipartUploadRequest.setObjectACL(CannedAccessControlList.Private);
            // Specifies whether to list all parts that are uploaded by using the current upload ID. If you want to combine the parts by listing the
parts in the server side, you have the option to leave partETags contained in CompleteMultipartUploadRequest empty.
            // Map<String, String> headers = new HashMap<String, String>();
            // If x-oss-complete-all:yes is specified in the request, OSS lists all parts that are uploaded by using the current upload ID, sorts the
parts by part number, and then performs the CompleteMultipartUpload operation.
            // If you configure x-oss-complete-all:yes in the request, the request body cannot be specified. Otherwise, an error occurs.
            // headers.put("x-oss-complete-all","yes");
            // completeMultipartUploadRequest.setHeaders(headers);
            // Complete the multipart upload task.
            CompleteMultipartUploadResult completeMultipartUploadResult = ossClient.completeMultipartUpload(completeMultipartUploadRequest);
            System.out.println(completeMultipartUploadResult.getETag());
        } catch (OSSException oe) {
            System.out.println("Caught an OSSException, which means your request made it to OSS, "
                    + "but was rejected with an error response for some reason.");
            System.out.println("Error Message:" + oe.getErrorMessage());
            System.out.println("Error Code:" + oe.getErrorCode());
            System.out.println("Request ID:" + oe.getRequestId());
            System.out.println("Host ID:" + oe.getHostId());
        } catch (ClientException ce) {
            System.out.println("Caught an ClientException, which means the client encountered "
                    + "a serious internal problem while trying to communicate with OSS, "
                    + "such as not being able to access the network.");
            System.out.println("Error Message:" + ce.getMessage());
        } finally {
            if (ossClient != null) {
                ossClient.shutdown();
            }
        }
    }
}
```

Use ossutil

For more information about how to perform multipart upload by using ossutil, see Upload objects.

Use the RESTful API

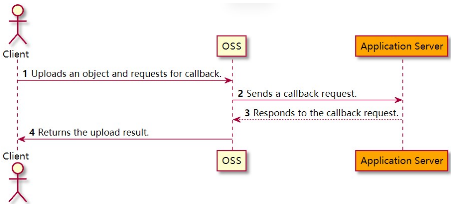
If your program requires more custom options to perform multipart upload, you can call RESTful API operations. In this case, you must manually write code to calculate the signature. For more information, see InitiateMultipartUpload.

6.3.3. Upload callback

After an object is uploaded, Object Storage Service (OSS) can start a callback process for the application server. To implement callback, you need only to send a request that contains relevant callback parameters to OSS.

Scenarios

Upload callback is used together with authorized third-party upload. You can simplify the client logic and save network resources by using upload callback properly. The following figure shows how upload callback works.


Client
OSS
Application Server
1 Uploads an object and requests for callback.
2 Sends a callback request.
3 Responds to the callback request.
4 Returns the upload result.
Client
OSS
Application Server

1. The client specifies upload callback in an upload request sent from the client to OSS.
2. After the upload task is completed, OSS sends an HTTP request for upload callback to the application server.
3. The application server receives the request that notifies the application server of the upload operation. Then, the application server performs operations such as modifying the database and responds to the OSS request.
4. After OSS receives the response, OSS returns the upload result to the client.

When OSS sends a POST callback request to the application server, OSS includes parameters in the POST request body to carry specific information. The parameters consist of system-defined parameters, such as those used to specify the bucket name and the object name, and custom parameters used to carry some information related to the application logic, such as the ID of the user who initiated the upload request.

Usage notes

Only simple upload (by calling the PutObject operation), form upload (by calling the PostObject operation), and multipart upload (by calling the CompleteMultipartUpload operation) support upload callback.

Use OSS SDKs

Python

```
import com.aliyun.oss.OSS;
import com.aliyun.oss.OSSClientBuilder;
import com.aliyun.oss.OSSException;
import com.aliyun.oss.model.*;
import java.io.ByteArrayInputStream;
public class Demo {
    public static void main(String[] args) {
        // In this example, the endpoint of the China (Hangzhou) region is used. Specify the actual endpoint.
        String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
        // Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on al
l API operations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console
.
        String accessKeyId = "yourAccessKeyId";
        String accessKeySecret = "yourAccessKeySecret";
        // Specify the name of the bucket. Example: examplebucket.
        String bucketName = "examplebucket";
        // Specify the full path of the object. Example: exampledir/exampleobject.txt. The full path of the object cannot contain the bucket name.
        String objectName = "exampledir/exampleobject.txt";
        // Specify the address of the server to which the callback request is sent. Example: https://example.com:23450 or https://127.0.0.1:9090.
        String callbackUrl = "yourCallbackServerUrl";
        // Create an OSSClient instance.
        OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
        try {
            String content = "Hello OSS";
            PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, objectName,new ByteArrayInputStream(content.getBytes()));
            // Upload callback parameters.
            Callback callback = new Callback();
            callback.setCallbackUrl(callbackUrl);
            // Optional. Set the value of the Host field included in the callback request header.
            // callback.setCallbackHost("yourCallbackHost");
            // Set the value of the body field included in the callback request.
            callback.setCallbackBody("{\\\"mimeType\\\":${mimeType},\\\"size\\\":${size}}");
            // Set Content-Type for the callback request.
            callback.setCalbackBodyType(Callback.CalbackBodyType.JSON);
            // Configure custom parameters for the callback request. Each custom parameter consists of a key and a value. The key must start with x:.

            callback.addCallbackVar("x:var1", "value1");
            callback.addCallbackVar("x:var2", "value2");
            putObjectRequest.setCallback(callback);
            PutObjectResult putObjectResult = ossClient.putObject(putObjectRequest);
            // Read the message content returned from upload callback.
            byte[] buffer = new byte[1024];
            putObjectResult.getResponse().getContent().read(buffer);
            // You must close the stream after the data is read. Otherwise, connection leaks may occur. Consequently, no connections are available an
d an exception occurs.
            putObjectResult.getResponse().getContent().close();
        } catch (OSSException oe) {
            System.out.println("Caught an OSSException, which means your request made it to OSS, "
                    + "but was rejected with an error response for some reason.");
            System.out.println("Error Message:" + oe.getErrorMessage());
            System.out.println("Error Code:" + oe.getErrorCode());
            System.out.println("Request ID:" + oe.getRequestId());
            System.out.println("Host ID:" + oe.getHostId());
        } catch (Throwable ce) {
            System.out.println("Caught an ClientException, which means the client encountered "
                    + "a serious internal problem while trying to communicate with OSS, "
                    + "such as not being able to access the network.");
            System.out.println("Error Message:" + ce.getMessage());
        } finally {
            if (ossClient != null) {
                ossClient.shutdown();
            }
        }
    }
}
```

Use the RESTful API

If your program requires more custom options to configure upload callback, you can call RESTful API operations. In this case, you must manually write code to calculate the signature. For more information, see Callback.

References

- For more information about the common errors and causes when you configure upload callback, see Upload callback.
- For more information about how to obtain signature information from the server in various programming languages based on POST policies, configure upload callback, and then directly upload data to OSS by using form upload, see Overview.
- For more information about how to set up an OSS-based direct data transfer service for mobile apps and configure upload callback, see Set up upload callback for mobile apps.

6.3.4. Authorized third-party upload

This topic describes how to use a signed URL or a temporary access credential from Security Token Service (STS) to grant a third-party user permissions to upload objects directly to Object Storage Service (OSS).

Prerequisites

A bucket is created. For more information, see Create buckets.

Background information

In a standard client/server system architecture, the server is responsible for receiving and processing requests from the client, and OSS is used as a backend storage service. In that case, the client sends the objects to upload to the application server. Then, the server forwards the objects to OSS. In this process, objects are transmitted twice: from the client to the server and from the server to OSS. In the case of bursts of access requests, the server must provide sufficient bandwidth resources for multiple clients to upload objects simultaneously. This presents a challenge to the architecture scalability.

Benefits

To resolve this issue, OSS provides authorized third-party upload. This way, each client can upload objects directly to OSS without transmitting them to the server. This reduces the cost of application servers and maximizes the OSS capability to process large amounts of data. Furthermore, you can focus on your business without worries about bandwidth and concurrency limits.

You have the option to use a signed URL or a temporary access credential to grant a third-party user permissions to upload objects directly to OSS.

Temporary access credential

You can use Alibaba Cloud STS to authorize temporary access to OSS. STS is a web service that provides temporary access tokens for users. You can use STS to grant an access credential that has a custom validity period and custom permissions for a third-party application or a Resource Access Management (RAM) user managed by you. For more information about STS, see What is STS?.

STS has the following benefits:

- You need only to generate an access token and send the access token to a third-party application, instead of exposing your AccessKey pair to the third-party application. You can customize the access permissions and validity period of this token.
- The access token automatically expires after the validity period. Therefore, you do not need to manually revoke the permissions of an access token.

> **Note** You can call the AssumeRole operation or use STS SDKs for various programming languages to obtain a temporary access credential. The temporary access credential contains a security token and a temporary AccessKey pair. The AccessKey pair consists of an AccessKey ID and an AccessKey secret. The minimum validity period of a temporary access credential is 900 seconds. The maximum validity period of a temporary access credential is the maximum session duration specified for the current role. For more information, see Specify the maximum session duration for a RAM role.

Use OSS SDK

The following code provides examples on how to use a temporary access credential to grant a third-party user permissions to directly upload objects to OSS by using OSS SDKs for common programming languages. For more information about how to use OSS SDKs for other programming languages to perform this operation, see Overview.

Java

```
// Specify the endpoint of the region in which the bucket is located. For example, if the bucket is located in the China (Hangzhou) region, set the e
ndpoint to https://oss-cn-hangzhou.aliyuncs.com.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// Specify the temporary AccessKey pair obtained from STS.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Specify the security token obtained from STS.
String securityToken = "yourSecurityToken";
// Specify the name of the bucket. Example: examplebucket.
String bucketName = "examplebucket";
// Specify the full path of the object. The path cannot contain the bucket name. Example: exampledir/exampleobject.txt.
String objectName = "exampledir/exampleobject.txt";
// You can use the AccessKey pair and security token contained in the temporary access credential obtained from STS to create an OSSClient.
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret, securityToken);
// Upload the object by using the temporary access credential that is obtained from STS.
// Specify the full path of the local file to upload. By default, if you do not specify the local file, the local file is uploaded from the path of t
he project to which the sample program belongs.
PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, objectName, new File("D:\\localpath\\examplefile.txt"));
ossClient.putObject(putObjectRequest);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Signed URL

> **Notice** A validity period must be set for an STS temporary access credential and a signed URL. When you use an STS temporary account credential to generate a signed URL that is used to perform operations such as object upload and download, the minimum validity period takes precedence. For example, you can set the validity period of your STS temporary access credential to 1,200 seconds and the validity period of your signed URL generated by using this credential to 3,600 seconds. In this case, the signed URL cannot be used to upload objects after the STS temporary access credential expires, even if the signed URL is within its validity period.

You can generate a signed URL and provide the URL to a visitor for temporary access. When you generate a signed URL, you can specify the validity period of the URL to limit the period of access from visitors. By default, the validity period of a signed URL is 3,600 seconds. The maximum validity period of a signed URL is 32,400 seconds.

You can add signature information to a URL and provide the URL to a third-party user for authorized access. For more information, see Add signatures to a URL.

Use OSS SDK

The following code provides examples on how to use a signed URL to grant a third-party user permissions to directly upload objects to OSS by using OSS SDKs for common programming languages. For more information about how to use OSS SDKs for other programming languages to perform this operation, see Overview.

Java

```
// Set yourEndpoint to the endpoint of the region in which the bucket is located. For example, to create a bucket in the China (Hangzhou) region, set
yourEndpoint to https://oss-cn-hangzhou.aliyuncs.com.
String endpoint = "yourEndpoint";
// Specify the temporary AccessKey pair obtained from STS.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Specify the security token obtained from STS.
String securityToken = "yourSecurityToken";
// Specify the name of the bucket. Example: examplebucket.
String bucketName = "examplebucket";
// Specify the full path of the object. Example: exampleobject.txt. The full path of the object cannot contain the bucket name.
String objectName = "exampleobject.txt";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret, securityToken);
// Specify the expiration date of the signed URL.
Date expiration = null;
try {
    expiration = DateUtil.parseRfc822Date("Wed, 18 Mar 2022 14:20:00 GMT");
} catch (ParseException e) {
    e.printStackTrace();
}
// Generate a signed URL.
GeneratePresignedUrlRequest request = new GeneratePresignedUrlRequest(bucketName, objectName, HttpMethod.PUT);
// Set the validity period of the signed URL.
request.setExpiration(expiration);
// Set ContentType.
request.setContentType("application/txt");
// Add user metadata.
request.addUserMetadata("author", "aliy");
// Generate a signed URL that allows HTTP PUT requests.
URL signedUrl = ossClient.generatePresignedUrl(request);
System.out.println("signed url for putObject: " + signedUrl);
// Use the signed URL to send a request.
// Specify the full path of the local file to upload. By default, if you do not specify the local file, the local file is uploaded from the path of t
he project to which the sample program belongs.
File f = new File("D:\\localpath\\examplefile.txt");
FileInputStream fin = null;
try {
    fin = new FileInputStream(f);
} catch (FileNotFoundException e) {
    e.printStackTrace();
}
// Add headers to the PutObject request.
Map<String, String> customHeaders = new HashMap<String, String>();
customHeaders.put("Content-Type", "application/txt");
customHeaders.put("x-oss-meta-author", "aliy");
PutObjectResult result = ossClient.putObject(signedUrl, fin, f.length(), customHeaders);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

6.3.5. Form upload

To use form upload, you can call the PostObject operation to upload an object whose size does not exceed 5 GB.

> **Note** For more information about the PostObject operation, see PostObject.

Scenarios

You can use form upload on HTML web pages to upload objects. For example, you can use form upload in web applications. The following table describes the comparison between the form upload process and other upload processes on a job-search website.

Upload method	Other upload methods	Form upload
Access process	1. A website user sends a request to upload a resume. 2. The website server responds with a resume upload page. 3. The resume is uploaded to the website server. 4. The website server uploads the resume to Object Storage Service (OSS).	1. A website user sends a request to upload a resume. 2. The website server responds with a resume upload page. 3. The resume is uploaded to OSS.

- The form upload process is easier, because data is directly uploaded to OSS without being forwarded by the website server.
- In other upload processes, objects are uploaded to the website server first. When a large number of objects are uploaded, the website server must be scaled out. In the form upload process, objects are directly uploaded from the client to OSS. When a large number of objects are uploaded, the service quality can be ensured by OSS.

SDK demo

For more information, see Form upload in OSS SDK for Java.

Usage notes

- Object size

The size of the object that you can upload by using form upload cannot exceed 5 GB. If the object that you want to upload is larger than 5 GB, use resumable upload. For more information, see Multipart upload and resumable upload.

- Naming conventions
  - The name must be encoded in UTF-8.
  - The name must be 1 to 1,023 characters in length.
  - The name cannot start with a forward slash (/) or a backslash (\).
- Performance tuning of object upload

  If you upload a large number of objects with sequential prefixes such as timestamps and letters in the object names, many object indexes may be stored in a single partition. In this case, if you send a large number of requests to query these objects, the latency may increase. We recommend that you do not upload a large number of objects with sequential prefixes. For more information about how to change sequential prefixes to random prefixes, see OSS performance and scalability best practices.
- Preventing an object that has the same name from being overwritten

  By default, when you upload an object to OSS, the existing object that has the same name is overwritten. You can use the following methods to prevent your objects from being unexpectedly overwritten:
  - Enable versioning

    If versioning is enabled, overwritten objects are saved as previous versions. You can restore the previous versions at any time. For more information, see Overview.
  - Add a specific parameter in the upload request

    Include the x-oss-forbid-overwrite parameter in the upload request header and set the value to *true*. This way, if you upload an object whose name is the same as an existing object, the upload fails and OSS returns the `FileAlreadyExists` error. If this parameter is not included in the request header or the value of this parameter is set to *false*, the object that has the same name is overwritten. For more information, see PutObject.

Process analysis

1. Create a POST policy.

   The policy form field of a POST request is used to verify the validity of the request. For example, you can configure a policy to specify the size and name of the object that you want to upload, as well as the URL to which the client is redirected and the HTTP status code that the client receives after a successful upload. For more information, see PostObject.

   The following example uses the Python code. The policy is a JSON string.

   ```
   # Set the expiration time before which website users can upload data to 2115-01-27T10:56:19Z and the maximum object size to 104,857,600 bytes. To ensure successful testing, a long expiration period is specified, which is not recommended in actual scenarios.
   policy="{\"expiration\":\"2115-01-27T10:56:19Z\",\"conditions\":[[\"content-length-range\", 0, 104857600]]}"
   ```
2. Encode the policy string in Base64.
3. Add a signature to the Base64-encoded policy by using the AccessKey secret of the account that is used to access OSS.
4. Create an HTML page for upload.
5. Open the HTML page and select the object that you want to upload.

The following example shows the complete sample Python code:

```
#coding=utf8
import md5
import hashlib
import base64
import hmac
from optparse import OptionParser
def convert_base64(input):
    return base64.b64encode(input)
def get_sign_policy(key, policy):
    return base64.b64encode(hmac.new(key, policy, hashlib.sha1).digest())
def get_form(bucket, endpoint, access_key_id, access_key_secret, out):
    # 1 Create a POST policy.
    policy="{\"expiration\":\"2115-01-27T10:56:19Z\",\"conditions\":[[\"content-length-range\", 0, 1048576]]}"
    print("policy: %s" % policy)
    # 2 Encode the policy string in Base64.
    base64policy = convert_base64(policy)
    print("base64_encode_policy: %s" % base64policy)
    # 3 Add a signature to the Base64-encoded policy by using the AccessKey secret of the account that is used to access OSS.
    signature = get_sign_policy(access_key_secret, base64policy)
    # 4 Create an HTML page for the upload.
    form = '''
    <html>
        <meta http-equiv=content-type content="text/html; charset=UTF-8">
        <head><title>OSS form upload (by calling the PostObject operation)</title></head>
        <body>
            <form  action="http://%s.%s" method="post" enctype="multipart/form-data">
                <input type="text" name="OSSAccessKeyId" value="%s">
                <input type="text" name="policy" value="%s">
                <input type="text" name="Signature" value="%s">
                <input type="text" name="key" value="upload/${filename}">
                <input type="text" name="success_action_redirect" value="http://oss.aliyun.com">
                <input type="text" name="success_action_status" value="201">
                <input name="file" type="file" id="file">
                <input name="submit" value="Upload" type="submit">
            </form>
        </body>
    </html>
    ''' % (bucket, endpoint, access_key_id, base64policy, signature)
    f = open(out, "wb")
    f.write(form)
    f.close()
    print("form is saved into %s" % out)
if __name__ == '__main__':
    parser = OptionParser()
    parser.add_option("", "--bucket", dest="bucket", help="specify ")
    parser.add_option("", "--endpoint", dest="endpoint", help="specify")
    parser.add_option("", "--id", dest="id", help="access_key_id")
    parser.add_option("", "--key", dest="key", help="access_key_secret")
    parser.add_option("", "--out", dest="out", help="out put form")
    (opts, args) = parser.parse_args()
    if opts.bucket and opts.endpoint and opts.id and opts.key and opts.out:
        get_form(opts.bucket, opts.endpoint, opts.id, opts.key, opts.out)
    else:
        print "python %s --bucket=your-bucket --endpoint=oss-cn-hangzhou.aliyuncs.com --id=your-access-key-id --key=your-access-key-secret --out=out-put-form-name" % __file__
```

Save the preceding sample code as *post_object.py* and run the code by using `python post_object.py`.

Run the saved file in the following way:

```
python post_object.py --bucket=Your bucket name --endpoint=Bucket endpoint --id=Your AccessKey ID --key=Your AccessKey secret --out=Output file name
```

Example:

```
python post_object.py --bucket=oss-sample --endpoint=oss-cn-hangzhou.aliyuncs.com --id=tphpxp --key=ZQNJzf4QJRkrH4 --out=post.html
```

Note

- In the created form, `success_action_redirect value=http://oss.aliyun.com` specifies the web page that appears if the upload is successful. You can replace this with your web page.
- In the created form, `success_action_status value=201` specifies that HTTP status code 201 is returned if the upload is successful. You can change this to another HTTP status code.
- If the generated HTML page is *post.html*, open *post.html* and select the object that you want to upload. In this example, if the object is successfully uploaded, you are redirected to the OSS product page.

Security and authorization

To prevent unauthorized third-party users from uploading data to your bucket, OSS provides bucket-level and object-level access control. For more information, see Overview.

To authorize third-party users to upload objects to your buckets, OSS also provides account-level authorization. For more information, see Authorized third-party upload.

6.3.6. Append upload

You can use append upload to append the content of an object to an existing append object.

Prerequisites

A bucket is created. For more information, see Create buckets.

Background information

Objects that are uploaded by using simple upload are considered normal objects. Objects that are uploaded by using multipart upload are considered multipart objects. For more information, see Simple upload and Multipart upload. The content of normal and multipart objects can be only read and not modified after the objects are uploaded. To change the content of an existing normal or multipart object, you must upload an object with the same name as the existing object to overwrite the existing object.

If you use simple upload or multipart upload to upload real-time video streams generated by surveillance and live streaming services, you must split the streams into parts based on specific rules and continuously upload new parts to OSS as separated objects. This upload method has the following disadvantages:

- You must develop a complex software to handle the details during upload, such as splitting the streams into parts.
- You must reserve storage spaces to store object metadata, such as the list of uploaded objects. The client must repeatedly query the metadata to determine whether a new object is uploaded. In this case, the client must send two requests to check whether new objects are uploaded, which increases network latency and the workload of the OSS server.
- If the streams are split into small objects, the network latency decreases, but object management becomes more complex. If the streams are split into large objects, the network latency significantly increases.

To update the content of uploaded objects in real time in the preceding scenarios, OSS allows you to use append upload to append data to existing append objects. Objects uploaded by using append upload are considered append objects. You can append content to existing append objects. Data that is appended to an append object can be immediately read.

Benefits

You can use append upload to upload the data generated by a video stream to a single object. The client needs only to periodically compare the current length of the object with the object length obtained last time to determine whether new data is uploaded. If the client determines that new data is appended, the client sends a request to obtain the appended data. This way, you can simplify your software and improve scalability.

Limits

- Limits on object size

  The object that you can upload by using append upload cannot be larger than 5 GB in size.

- Limits on object names
  - The name must be encoded in UTF-8.
  - The name must be 1 to 1,023 characters in length.
  - The name cannot start with a forward slash (/) or a backslash (\).

- Limits on operations
  - Append objects cannot be copied. However, you can modify the metadata of append objects.
  - Upload callback is not supported in append upload.

Use OSS SDKs

The following code provides examples on how to perform append upload operation by using OSS SDKs for common programming languages. For more information about how to perform append upload operation by using OSS SDKs for other programming languages, see Overview.

Python

```
// Set yourEndpoint to the endpoint of the region in which the bucket is located. For example, if the bucket is located in the China (Hangzhou) region, set yourEndpoint to https://oss-cn-hangzhou.aliyuncs.com.
String endpoint = "yourEndpoint";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access Object Storage Service (OSS) because the account has permissions on all API operations. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine Operation and Maintenance (O&M). To create a RAM user, log on to the RAM console.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Specify the bucket name. Example: examplebucket.
String bucketName = "examplebucket";
// Specify the full path of the object. The full path cannot contain the bucket name. Example: exampledir/exampleobject.txt.
String objectName = "exampledir/exampleobject.txt";
String content1 = "Hello OSS A \n";
String content2 = "Hello OSS B \n";
String content3 = "Hello OSS C \n";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
ObjectMetadata meta = new ObjectMetadata();
// Specify the type of content that you want to upload.
meta.setContentType("text/plain");
// Specify the caching behavior of the web page for the object.
//meta.setCacheControl("no-cache");
// Specify the name of the object when the object is downloaded.
//meta.setContentDisposition("attachment;filename=oss_download.txt");
// Specify the encoding format for the content of the object.
//meta.setContentEncoding(OSSConstants.DEFAULT_CHARSET_NAME);
// Specify the request header that is used to check whether the content of the received message is the same as the content of the sent message.
//meta.setContentMD5("ohhnqLBJFiKkPSBO1eNaUA==");
// Specify the expiration time.
//try {
//    meta.setExpirationTime(DateUtil.parseRfc822Date("Wed, 08 Jul 2022 16:57:01 GMT"));
//} catch (ParseException e) {
//    e.printStackTrace();
//}
// Specify the server-side encryption method. In this example, the method is set to server-side encryption by using OSS-managed keys (SSE-OSS).
//meta.setServerSideEncryption(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);
// Specify the access control list (ACL) of the object. In this example, the ACL of the object is set to private.
//meta.setObjectAcl(CannedAccessControlList.Private);
// Specify the storage class of the object.
//meta.setHeader(OSSHeaders.OSS_STORAGE_CLASS, StorageClass.Standard);
// You can add parameters whose names are prefixed with x-oss-meta- when you call the AppendObject operation to create an append object. These parameters cannot be included in the requests when you append content to an existing append object. Parameters whose names are prefixed with x-oss-meta- are considered as the metadata of the object.
//meta.setHeader("x-oss-meta-author", "Alice");
// Configure multiple parameters by using AppendObjectRequest.
AppendObjectRequest appendObjectRequest = new AppendObjectRequest(bucketName, objectName, new ByteArrayInputStream(content1.getBytes()),meta);
// Configure a single parameter by using AppendObjectRequest.
// Specify the bucket name.
//appendObjectRequest.setBucketName(bucketName);
// Specify the name of the object.
//appendObjectRequest.setKey(objectName);
// Specify the content that you want to append. Two types of content are available: InputStream and File. In this example, the content is set to the InputStream type.
//appendObjectRequest.setInputStream(new ByteArrayInputStream(content1.getBytes()));
// Specify the content that you want to append. Two types of content are available: InputStream and File. In this example, the content is set to the File type,
//appendObjectRequest.setFile(new File("D:\\localpath\\examplefile.txt"));
// Specify the metadata of the object. You can specify the metadata of an object only when you perform the first append operation on the object.
//appendObjectRequest.setMetadata(meta);
// Perform the first append operation.
// Specify the position from which the append operation starts.
appendObjectRequest.setPosition(0L);
AppendObjectResult appendObjectResult = ossClient.appendObject(appendObjectRequest);
// Calculate the 64-bit CRC value of the object. The value is calculated based on the ECMA-182 standard.
System.out.println(appendObjectResult.getObjectCRC());
// Perform the second append operation.
// NextPosition specifies the position from which the next append operation starts, which is the current length of the object.
appendObjectRequest.setPosition(appendObjectResult.getNextPosition());
appendObjectRequest.setInputStream(new ByteArrayInputStream(content2.getBytes()));
appendObjectResult = ossClient.appendObject(appendObjectRequest);
// Perform the third append operation.
appendObjectRequest.setPosition(appendObjectResult.getNextPosition());
appendObjectRequest.setInputStream(new ByteArrayInputStream(content3.getBytes()));
appendObjectResult = ossClient.appendObject(appendObjectRequest);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Use ossutil

For more information about how to perform append upload operation by using ossutil, see appendfromfile.

Use RESTful API operations

If your program requires more custom options to perform append upload operation, you can call RESTful API operations. In this case, you must manually write code to calculate the signature. For more information, see AppendObject.

6.3.7. RTMP-based stream ingest

Object Storage Service (OSS) allows you to use Real-Time Messaging Protocol (RTMP) to ingest H.264-encoded video streams and Advanced Audio Coding (AAC)-encoded audio streams to OSS. Audio and video data uploaded to OSS can be played on demand or be used for live streaming in latency-insensitive scenarios.

When you upload audio and video data to OSS in compliance with RTMP, take note of the following limits:

- You can only ingest video or audio streams but not pull the streams if you use RTMP.
- The uploaded audio and video data must include H.264 video streams.
- You can select whether to include audio streams in the audio and video data. If you want to include audio streams, only AAC audio streams are supported. Auto streams in other formats are discarded.
- You can use only HTTP Live Streaming (HLS) to store audio and video data in OSS.
- A LiveChannel can receive streams ingested from only one client at a time.

The following sections describe how to ingest audio and video streams to OSS and how to play the uploaded audio and video data on demand and for live streaming.

Ingest audio and video streams to OSS

- Obtain an ingest URL

  Use an SDK to call the PutLiveChannel operation, create a LiveChannel, and then obtain the corresponding ingest URL. If the bucket ACL is set to public read/write, you can directly use the obtained ingest URL. Otherwise, add a signature to the ingest URL.

  The following code uses the Python SDK in the example to show how to obtain an ingest URL without a signature and how to obtain a signed ingest URL:

  ```
  from oss2 import *
  from oss2.models import *
  host = "oss-cn-hangzhou.aliyuncs.com" #just for example
  accessid = "your-access-id"
  accesskey = "your-access-key"
  bucket_name = "your-bucket"
  channel_name = "test-channel"
  auth = Auth(accessid, accesskey)
  bucket = Bucket(auth, host, bucket_name)
  channel_cfg = LiveChannelInfo(target = LiveChannelInfoTarget())
  channel = bucket.create_live_channel(channel_name, channel_cfg)
  publish_url = channel.publish_url
  signed_publish_url = bucket.sign_rtmp_url("test-channel", "playlist.m3u8", 3600)
  ```

  The following example shows the obtained ingest URLs:

  ```
  publish_url = rtmp://your-bucket.oss-cn-hangzhou.aliyuncs.com/live/test-channel
  signed_publish_url = rtmp://your-bucket.oss-cn-hangzhou.aliyuncs.com/live/your-channel?OSSAccessKeyId=LGarxxxxxxHjKWg6&playlistName=t.m3u8&Expires=
  1472201595&Signature=bjKraZTTyzz9%2FpYoomDx4Wgh%2FlM%3D"
  ```

- Use FFmpeg for stream ingest

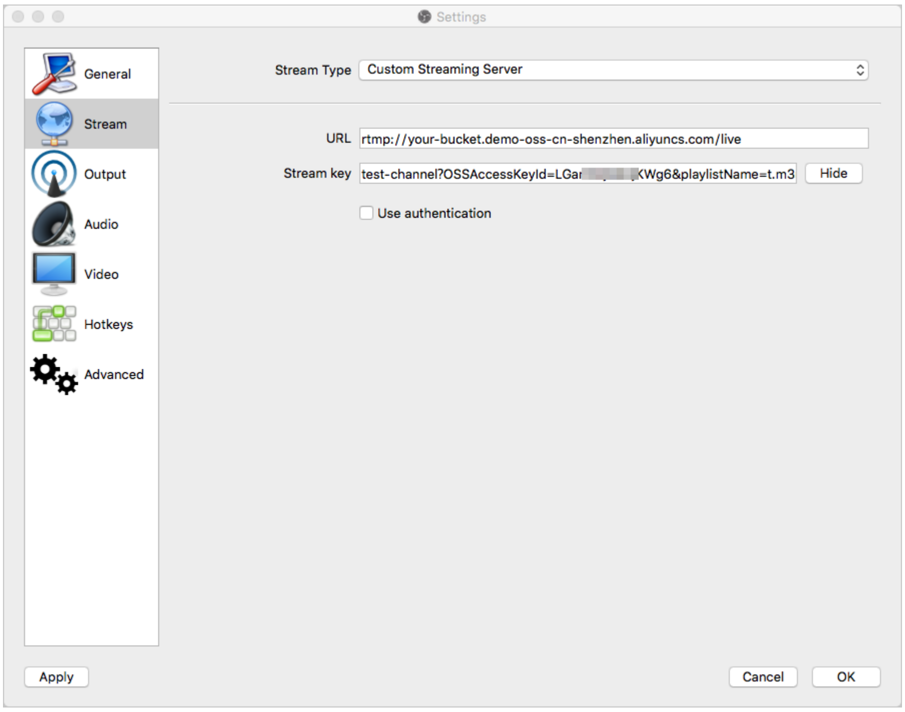
  You can use FFmpeg to upload local video objects to OSS by running the following command:

  ```
  ffmpeg -i 1.flv -c copy -f flv "rtmp://your-bucket.oss-cn-hangzhou.aliyuncs.com/live/test-channel?OSSAccessKeyId=LGarxxxxxxHjKWg6&Expires=147219909
  5&Signature=%2FAvRo7FTss1InBKgwn7Gz%2FUlp9w%3D"
  ```

- Use OBS for stream ingest

  Click **Settings**. In the URL field, enter the ingest URL that you obtain in the preceding step, and then click **OK**.

  Take note of how the ingest URL is split based on the following figure.



Stream Type: Custom Streaming Server

URL: rtmp://your-bucket.demo-oss-cn-shenzhen.aliyuncs.com/live

Stream key: test-channel?OSSAccessKeyId=LGa[redacted]KWg6&playlistName=t.m3

☐ Use authentication

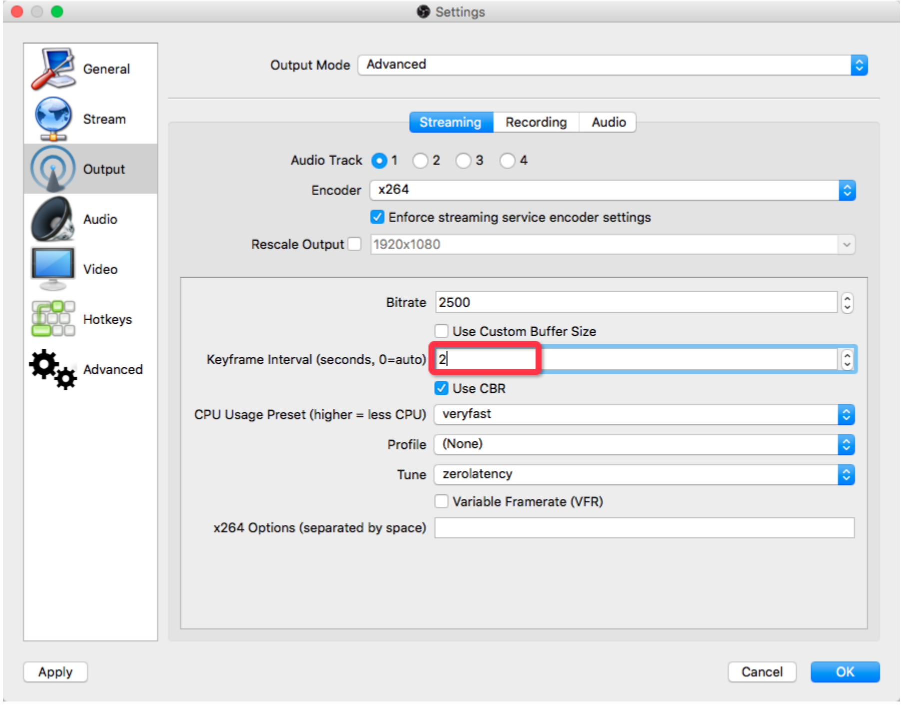
Play the audio and video data uploaded to OSS

- Live streaming

  During stream ingest, you can use HLS to play the audio and video content that is being uploaded on the following platforms in different ways:

  - On mobile platforms such as Android and iOS, enter the corresponding streaming URL of a LiveChannel in the browser.
  - On the macOS platform, use Safari to play the content.
  - On a PC, install the VLC media player to play the content.

  To smoothly play the uploaded audio and video data for live streaming, you can set FragDuration to a small value such as 2s. You can also set the group of pictures (GOP) to a fixed value, which is the same as that of FragDuration of the LiveChannel. The following figure shows how to set the GOP (Keyframe Interval) in OBS.



Output Mode: Advanced

Audio Track: ● 1 ○ 2 ○ 3 ○ 4

Encoder: x264

☑ Enforce streaming service encoder settings

Rescale Output ☐ 1920x1080

Bitrate: 2500

☐ Use Custom Buffer Size

Keyframe Interval (seconds, 0=auto): 2

☑ Use CBR

CPU Usage Preset (higher = less CPU): veryfast

Profile: (None)

Tune: zerolatency

☐ Variable Framerate (VFR)

x264 Options (separated by space):

- Playback on demand

  When you ingest a stream, OSS uses live streaming to push or update M3U8 objects. In on-demand playback scenarios, you must call the PostVodPlaylist operation after stream ingest to assemble an M3U8 object for on-demand playback and use the object URL to play the uploaded audio and video data.

  In on-demand playback scenarios, you can set a larger GOP to reduce the number of TS objects and the bit rate.

6.4. Download files

6.4.1. Simple download

When you download an object by using simple download from Object Storage Service (OSS), the GetObject operation is called to download the object. Simple download is applicable to downloads that can be completed with an HTTP request.

Prerequisites

When you download Archive or Cold Archive objects, make sure that the objects are restored. For more information, see Restore objects.

Use the OSS console

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the page that appears, click the name of the bucket from which you want to download objects.
3. In the left-side navigation pane, click **Files**, and then download one or more objects.
   - Download a single object

     Method 1: Choose **More > Download** in the Actions column that corresponds to the object that you want to download.

     Method 2: Click the name of the object that you want to download, or click **View Details** in the Actions column that corresponds to the object you want to download. In the **View Details** panel, click **Download**.
   - Download multiple objects

     Select the objects that you want to download. Then, choose **Batch Operation > Download**. You can batch download up to 100 objects in the OSS console.

   For more information about how to download objects from versioned buckets, see Configure versioning.

Use ossbrowser

ossbrowser supports the same operations related to objects as the OSS console. You can follow the on-screen instructions in ossbrowser to perform simple download. For more information about how to use ossbrowser, see Use ossbrowser.

Use OSS SDKs

The following code provides examples on how to perform simple download by using OSS SDKs for common programming languages. For more information about how to perform simple download by using OSS SDKs for other programming languages, see Overview.

Python

```
// Set yourEndpoint to the endpoint of the region where the bucket is located. For example, if your bucket is in the China (Hangzhou) region, set you
rEndpoint to https://oss-cn-hangzhou.aliyuncs.com.
String endpoint = "yourEndpoint";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to log on to OSS because the account has permissions on all API
operations. We recommend that you use your RAM user's credentials to call API operations or perform routine operations and maintenance. To create a R
AM user, log on to the RAM console.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Specify the bucket name.
String bucketName = "examplebucket";
// Specify the complete path of the object excluding the bucket name. Example: testfolder/exampleobject.txt.
String objectName = "testfolder/exampleobject.txt";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Download the object to a file, and save it to a specified path. If the specified file exists, the downloaded object replaces the file. Otherwise,
the file is created.
// By default, if the path is not specified, the downloaded file is saved to the path of the project to which the sample program belongs.
ossClient.getObject(new GetObjectRequest(bucketName, objectName), new File("D:\\localpath\\examplefile.txt"));
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Use ossutil

For more information about how to use ossutil to perform simple download, see Download objects.

Use the RESTful API

If your program requires more custom options to perform simple download, you can call RESTful API operations. In this case, you need to manually write code to calculate the signature. For more information, see GetObject.

References

- To prevent unauthorized third-party users from downloading data from your bucket, OSS provides bucket- and object-level access control. For more information, see Overview.
- If you want to grant third-party users the permission to download objects from your bucket whose access control list (ACL) is private, use Security Token Service (STS) to generate a temporary access token or use a signed URL. For more information, see Authorize third-party users to download objects.
- If you want interrupted downloads to continue to download from where they are interrupted when you download large objects, you can use resumable download. For more information, see Resumable download.

6.4.2. Resumable download

OSS allows you to download data from a specified position of an object. When you download a large object, you can split it into multiple parts and download them at different points of time. If a download is paused or interrupted, you can also resume it at the paused or interrupted position.

Similar to simple upload, you must have the read permissions on the object. You can set the Range parameter to enable resumable download. We recommend that you use this feature to download large objects. For more information about the Range parameter, visit HTTP RFC. If the Range parameter is specified in the request header, the response contains the length of the entire object and the range returned in this response. For example, "Content-Range: bytes 0-9/44" indicates that the total size of the object is 44 bytes, and the range of data returned is the first 10 bytes. If the specified Range parameter value is invalid, the entire object is transmitted. The response excludes Content-Range. However, HTTP status code 206 is returned.

Implementation modes

Implementation mode	Description
Java SDK	SDK demos for various programming languages
Python SDK	
Go SDK	
C SDK	
iOS SDK	

Download security and authorization

- To prevent unauthorized third-party users from downloading data from your bucket, OSS provides bucket- and object-level access control. For more information, see Overview.
- For more information about how to authorize a third-party user to download objects from a private bucket, see Authorized third-party users to download objects.

6.4.3. Authorize third-party users to download objects

To authorize a third-party user to download objects from a private bucket, you can provide a signed URL or a temporary access credential instead of your AccessKey pair to the user.

Add a signature to a URL

Object Storage Service (OSS) allows users to use a signed URL to download data. You can add signature information to a URL and provide the URL to a third-party user for authorized access. The third-party user can access the URL by sending a GET request to download objects.

- Example

  ```
  http://<bucket>.<region>.aliyuncs.com/<object>?OSSAccessKeyId=<user access_key_id>&Expires=<unix time>&Signature=<signature_string>
  ```

  The URL must be URL-encoded. To add a signature to the URL, you must include at least the following parameters:

  - OSSAccessKeyId: The AccessKey ID of your Alibaba Cloud account.
  - Expires: The expected expiration time of the URL.
  - Signature: The signature string. For more information, see Add signatures to a URL.

- Implementation methods
  - Console
  - Java SDK
  - Python SDK
  - PHP SDK
  - Go SDK
  - C SDK
  - C++ SDK
  - .NET SDK
  - Node.js
  - Browser.js
  - Android SDK
  - iOS SDK

Temporary access credential

You can use Alibaba Cloud Security Token Service (STS) to authorize temporary access to OSS. You can use STS to grant a third-party application or your Resource Access Management (RAM) user an access credential that specifies the custom validity period and permissions. This authorization method is applicable to scenarios in which objects are downloaded from mobile devices. For more information, see Use a temporary credential provided by STS to access OSS.

- Process

  A third-party user sends a request to the application server to obtain the AccessKey ID, AccessKey secret, and STS token. The user then uses the access credential to request objects of developers.

- Implementation methods
  - Java SDK
  - Python SDK
  - PHP SDK
  - Go SDK
  - C SDK
  - C++ SDK
  - .NET SDK
  - Node.js
  - Browser.js

- Android SDK
- iOS SDK

6.5. Manage files

6.5.1. List objects

By default, when you list objects in a bucket, the objects are returned in alphabetical order. You can list all objects in a bucket, objects with a specified prefix in their names, or a specified number of objects.

Use the OSS console

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**.
3. On the **Buckets** page, find the bucket that you want to manage and click the bucket name.
   All objects in the bucket are displayed by page. By default, 50 objects are displayed per page.

Use ossbrowser

ossbrowser supports the same operations related to objects as the Object Storage Service (OSS) console. You can follow the on-screen instructions in ossbrowser to list objects. For more information about how to use ossbrowser, see Use ossbrowser.

Use OSS SDKs

The following code provide examples on how to use OSS SDKs for common programming languages to list all objects in a bucket. For more information about how to use OSS SDKs for other programming languages to list objects that meet specified conditions in different scenarios, see Overview.

[illegible]

```
// Set yourEndpoint to the endpoint of the region in which the bucket is located. For example, if your bucket is located in the China (Hangzhou) regi
on, set yourEndpoint to https://oss-cn-hangzhou.aliyuncs.com.
String endpoint = "yourEndpoint";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API op
erations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Specify the name of the bucket. Example: examplebucket.
String bucketName = "examplebucket";
// Specify the prefix that is contained in the names of the objects. Example: exampledir/object.
String keyPrefix = "exampledir/object";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// List the objects in the bucket. If you do not specify keyPrefix, all objects in the bucket are listed. If you set keyPrefix, objects whose names c
ontain the specified prefix in the bucket are listed.
ListObjectsV2Result result = ossClient.listObjectsV2(bucketName, keyPrefix);
List<OSSObjectSummary> ossObjectSummaries = result.getObjectSummaries();
for (OSSObjectSummary s : ossObjectSummaries) {
    System.out.println("\t" + s.getKey());
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Use ossutil

For more information about how to use ossutil to list objects, see List objects.

Use the RESTful API

If your program requires more custom options to list objects, you can call RESTful API operations. In this case, you need to manually write code to calculate the signature.

You can call GetBucket (ListObjects) or GetBucketV2 (ListObjectsV2) to list objects in a bucket. We recommend that you use GetBucketV2 (ListObjectsV2) when you develop your applications. To provide backward compatibility, OSS continues to support the GetBucket (ListObjects) operation.

6.5.2. Copy objects

You can copy an object from the source bucket to the destination bucket within the same region without modifying the content of the object.

Limits

- Objects cannot be copied between buckets located in different regions. For example, an object cannot be copied from a bucket in the China (Hangzhou) region to a bucket in the China (Shanghai) region.
- Objects uploaded by calling the AppendObject operation cannot be copied.

Usage notes

- You must have the read permissions on the source object and the read and write permissions on the destination bucket. Otherwise, the copy operation may fail.
- The source bucket and the destination bucket must have no retention policies configured. Otherwise, the error message `The object you specified is immutable.` is returned.
- By default, when you copy an object, an existing object that has the same name is overwritten. You can use the following methods to protect your objects from being unexpectedly overwritten:

- Enable versioning

  If versioning is enabled for a bucket, deleted or overwritten objects in the bucket are saved as previous versions. You can restore an object to a previous version at any time. For more information, see Overview.

- Include a parameter in the copy request header

  Include the x-oss-forbid-overwrite parameter in the copy request header and set the parameter to *true*. This way, when you copy an object that has the same name as an existing object in the destination bucket, the object cannot be copied. Object Storage Service (OSS) returns the `FileAlreadyExists` error.

Use ossbrowser

If you use ossbrowser to copy objects, the objects must be smaller than 5 GB in size. For more information about how to use ossbrowser to copy objects, see Use ossbrowser.

Use OSS SDKs

The following code provides examples on how to call the CopyObject operation to copy an object smaller than 1 GB by using OSS SDKs for common programming languages: For the sample code that uses OSS SDKs for other programming languages to copy objects smaller than 1 GB, and the sample code used to call the UploadPartCopy operation to upload objects larger than 1 GB, see Overview.

[illegible]

```
// Set yourEndpoint to the endpoint of the region in which the bucket is located. For example, if the bucket is located in the China (Hangzhou) regio
n, set the endpoint to https://oss-cn-hangzhou.aliyuncs.com.
String endpoint = "yourEndpoint";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API op
erations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Specify the name of the source bucket.
String sourceBucketName = "srcexamplebucket";
// Specify the full path of the source object. The full path of the object cannot contain the bucket name.
String sourceKey = "srcexampleobject.txt";
// Specify the name of the destination bucket. The destination bucket must be in the same region as the source bucket.
String destinationBucketName = "desexamplebucket";
// Specify the full path of the destination object. The full path of the object cannot contain the bucket name.
String destinationKey = "desexampleobject.txt";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Create a CopyObjectRequest object.
CopyObjectRequest copyObjectRequest = new CopyObjectRequest(sourceBucketName, sourceKey, destinationBucketName, destinationKey);
// Configure the metadata of the destination object.
ObjectMetadata meta = new ObjectMetadata();
meta.setContentType("text/txt");
// Specify whether the CopyObject operation overwrites the object with the same name. In this example, this parameter is set to true, which indicates
that the object with the same name cannot be overwritten.
// meta.setHeader("x-oss-forbid-overwrite", "true");
// Specify the path of the source object.
// meta.setHeader(OSSHeaders.COPY_OBJECT_SOURCE, "/examplebucket/recode-test.txt");
// If the ETag value of the source object is the same as the ETag value that is specified in the request, OSS copies the object and returns 200 OK.
// meta.setHeader(OSSHeaders.COPY_OBJECT_SOURCE_IF_MATCH, "5B3C1A2E053D763E1B002CC607C5****");
// If the ETag value of the source object is different from the ETag value that is specified in the request, OSS copies the object and returns 200 OK
.
// meta.setHeader(OSSHeaders.COPY_OBJECT_SOURCE_IF_NONE_MATCH, "5B3C1A2E053D763E1B002CC607C5****");
// If the time that is specified in the request is later than or equal to the actual modified time of the source object, OSS copies the object and re
turns 200 OK.
// meta.setHeader(OSSHeaders.COPY_OBJECT_SOURCE_IF_UNMODIFIED_SINCE, "2021-12-09T07:01:56.000Z");
// If the source object is modified after the time that is specified in the request, OSS copies the object.
// meta.setHeader(OSSHeaders.COPY_OBJECT_SOURCE_IF_MODIFIED_SINCE, "2021-12-09T07:01:56.000Z");
// Specify the method that is used to configure the metadata of the destination object. In this example, the method is set to COPY, which indicates t
hat the metadata of the source object is copied to the destination object.
// meta.setHeader(OSSHeaders.COPY_OBJECT_METADATA_DIRECTIVE, "COPY");
// Specify the server-side encryption algorithm that is used to encrypt the destination object when OSS creates the object.
// meta.setHeader(OSSHeaders.OSS_SERVER_SIDE_ENCRYPTION, ObjectMetadata.KMS_SERVER_SIDE_ENCRYPTION);
// Specify the customer master key (CMK) that is managed by Key Management Service (KMS). This parameter takes effect only when you set x-oss-server-
side-encryption to KMS.
// meta.setHeader(OSSHeaders.OSS_SERVER_SIDE_ENCRYPTION_KEY_ID, "9468da86-3509-4f8d-a61e-6eab1eac****");
// Specify the access control list (ACL) of the destination object. In this example, the ACL is set to private, which indicates that only the owner o
f the object and authorized users have read and write permissions on the object.
// meta.setHeader(OSSHeaders.OSS_OBJECT_ACL, CannedAccessControlList.Private);
// Specify the storage class of the destination object. In this example, the storage class is set to Standard.
// meta.setHeader(OSSHeaders.OSS_STORAGE_CLASS, StorageClass.Standard);
// Specify tags for the destination object. You can specify multiple tags for the destination object at the same time.
// meta.setHeader(OSSHeaders.OSS_TAGGING, "a:1");
// Specify the method that is used to configure tags for the destination object. In this example, the method is set to COPY, which indicates that the
tags of the source object are copied to the destination object.
// meta.setHeader(OSSHeaders.COPY_OBJECT_TAGGING_DIRECTIVE, "COPY");
copyObjectRequest.setNewObjectMetadata(meta);
// Copy the object.
CopyObjectResult result = ossClient.copyObject(copyObjectRequest);
System.out.println("ETag: " + result.getETag() + " LastModified: " + result.getLastModified());
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Use ossutil

For more information about how to copy objects by using ossutil, see Copy objects.

Use the RESTful API

If your program requires more custom options to copy objects, you can call RESTful API operations. In this case, you must manually write code to calculate the signature. For more information, see CopyObject.

6.5.3. Restore objects

You must restore an Archive or a Cold Archive object before you can read it. This topic describes how to restore an Archive or a Cold Archive object.

Restoration

If you want to read an Archive object or a Cold Archive object, you must restore the object in advance. It takes several minutes to restore an Archive object and takes several hours to restore a Cold Archive object.

The following section describes the status of an Archive object or a Cold Archive object throughout the restoration process:

1. By default, an Archive or a Cold Archive object is in the frozen state before restoration.
2. After you submit a restore request, the object is in the restoring state.
3. After the server completes the restore task, the object enters the restored state and you can read the object.
    - Archive objects

      For Archive objects, the restored state lasts 24 hours by default. During the 24 hours, if you call RestoreObject again, the restored state is extended by 24 hours. You can extend the restored state to seven days by calling RestoreObject six times during the first 24 hours of restored state. You can also configure the duration of the restored state in days by calling RestoreObject once. You can specify a duration of at most seven days.

    - Cold Archive objects

      For Cold Archive objects, you can specify the duration of restored state and restoration priority. The duration of the restored state must be at least one day and at most 365 days. The time required to restore a Cold Archive object to the readable state is determined based on the restoration priority of the object:

        - Expedited: The object is restored within one hour.
        - Standard: The object is restored within two to five hours. If the JobParameters element is not passed in, the default restoration priority is Standard.
        - Bulk: The object is restored within five to twelve hours.
4. After the restored state expires, the object returns to the frozen state.

Precautions

- The RestoreObject operation applies only to archived objects or cold archived objects. This operation does not apply to Standard or IA objects.
- The first time RestoreObject is called on an object, HTTP status code 202 is returned.
- The first time RestoreObject is called on an object, HTTP status code 202 is returned. When you call RestoreObject that was called on a frozen object whose restored state expires, 200 OK is returned.
- In a versioned bucket, the storage classes of different versions of an object can be different. By default, when you call RestoreObject to restore an object, the current version of the object is restored. You can specify a version ID in the request to restore the specified version of the object.

Billing

- Data retrieval fees are generated when you restore Archive and Cold Archive objects. For more information, see Data processing fees.
- The restored state of an Archive object can persist up to seven days and that of a Cold Archive object can persist up to 365 days. You are not repeatedly charged data retrieval fees during this process.
- After the restored state expires, the object returns to the frozen state. Data retrieval fees are generated if you perform the restore operation on the object again.
- When you restore a Cold Archive object, a Standard replica is generated for temporary access. OSS charges the temporary storage fees of the replica for the duration during which the replica is available based on Standard storage. For more information, see Temporary storage fees.

Use the OSS console

1.

Use ossbrowser

ossbrowser allows you to perform bucket management operations that you can perform in the OSS console. You can follow the on-screen instructions in ossbrowser to restore objects. For more information about how to use ossbrowser, see Use ossbrowser.

Use OSS SDKs

The following code provides examples on how to restore objects by using OSS SDKs for common programming languages. For more information about how to restore objects by using OSS SDKs for other programming languages, see Overview.

Python

```
// Specify the endpoint of the region in which the bucket is located. For example, if the bucket is located in the China (Hangzhou) region, set endpoint to https://oss-cn-hangzhou.aliyuncs.com.
String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API operations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Specify the name of the bucket.
String bucketName = "yourBucketName";
// Specify the full path of the Archive object. The path cannot contain the bucket name.
String objectName = "yourArchiveObjectName";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
ObjectMetadata objectMetadata = ossClient.getObjectMetadata(bucketName, objectName);
// Check whether the object is an Archive object.
StorageClass storageClass = objectMetadata.getObjectStorageClass();
if (storageClass == StorageClass.Archive) {
    // Restores the object.
    ossClient.restoreObject(bucketName, objectName);
    // Wait until the object is restored.
    do {
        Thread.sleep(1000);
        objectMetadata = ossClient.getObjectMetadata(bucketName, objectName);
    } while (!objectMetadata.isRestoreCompleted());
}
// Obtain the restored object.
OSSObject ossObject = ossClient.getObject(bucketName, objectName);
ossObject.getObjectContent().close();
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Use ossutil

For more information about how to restore Archive and Cold Archive objects by using ossutil, see restore (restore objects).

Use the RESTful API

If your program requires more custom options, you can call RESTful API operations. In this case, you must write code to calculate the signature. For more information, see RestoreObject.

6.5.4. Delete objects

You can use multiple methods to delete objects that you no longer need from your buckets in Object Storage Service (OSS).

Warning You cannot recover deleted objects. Exercise caution when you perform this operation.

Deletion rules

OSS allows you to automatically or manually delete a single object or multiple objects based on the following rules:

- Delete a single object: You can delete a specific object.
- Delete multiple objects: You can delete up to 1,000 objects at a time.
- Delete objects automatically: This is applicable to scenarios where you want to delete multiple objects with specific characteristics. For example, objects that are last modified before a specific date, have a specific prefix in their names, or are stored in a specific bucket. In this case, we recommend that you configure a lifecycle rule to delete the objects. After you configure the lifecycle rule, OSS automatically deletes specified objects based on the rule. This automates the delete procedure and improves efficiency. For more information, see Overview.

Implementation methods

Implementation method	Description
Console	A user-friendly and intuitive web application
ossbrowser	An easy-to-operate graphical tool
ossutil	A high-performance command-line tool
Java SDK	SDK demos for various programming languages
Python SDK	
PHP SDK	
Go SDK	
C SDK	
.NET SDK	
Android SDK	
iOS SDK	

Implementation method	Description
Node.js SDK	
Browser.js SDK	
Ruby SDK	

6.5.5. Object tagging

Object Storage Service (OSS) allows you to configure object tags to classify objects. You can configure lifecycle rules and control access to objects based on tags.

> Note You are charged when you use object tagging. For more information, see Object Storage Service Pricing.

Object tagging uses a key-value pair to identify objects. You can add tags to objects when and after you upload the objects.

- A maximum of 10 tags can be configured for each object. Tags associated with an object must have unique tag keys.
- A tag key can be up to 128 characters in length. A tag value can be up to 256 characters in length.
- Tag keys and tag values are case-sensitive.
- The key and value of a tag can contain letters, digits, spaces, and the following characters:

  + - = . _ : /
- Only the bucket owner and authorized users have read and write permissions on object tags. These permissions are independent of object access control lists (ACLs).
- In cross-region replication (CRR), object tags are also replicated to the destination bucket.

Scenarios

You can perform the following operations on multiple objects that are in different directories and have a specific tag:

- Configure lifecycle rules based on a specific tag. For example, when you upload objects, you can configure tagging for temporary objects that are periodically generated. After you configure lifecycle rules for these temporary objects, you can delete these objects based on the tag.
- Use Resource Access Management (RAM) to grant permissions to access objects that have specific tags.

Implementation methods

Implementation method	Description
Console	A user-friendly and intuitive web application
ossutil	A high-performance command-line tool
Java SDK	SDK demos for various programming languages
Python SDK	
Go SDK	
C++ SDK	

Instructions

- API operations related to object tagging
  - PutObjectTagging: configures tagging for an object. If the object already has tags, the existing tags are overwritten.
  - GetObjectTagging: reads tags of an object.
  - DeleteObjectTagging: deletes tags of an object.
  - PutObject: You can use the `x-oss-tagging` request header to specify tags when you upload an object.
  - InitiateMultipartUpload: You can use the `x-oss-tagging` request header to specify tags when you initiate a multipart upload task.
  - CopyObject: You can use the `x-oss-tagging-directive` request header to specify whether to replicate tags of source objects. You can use the `x-oss-tagging` request header to specify tags of destination objects.
  - GetObject: If you have permissions to read the object tags, the tag count is included in the `x-oss-tagging-count` response header.
  - HeadObject: If you have permissions to read the object tags, the tag count is included in the `x-oss-tagging-count` response header.
- Required permissions

  Users, roles, or services that perform operations on tags must have the following permissions:
  - GetObjectTagging: the permission to query object tags. If you have this permission, you can view the existing tags of an object.
  - PutObjectTagging: the permission to configure tagging for objects. If you have this permission, you can configure tagging for objects.
  - DeleteObjectTagging: the permission to delete object tags. If you have this permission, you can delete object tags.

Object tagging and lifecycle management

When you configure lifecycle rules, you can configure conditions for lifecycle rules to select subsets of objects to which the rules apply. You can configure conditions based on the object name prefixes, object tags, or both.

- If you configure tag conditions in one lifecycle rule, the rule applies only to objects that meet both the tag key and value conditions.
- If you configure object name prefixes and multiple object tags in one lifecycle rule, the rule applies only to objects that match the object name prefixes and object tags.

Example:

```
<LifecycleConfiguration>
<Rule>
<ID>r1</ID>
<Prefix>rule1</Prefix>
<Tag><Key>xx</Key><Value>1</Value></Tag>
<Tag><Key>yy</Key><Value>2</Value></Tag>
<Status>Enabled</Status>
<Expiration>
<Days>30</Days>
</Expiration>
</Rule>
<Rule>
<ID>r2</ID>
<Prefix>rule2</Prefix>
<Tag><Key>xx</Key><Value>1</Value></Tag>
<Status>Enabled</Status>
<Transition>
<Days>60</Days>
<StorageClass>Archive</StorageClass>
</Transition>
</Rule>
</LifecycleConfiguration>
```

In the preceding rules,

- Objects whose names are prefixed with rule1 and whose tagging configurations are xx=1 and yy=2 are deleted after the objects are stored for 30 days.
- The storage class of objects whose names are prefixed with rule2 and whose tagging configurations are xx=1 is converted to Archive after the objects are stored for 60 days.

Note For more information, see Lifecycle rules based on the last modified time.

Object tagging and RAM policies

You can authorize RAM users to manage object tags. You can also authorize RAM users to manage objects that have specific tags.

- Authorize RAM users to manage object tags

  You can authorize RAM users to manage all object tags or manage only specific object tags. If User A is authorized to set object tagging to allow=yes, this user can add the tagging configuration of allow=yes to objects. The following code provides an example on how to configure the corresponding RAM policy:

  ```
  {
    "Version": "1",
    "Statement": [
      {
        "Action": "oss:PutObjectTagging",
        "Effect": "Allow",
        "Resource": "*",
        "Condition": {
          "StringEquals": {
            "oss:RequestObjectTag/allow": [
              "yes"
            ]
          }
        }
      }
    ]
  }
  ```

  Notice After the RAM user is authorized to configure a specified tag for objects, the user can configure the tag only for existing objects. However, the user cannot configure the tag for objects when the user uploads the objects.

- Authorize RAM users to manage objects that have specific tags

  You can authorize RAM users to manage all objects that have specific tags. For example, you can authorize User A to access all objects that have the tagging configuration of allow=yes. The following code provides an example on how to configure the corresponding RAM policy:

  ```
  {
    "Version": "1",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": "oss:GetObject",
        "Resource": "*",
        "Condition": {
          "StringEquals": {
            "oss:ExistingObjectTag/allow": [
              "yes"
            ]
          }
        }
      }
    ]
  }
  ```

6.5.6. Manage object metadata

Object Storage Service (OSS) uses object metadata to describe object attributes. Object metadata includes standard HTTP headers and user metadata. You can configure HTTP headers to customize HTTP request policies, such as cache policies and policies for forced object download. You can also configure user metadata to identify the purposes or attributes of objects.

Standard HTTP headers

OSS retains standard HTTP headers for each object that is uploaded to a bucket. The following table describes the standard HTTP headers.

Header	Description
Content-Type	The content type of the object. The browser determines the format and encoding type that are used to read the object based on the content type of the object. If this attribute is not specified, the value is generated based on the extension of the object name. If the object name does not have an extension, the default value `application/octet-stream` is used as the content type of the object. For more information about how to set the content type of an object, see How do I configure the Content-Type of objects?.
Content-Encoding	The encoding method of the object. You must set this parameter based on the encoding type of the object. Otherwise, the browser that serves as the client may fail to parse the encoding type of the object, or the object may fail to be downloaded. If the object is not encoded, leave this parameter empty. Default value: identity. Valid values: - *identity*: OSS does not compress or encode the object. - *gzip*: OSS uses the LZ77 compression algorithm created by Lempel and Ziv in 1977 and 32-bit cyclic redundancy check (CRC) to encode the object. - *compress*: OSS uses the Lempel-Ziv-Welch (LZW) compression algorithm to encode the object. - *deflate*: OSS uses the zlib library and the deflate algorithm to encode the object. - *br*: OSS uses the Brotli algorithm to encode the object.
Content-Language	The language of the object content.
Content-Disposition	The method used to access the object. Valid values: - *inline*: The object is previewed when you access the object. - *attachment*: The object is downloaded to the local computer when you access the object. For example, if this header is set to `attachment; filename="example.jpg"` , the object is downloaded as a file named `example.jpg` when you access the object. **Notice** When you use a browser to access an object in OSS, the object is downloaded even if the Content-Disposition header is set to inline in the following scenarios: - The object is a web page object and is not accessed by using the custom domain name that is mapped to the bucket. - The object is an image, and the bucket in which the object is stored is created after September 23, 2019. In addition, the object is not accessed by using the custom domain name that is mapped to the bucket. - The object cannot be previewed by using the browser.
Cache-Control	The caching behavior for the object. Valid values: - *no-cache*: Each time you access a cached object, the server checks whether the object is updated. If the object is updated, the cache expires. The object must be downloaded from the server again. If the object is not updated, the cache does not expire, and you can directly use the cached object. - *no-store*: All content of the object is not cached. - *public*: All content of the object is cached. - *private*: All content of the object is cached only in the client. - *max-age=<seconds>*: the validity period of the cached content. Unit: seconds. This option is available only in HTTP/1.1.
Expires	The absolute expiration time of the cache in Greenwich Mean Time (GMT). Example: `2022-10-12T00:00:00.000Z` . If `max-age=<seconds>` is set for Cache-Control, `max-age=<seconds>` takes precedence over Expires.
Last-Modified	The time when the object is last modified.
Content-Length	The size of the object. Unit: bytes.

User metadata

When you upload an object, you can add user metadata to identify the purposes or attributes of the object.

- You can configure multiple user metadata parameters for an object. However, the total size of the user metadata of an object cannot exceed 8 KB.
- User metadata is a set of key-value pairs. The name of a user metadata header must start with `x-oss-meta-` . Example: `x-oss-meta-last-modified:20210506` , which indicates that the local file was last modified on May 6, 2021.
- When you call the GetObject operation or the HeadObject operation, the user metadata of the object is returned as HTTP headers.

Implementation methods

The following table describes the methods that you can use to configure, query, and modify the metadata of objects.

Implementation method	Description
Console	A user-friendly and intuitive web application.
ossbrowser	An easy-to-use graphical tool.
ossutil	A high-performance command-line tool.

Implementation method	Description
Java SDK	SDK demos for various programming languages.
Python SDK	
PHP SDK	
Go SDK	
C++ SDK	
C SDK	
.NET SDK	
Node.js SDK	
Android SDK	

6.5.7. Use directories to manage objects

Object Storage Service (OSS) uses a flat structure instead of a hierarchical structure used by traditional file systems to store objects. All elements in OSS are stored as objects in buckets. You can create simulated directories in OSS to help you categorize objects and control access to your objects in a simplified manner.

Structure

OSS uses objects whose names end with a forward slash (/) to simulate directories. The following example shows the structure of a bucket named examplebucket:

```
examplebucket
    └── log/
       ├── date1.txt
       ├── date2.txt
       ├── date3.txt
    └── destfolder/
       └── 2021/
          ├── photo.jpg
```

In the preceding structure:

- The following three objects have the log prefix in their names: log/date1.txt, log/date2.txt, and log/date3.txt. In the OSS console, a directory named log is displayed. Three objects named date1.txt, date2.txt, and date3.txt are stored within the directory.
- The following object has the destfolder prefix in its name: destfolder/2021/photo.jpg. In the OSS console, a directory named destfolder is displayed, which contains a subdirectory named 2021. An object named photo.jpg is stored in the 2021 subdirectory.

Access control based on directories

The following examples show how to grant third-party users different permissions to access the specified directories and objects in examplebucket described in the preceding section:

- The following objects within the log directory store the OSS access logs of a user in the last three days: log/date1.txt, log/date2.txt, and log/date3.txt. Technical support needs to view the logs stored in the three objects to troubleshoot the issues, such as slow access and object upload failures, reported by the user. In this case, you can configure a bucket policy to grant the technical support permissions to access the log directory and objects within the directory. For more information about how to configure a bucket policy, see Configure bucket policies to authorize other users to access OSS resources.
- An object named destfolder/2021/photo.jpg in examplebucket is a group photo of all your employees that was taken on a 2021 spring outing. You want all your employees to have access to the object. In this case, you can set the access control list (ACL) of the object to public read. For more information, see Configure ACL for objects.

Implementation methods

You can create a directory by using only the OSS console, ossbrowser, and ossutil. After you create a directory, you can upload objects to the directory. You can delete directories that you no longer need by using the OSS console, ossbrowser, and ossutil.

Operation	Implementation method
Create directories	Console
	ossutil
	ossbrowser
Delete directories	Console
	ossutil
	ossbrowser

Directories cannot be created or deleted by calling API operations. However, you can use OSS SDKs for various programming languages to create or delete directories by using the following methods:

- When you upload an object to OSS, you can add a directory name that ends with a forward slash (/) to the object name (key) to create a directory for the object. For example, when you upload a local file named localfile.txt to a bucket named examplebucket, you can set the name of the uploaded object to destfolder/localfile.txt. In this case, a directory named destfolder is created in examplebucket, and the uploaded object named localfile.txt is stored in destfolder. In this example, the destfolder directory is simulated by an object whose name is destfolder/ and whose size is 0. For more information about how to use OSS SDK for Java to create directories, see Upload objects.

- When you delete objects, you can specify a prefix that is contained in the names of all objects you want to delete. In this case, the directory whose name is the specified prefix and all objects within the directory are deleted. For example, if you specify a prefix "log", the directory named log and all objects within the directory are deleted. For more information about how to use OSS SDK for Java to delete directories, see Delete objects.

6.5.8. Call the SelectObject operation to query objects

You can call the SelectObject operation to execute SQL statements on an object and return the execution result.

Background information

Hadoop 3.0 is supported on Object Storage Service (OSS). You can directly read and process data in OSS when you run services such as Spark, Hive, and Presto in E-MapReduce. Alibaba Cloud services such as MaxCompute and Data Lake Analytics (DLA) are also supported.

However, the current GetObject operation provided by OSS requires the big data platform to download all OSS data locally for analysis and filtering. As a result, large amounts of bandwidth and client resources are wasted in many query scenarios.

To resolve this issue, OSS provides the SelectObject operation. SelectObject allows OSS to preliminarily filter data by using conditions and projections provided by the big data platform. As a result, only useful data is returned to the big data platform. This way, the client can consume fewer bandwidth resources and process less data to maximize CPU and memory resources. This makes OSS-based data warehousing and data analysis a better option.

Detail analysis

The following section describes the object types and SQL statements supported by SelectObject in detail.

- Object types supported by SelectObject

  > **Note** Use SelectObject for normal objects. We recommend that you do not use SelectObject to query Multipart and Appendable objects. The differences in their internal structures may deteriorate query performance.

  - CSV objects (and CSV-like objects such as TSV objects) that conform to RFC 4180. You can customize row and column delimiters and quote characters in CSV objects.
  - UTF-8 encoded JSON objects. SelectObject supports JSON objects in DOCUMENT and LINES formats.
    - A JSON DOCUMENT object contains a single object.
    - A JSON LINES object consists of lines of objects separated by row delimiters. However, the complete JSON object itself may not be valid. SelectObject supports typical delimiters such as \n and \r\n. You do not need to specify these delimiters.
  - Standard and Infrequent Access (IA) objects. You must restore Archive and Cold Archive objects before you access them.
  - Objects fully managed and encrypted by OSS or encrypted by using customer master keys (CMKs) managed by Key Management Service (KMS).

- Supported SQL syntax
  - SQL statement: SELECT FROM WHERE
  - Data types: string, int(64bit), double(64bit), decimal(128), timestamp, and bool.
  - Operators: logical operators (AND, OR, and NOT), arithmetic operators (+, -, *, /, and %), comparison operators (>, =, <, >=, <=, and !=),and string operators (LIKE and ||).

    > **Notice** The matching is case-sensitive when you use LIKE for fuzzy matches.

- Multipart query

  SelectObject supports multipart query similar to byte-based multipart download supported by the GetObject operation. Data is split into parts by row or split.

  - By row: This method is used in most cases but may result in unbalanced loads when sparse data is split.
  - By split: A split includes multiple rows. Each split is about the same size.

  > **Note** The method of splitting data by split is more efficient.

- Data types

  In OSS, data in CSV objects is of the STRING type by default. You can use the CAST function to convert the data type. For example, the following SQL statement converts the data in the first and second columns to the INTEGER type and compares them:

  ```
  Select * from OSSObject where cast (_1 as int) > cast(_2 as int)
  ```

  In addition, SelectObject allows you to implicitly convert the data type in a WHERE clause. For example, the following SQL statement converts the data in the first and second columns to the INTEGER type:

  ```
  Select _1 from ossobject where _1 + _2 > 100
  ```

  If you do not use the CAST function, the data type of a JSON object is determined by the type of data in the object. A standard JSON object can support data types such as NULL, BOOLEAN, Int64, DOUBLE, and STRING.

SQL statement examples

SQL statement examples are provided for CSV and JSON objects.

- SQL statement examples for CSV objects

Scenario	SQL statement
Return the first 10 rows.	SELECT * FROM ossobject limit 10
Return integers in the first and third columns, in which the values of the integers in the first column are greater than those in the third column.	SELECT _1, _3 FROM ossobject WHERE CAST(_1 AS INT) > CAST(_3 AS INT)
Return the number of records in which the data in the first column starts with X. A Chinese character specified after LIKE must be UTF-8 encoded.	SELECT COUNT(*) FROM ossobject WHERE _1 LIKE 'X%'

Scenario	SQL statement
Return all records in which the time of the data in the second column is later than 2018-08-09 11:30:25 and the data in the third column is greater than 200.	SELECT * FROM ossobject WHERE _2 > CAST('2018-08-09 11:30:25' AS TIMESTAMP) AND _3 > 200
Return the average value, sum, the maximum value, and the minimum value of the floating-point numbers in the second column.	SELECT AVG(CAST(_2 AS DOUBLE)), SUM(CAST(_2 AS DOUBLE)), MAX(CAST(_2 AS DOUBLE)), MIN(CAST(_2 AS DOUBLE))
Return all records in which the strings concatenated by the data in the first and third columns start with Tom and end with Anderson.	SELECT * FROM ossobject WHERE (_1 || _3) LIKE 'Tom%Anderson'
Return all records in which the data in the first column is divisible by 3.	SELECT * FROM ossobject WHERE (_1 % 3) = 0
Return all records in which the data in the first column ranges from 1995 to 2012.	SELECT * FROM ossobject WHERE _1 BETWEEN 1995 AND 2012
Return all records in which the data in the fifth column is N, M, G, or L.	SELECT * FROM ossobject WHERE _5 IN ('N', 'M', 'G', 'L')
Return all records in which the product of the data in the second and third columns is greater than the sum of 100 and the data in the fifth column.	SELECT * FROM ossobject WHERE _2 * _3 > _5 + 100

- SQL statement examples for JSON objects

  An example of a JSON object:

```
{
  "contacts":[
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
    {
      "type": "mobile",
      "number": "123 456-7890"
    }
  ],
  "children": [],
  "spouse": null
}, …… # Other similar nodes are omitted.
]}
```

  The following table describes the SQL statement examples.

Scenario	SQL statement
Return all records in which the value of age is 27.	SELECT * FROM ossobject.contacts[*] s WHERE s.age = 27
Return all home phone numbers.	SELECT s.number FROM ossobject.contacts[*].phoneNumbers[*] s WHERE s.type = "home"
Return all records in which the value of spouse is null.	SELECT * FROM ossobject s WHERE s.spouse IS NULL
Return all records in which the number of children is 0.	SELECT * FROM ossobject s WHERE s.children[0] IS NULL Note The preceding statement is used because an empty array cannot be specified in other ways.

Scenarios

In most cases, SelectObject is used for the multipart query, JSON object query, and analysis of log objects.

- Query large objects in multipart query.

  If columns in a CSV object do not include line feeds, you can divide the object into parts based on bytes. This method is the simplest because you do not need to create Select Meta for the object. To query a JSON or CSV object where columns include line feeds, perform the following steps:

  i. Call the CreateSelectObjectMeta operation to obtain the total number of splits for the object. Before you call SelectObject for the object, asynchronously call the CreateSelectObjectMeta operation to shorten the scanning time.

   ii. Select the appropriate concurrency level (n) based on resources on the client. Divide the total number of splits by concurrency level (n) to obtain the number of splits to contain in each query.

   iii. Set parameters, such as split-range=1-20, in the request body to perform multipart query.

   iv. Combine the results.

- When you query a JSON object, narrow down the JSON path range in the FROM clause.

  An example of a JSON object:

```
{ contacts:[
        {"firstName":"John", "lastName":"Smith", "phoneNumbers":[{"type":"home", "number":"212-555-1234"}, {"type":"office", "number":"646-555-4567
"}, {"type":"mobile", "number":"123 456-7890"}], "address":{"streetAddress": "21 2nd Street", "city":"New York", "state":NY, "postalCode":"10021-31
00"}
         }
]}
```

  To query all streetAddress data of records in which the postal code starts with 10021, execute the following SQL statement: `SELECT s.address.streetAddress FROM ossobject.contacts[*] s WHERE s.address.postalCode LIKE '10021%'` or `SELECT s.streetAddress FROM ossobject.contacts[*].address s WHERE s.postalCode LIKE '10021%'` .

  The performance of the JSON path is better because it is more accurate when you execute `SELECT s.streetAddress FROM ossobject.contacts[*].address s WHERE s.postalCode LIKE '10021%'` .

- Process high-precision floating-point numbers in JSON objects.

  If you want to calculate high-precision floating-point numbers in a JSON object, we recommend that you set the ParseJsonNumberAsString parameter to true, and use the CAST function to convert the parsed data to the DECIMAL type. For example, if the value of attribute a is 123456789.123456789, you can execute `SELECT s.a FROM ossobject s WHERE CAST(s.a AS DECIMAL) > 123456789.12345` to maintain the accuracy of attribute a.

APIs and SDKs

- API: SelectObject
- OSS SDK for Java: Query objects
- OSS SDK for Python: Query objects

6.6. FAQ

6.6.1. Does OSS throttle bandwidth and QPS?

Object Storage Service (OSS) throttles bandwidth and queries per second (QPS) throttling when you upload data to and download data from OSS.

You can use ossutil to run the probe command to check network conditions. For more information, see probe.

Item	Description
Bandwidth	Default bandwidth limit: 10 Gbit/s in mainland China regions and 5 Gbit/s in regions outside mainland China. If this limit is reached, requests are throttled. **Note** When a request is throttled, the response to the request contains the `x-oss-qos-delay-time: number` header in which `number` indicates the duration over which the request is throttled. Unit: ms. For upload requests, the exact duration over which a request is throttled is returned. For download requests, the estimated duration over which a request is throttled is returned. The duration is estimated based on the extent of throttling and the downloaded object size. If you require a higher bandwidth (10 Gbit/s to 100 Gbit/s) for your business such as big data offline processing, contact technical support.
Queries per second (QPS)	The limit of the total QPS for a single account is 10,000. The actual values that can be achieved are different in the different read and write modes: • Sequential read and write: 2,000 If you use sequential prefixes such as timestamps or alphabetical letters in the names of large numbers of objects, multiple object indexes may be stored in a single partition. In this case, when you send a large number of requests to query these objects, latency may increase. We recommend that when you upload a large number of objects, do not use sequential prefixes in the names of the objects. For more information about how to change sequential prefixes to random prefixes, see OSS performance and scalability best practices. • Non-sequential read and write: 10,000 If you require a higher QPS, contact technical support.

6.6.2. How do I obtain the URL of an uploaded object?

This topic describes how to obtain the URL of an uploaded object.

Public read object

If the access control list (ACL) of an object is public read, the object can be accessed by anonymous users. The URL of the object is in the following format: `https://BucketName.Endpoint/ObjectName` . In the preceding URL, ObjectName is the full path of the object that includes the object prefix and suffix. For more information about the endpoints of each region, see Regions and endpoints.

For example, an object named example.jpg is stored within the example directory of a bucket named bucketexample, which is located in the China (Hangzhou) region. The following URLs can be used to access the object:

- URL for access over the Internet: `https://bucketexample.oss-cn-hangzhou.aliyuncs.com/example/example.jpg` .
- URL for access over the internal network (from Elastic Compute Service (ECS) instances that are located in the same region as the object): `https://bucketexample.oss-cn-hangzhou-internal.aliyuncs.com/example/example.jpg`

Notice To make sure that an image object is previewed when you access the image object, you must map a custom domain name to your bucket and add a Canonical Name (CNAME) record. For more information, see Use a custom domain name to access OSS resources.

Private object

If the ACL of an object is private, the URL of the object must be signed. The URL of a private object is in the following format: `https://BucketName.Endpoint/Object?SignatureParameters`. You can use the following methods to obtain the URL of a private object and set the validity period of the URL:

- Console

  You can obtain the URL of a private object in the Object Storage Service (OSS) console. For more information, see Share objects. The validity period of an object URL can be set to different value ranges by different accounts. Alibaba Cloud accounts can set the validity period of an object URL to a value up to 32,400 seconds (9 hours). Resource Access Management (RAM) users and temporary users authorized by Security Token Service (STS) can set the validity period of an object URL to a value up to 3,600 seconds (1 hour). To set the validity period of an object URL to a greater value, you can use ossutil, ossbrowser, or OSS SDKs.

- ossutil

  For more information about how to use ossutil to obtain the URL of a private object and set the validity period of the URL, see ossutil-sign.

- ossbrowser

  For more information about how to use ossbrowser to obtain the URL of a private object and set the validity period of the URL, see ossutil-sign.

- SDKs

  For more information about how to use OSS SDKs to obtain the URL of a private object and set the validity period of the URL, see the following topics:

  - OSS SDK for Java
  - OSS SDK for Python
  - OSS SDK for Go
  - OSS SDK for PHP
  - OSS SDK for C
  - OSS SDK for .NET
  - OSS SDK for Android
  - OSS SDK for iOS
  - OSS SDK for Node.js
  - OSS SDK for Browser.js

Objects in a bucket to which a custom domain name is mapped

If the bucket in which an object is stored is mapped to a custom domain name, the URL of the object is in the following format: `https://YourDomainName/ObjectName`. In the preceding URL, ObjectName is the full path of the object that includes the object prefix and suffix.

For example, a bucket named bucketexample is located in the China (Hangzhou) region, and the following custom domain name is mapped to the bucket: `img.example.com`. In this bucket, an object named *example.jpg* is stored in a directory named example. You can use the following URL to access the object: `https://img.example.com/example/example.jpg`.

6.6.3. How do I configure the Content-Type of objects?

Content-Type specifies a standard Multipurpose Internet Mail Extensions (MIME) type that describes the format of received or sent data. A browser determines how to open the data based on the value of the header. In general, this header is used to specify how to open files customized by clients or media files.

By default, Object Storage Service (OSS) automatically identifies the type of an object. For example, if the type of an uploaded object is JPG, OSS automatically identifies the object as an image object. You can modify the type of an object by using the OSS console, ossbrowser, ossutil, or OSS SDKs for various programming languages such as Java SDK, Python SDK, PHP SDK, and Go SDK. For more information, see Configure object metadata, Use ossbrowser, and set-meta.

The following table describes the common values of Content-Type.

File extension	Content-Type(Mime-Type)	File extension	Content-Type(Mime-Type)
(binary stream or unknown file type)	application/octet-stream	.tif	image/tiff
0.001	application/x-001	0.301	application/x-301
0.323	text/h323	0.906	application/x-906
0.907	drawing/907	.a11	application/x-a11
.acp	audio/x-mei-aac	.ai	application/postscript
.aif	audio/aiff	.aifc	audio/aiff
.aiff	audio/aiff	.anv	application/x-anv
.apk	application/vnd.android.package-archive	.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

File extension	Content-Type(Mime-Type)	File extension	Content-Type(Mime-Type)
.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
.docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document	.dot	application/msword
.dotx	application/vnd.openxmlformats-officedocument.wordprocessingml.template	.drw	application/x-drw
.dtd	text/xml	.dwf	Model/vnd.dwf
.dwf	application/x-dwf	.dwg	application/x-dwg
.dxb	application/x-dxb	.dxf	application/x-dxf
.edn	application/vnd.adobe.edn	.emf	application/x-emf
.eml	message/rfc822	.ent	text/xml
.epi	application/x-epi	.eps	application/x-ps
.eps	application/postscript	.etd	application/x-ebx
.exe	application/x-msdownload	.fax	image/fax
.fdf	application/vnd.fdf	.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	.ipa	application/vnd.iphone
.isp	application/x-internet-signup	.IVF	video/x-ivf

File extension	Content-Type(Mime-Type)	File extension	Content-Type(Mime-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
.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	.pdb	chemical/x-pdb
.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

File extension	Content-Type(Mime-Type)	File extension	Content-Type(Mime-Type)
.png	application/x-png	.pot	application/vnd.ms-powerpoint
.potx	application/vnd.openxmlformats-officedocument.presentationml.template	.ppa	application/vnd.ms-powerpoint
.ppm	application/x-ppm	.pps	application/vnd.ms-powerpoint
.ppsx	application/vnd.openxmlformats-officedocument.presentationml.slideshow	.ppt	application/vnd.ms-powerpoint
.ppt	application/x-ppt	. pptx	application/vnd.openxmlformats-officedocument.presentationml.presentation
.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	.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
.sis	application/vnd.symbian.install	.sisx	application/vnd.symbian.install
.sit	application/x-stuffit	.slb	application/x-slb
.sld	application/x-sld	.sldx	application/vnd.openxmlformats-officedocument.presentationml.slide
.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	image/svg+xml
.swf	application/x-shockwave-flash	.tdf	application/x-tdf
.tg4	application/x-tg4	.tga	application/x-tga
.tif	image/tiff	.tif	application/x-tif
.tiff	image/tiff	.tld	text/xml

File extension	Content-Type(Mime-Type)	File extension	Content-Type(Mime-Type)
.top	drawing/x-top	.torrent	application/x-bittorrent
.tsd	text/xml	.txt	text/plain
.uin	application/x-icq	.uls	text/iuls
.vcf	text/x-vcard	.vda	application/x-vda
.vdx	application/vnd.visio	.vml	text/xml
.vpg	application/x-vpeg005	.vsd	application/vnd.visio
.vsd	application/x-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
.ws2	application/x-ws	.wsc	text/scriptlet
.wsdl	text/xml	.wvx	video/x-ms-wvx
.xap	application/x-silverlight-app	.x_b	application/x-x_b
.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	.xlsx	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
.xltx	application/vnd.openxmlformats-officedocument.spreadsheetml.template	.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_t	application/x-x_t
.yaml	text/vnd.yaml	.yml	text/vnd.yml
.webp	image/webp	N/A	N/A

6.6.4. What are the OSS batch operations?

OSS provides multiple methods to access and manage objects. This topic describes how to manage multiple objects at a time.

Upload multiple objects at a time

You can use the following methods to upload multiple objects at a time:

- Use ossimport

  You can use ossimport to migrate data to OSS from data sources such as local servers, third-party cloud storage services such as S3, Azure Blob, and Tencent COS, and OSS. ossimport is especially suitable for scenarios with large amounts of data. For more information, see Architectures and configurations.

- Use ossutil

  You can upload multiple objects to OSS at a time by adding the -r(--recursive) option to the **cp** command of ossutil. For more information, see Upload objects.

- Use ossbrowser

  You can use ossbrowser to select multiple objects at a time and upload them to OSS. For more information, see Upload objects.

- Use the OSS console

  You can use the OSS console to select multiple objects at a time and upload them to OSS. For more information, see Upload objects.

Download multiple objects at a time

You can use the following methods to download multiple objects at a time:

- Use ossutil

  You can download multiple objects at a time from the specified directory to your local device by adding the -r(--recursive) option to the **cp** command of ossutil. For more information, see Download objects.

- Use ossbrowser

  Use ossbrowser to select multiple objects or directories and download them to your local device at a time. For more information, see Download objects.

- Use the OSS console

  You can use the OSS console to select multiple objects and download them to your local device at a time. For more information, see Download objects.

Copy multiple objects at a time

You can use the following methods to copy multiple objects at a time:

- Use cross-region replication (CRR)

  You can use CRR to copy objects whose names contain a specific prefix at a time. You can also choose whether to synchronize historical data and whether to synchronize the delete operations. For more information, see Configure CRR.

- Use ossutil

  You can copy multiple objects at a time from the specified directory to another directory or to another bucket under the same account by adding the -r(--recursive) option to the **cp** command of ossutil. For more information, see Copy objects.

- Use ossbrowser

  You can use ossbrowser to select multiple folders or objects, and copy one or more objects to another directory or another bucket in the same account. For more information, see Copy objects.

Delete multiple objects at a time

You can use the following methods to delete multiple objects at a time:

> **Warning** You cannot recover deleted objects. Exercise caution when you perform this operation.

- Use OSS SDKs

  You can use SDKs to delete multiple objects at a time.

  - Java SDK
  - Python SDK
  - Go SDK
  - C++ SDK

  For the SDK examples for other programming languages, see Introduction.

- Use OSS APIs

  You can call the DeleteMultipleObjects operation to delete multiple objects at a time. For more information, see DeleteMultipleObjects.

- Use ossutil

  You can delete multiple objects whose names contain a specific prefix by adding the -r(--recursive) option to the **rm** command of ossutil. For more information, see Delete objects.

- Use ossbrowser

  You can use ossbrowser to select multiple objects or folders and delete them at a time. For more information, see Delete objects.

- Use the OSS console

  - You can use the OSS console to select multiple objects and delete them at a time. For more information, see Delete objects.

    You can also directly delete a folder. All objects in the folder are deleted at the same time.

  - You can use the fragment management function of the OSS console to delete multiple parts at a time. For more information, see Manage parts.

- Configure lifecycle rules

  You can configure lifecycle rules to automatically delete multiple objects at a time based on the lifecycle rules. For more information, see Lifecycle rules based on the last modified time.

Modify the storage classes of multiple objects at a time

You can use the following methods to modify the storage classes of multiple objects at a time:

- Use ossutil

  You can modify the storage classes of specified objects at a time by adding the -r(--recursive) option to the **set-meta** command of ossutil. For more information, see set-meta.

- Configure lifecycle rules

  You can configure lifecycle rules to automatically convert the storage classes of multiple objects at a time. For more information, see Lifecycle rules based on the last modified time.

Modify the access control lists (ACL) of multiple objects at a time

You can use ossutil to modify the ACLs of multiple objects at a time.

- You can modify the ACLs of multiple objects at a time by adding the -r(--recursive) option to the **set-acl** command of ossutil. For more information, see set-acl (configure or modify object or bucket ACLs).
- You can modify the ACLs of multiple objects at a time by modifying the metadata of the specified objects by adding the -r(--recursive) option to the **set-meta** command of ossutil. For more information, see set-meta.

Restore multiple objects at a time

You can use the following methods to restore multiple archives from the frozen state to the readable state at a time:

- Use ossutil

  You can restore multiple objects from the frozen state to the readable state at a time by adding the-r(--recursive) option to the **restore** command of ossutil. For more information, see restore (restore objects).

- Use ossbrowser

  You can use ossbrowser to select the objects to be restored and restore them at a time.

Configure the metadata of multiple objects at a time

You can use the following methods to modify the metadata of multiple objects at a time:

- Use ossutil

  You can modify the metadata of the specified objects at a time by adding the -r(--recursive) option to the **set-meta** command of ossutil. For more information, see set-meta.

  You can use this command to modify the storage classes and ACLs of multiple objects at a time.

- Use the OSS console

  You can modify the metadata of multiple objects at a time by selecting the objects for which you want to modify the information about HTTP headers. For more information, see Configure object metadata.

6.6.5. How do I upload and download folders to and from OSS?

OSS does not use a hierarchical structure for objects, but instead uses a flat structure. All elements are stored as objects in buckets. Therefore, OSS does not have folders and subfolders like a hierarchical file system would. However, OSS supports folders as a concept to group objects and simplify management. In the OSS console, a folder is an object whose name ends with a forward slash (/), which is similar to folders in Windows.

For example, a file whose path is *abc/efg/123.jpg* is the *123.jpg* object contained in the efg subfolder of the abc folder in the OSS console.

You can use the following methods to upload or download a folder:

- OSS console: a graphical management tool similar to the Windows resource manager.
  - Upload a folder: When you upload a folder, drag the folder to the upload section. The folder structure remains. For more information, see Upload objects.
  - Download a folder: OSS console does not support the direct download of folders. You can download multiple objects from a bucket to the specified folder that is created on the local computer. For more information, see Download objects.
- ossbrowser: a graphical management tool, which is easy to use. This tool is similar to the Windows resource manager.
  - Upload a folder: In the specified bucket or folder, click **Folder**. Select the folder you want to upload. You can also drag the folder to ossbrowser. For more information, see Upload objects.
  - Download a folder: Click Download in the Actions column corresponding to the folder. For more information, see Download objects.
- ossutil: a command line management tool that provides a wide range of simple and convenient commands to manage OSS data while high performance of operations is ensured.
  - Upload a folder: Include the -r option when you upload a folder. For more information, see Upload objects.
  - Download a folder: Include the -r option when you download a folder. For more information, see Download objects.
- SDK: provides OSS SDK demos in various programming languages to facilitate development.
  - Upload a folder: OSS SDKs do not support the direct upload of folders. However, you can set the same prefix for object names and separate each folder level with a forward slash (/) when you upload the objects. For example, when you upload the *a.txt, b.txt, and c.txt* objects to the abc folder, set the ObjectName parameter to *abc/a.txt, abc/b.txt, and abc/c.txt*.
  - Download a folder: OSS SDKs do not support the direct download of folders. However, you can download multiple objects to the same local folder.

6.6.6. How do I limit object formats and sizes when I upload objects to OSS?

Object Storage Service (OSS) does not limit object formats and sizes when you upload objects to OSS. If you want to impose limits on object formats and sizes, you must do it at your business level. This topic describes how to limit object formats and sizes when you use browsers or applications to directly upload objects to OSS.

Prerequisites

The environment for the direct data transfer to OSS by using browsers or applications is prepared.

For more information about direct data transfer to OSS, see Add signatures on the client by using JavaScript and upload data to OSS.

Procedure

You can use Plupload filters to set upload conditions, such as uploading only images, the size of objects to upload, and disabling repeated uploads of an object.

1. Open the *upload.js* file.
2. Add the following fields after `var uploader = new plupload.Uploader` and save the configurations:

```
    filters: {
        mime_types : [ // Only images and ZIP files can be uploaded.
        { title : "Image files", extensions : "jpg,gif,png,bmp" },
        { title : "Zip files", extensions : "zip" }
        ],
        max_file_size : '400kb', // The size of the object to upload cannot exceed 400 KB.
        prevent_duplicates : true // Objects cannot be uploaded repeatedly.
    },
```

   - mime_types: limits the extensions of objects to upload.
   - max_file_size: limits the sizes of objects to upload.
   - prevent_duplicates: specifies that an object cannot be uploaded repeatedly.
3. Open the *index.html* file and test uploads.

   Click **Select Files**. Only files that are in the following formats and are 400 KB or smaller in size can be selected: JPG, GIF, PNG, BMP, and ZIP.

6.6.7. Can I use ETag values as OSS MD5 hashes to check data consistency?

Objects in Object Storage Service (OSS) have ETag values that are used to identify whether changes are made to data on the server. However, these ETag values are not necessarily equal to the MD5 hashes of the objects. We recommend that you do not use ETag values to verify data consistency.

To check whether an uploaded object in OSS is consistent with the local file, you can include the Content-MD5 header value in the upload request. When OSS receives the object, OSS compares the MD5 hash with the Content-MD5 header value. The object can be uploaded only when the MD5 hash is consistent with the Content-MD5 header value. This way, data consistency is ensured.

The following examples use a string "123456789" to show how to calculate the Content-MD5 value of the request content:

- Correct calculation
  i. Calculate the MD5 hash of the string, which is a 128-bit binary array.
  ii. Encode the binary array (instead of the 32-bit string) in Base64.

  The following Python code provides an example on how to calculate the Content-MD5 value:

```
>>> import base64,hashlib
>>> hash = hashlib.md5()
>>> hash.update("0123456789")   // If you use Python 3, change the code to hash.update(b"0123456789").
>>> base64.b64encode(hash.digest())
'eB5eJF1ptWaXm4bijSPyxw=='
```

  Call hash.digest() to calculate the 128-bit binary array.

```
>>> hash.digest()
'x\x1e^$]i\xb5f\x97\x9b\x86\xe2\x8d#\xf2\xc7'
```

- Incorrect calculation

  Note A common incorrect operation is to encode the calculated 32-bit string in Base64 to obtain the Content-MD5 value.

```
# Call hash.hexdigest() to obtain a 32-bit plaintext string.
>>> hash.hexdigest()
'781e5e245d69b566979b86e28d23f2c7'
# The following code provides an example on encoding the incorrect MD5 hash in Base64:
>>> base64.b64encode(hash.hexdigest())
'NzgxZTVlMjQ1ZDY5YjU2Njk3OWI4NmUyOGQyM2YyYzc='
```

6.6.8. How do I compress objects that I download from OSS in the GZIP format?

When you send a GET request to Object Storage Service (OSS) to download a static object used in web pages, such as an HTML, JavaScript, XML, or JSON object, you can add the Accept-Encoding header to the GET request and set the value of this header to gzip. This way, the object is compressed in the GZIP format before being downloaded.

Prerequisites

- The object that you download is equal to or larger than 1 KB in size.
- The Content-Type header in the GET request is set to one of the following values: text/cache-manifest, text/xml, text/plain, text/css, application/javascript, application/x-javascript, application/rss+xml, application/json, or text/json.

Examples

- Sample requests

```
GET /ossutil.txt HTTP/1.1
Host: agent-test1.oss-cn-qingdao.aliyuncs.com
User-Agent: curl/7.47.0
Accept: */*
Accept-Encoding: gzip
```

- Sample responses

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Thu, 23 May 2019 02:03:39 GMT
Content-Type: text/plain; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
x-oss-request-id: 5CE5FF7BFEC931F2900F9F2A
Last-Modified: Thu, 23 May 2019 02:01:11 GMT
x-oss-object-type: Normal
x-oss-hash-crc64ecma: 316181249502703****
x-oss-storage-class: Standard
Content-MD5: XSpWpgD//mzytMaVJCE7****
x-oss-server-time: 0
Content-Encoding: gzip
[965 bytes data]
```

6.6.9. How do I configure an HTTPS request and an SSL certificate?

To use a custom domain name to access Object Storage Service (OSS) resources over HTTPS, you must purchase an SSL certificate and host your certificate in OSS. For more information, see Host SSL certificates.

By default, authorized users can access OSS resources over both HTTP and HTTPS. If a bucket owner whose UID is 175708322470**** wants anonymous users to access resources in the examplebucket bucket over HTTPS, configure the following bucket policy by specifying policy syntax:

```
{
  "Version": "1",
  "Statement": [{
      "Effect": "Deny",
      "Action": [
        "oss:*"
      ],
      "Principal": [
        "*"
      ],
      "Resource": [
        "acs:oss:*:175708322470****:examplebucket",
        "acs:oss:*:175708322470****:examplebucket/*"
      ],
      "Condition": {
        "Bool": {
          "acs:SecureTransport": [
            "false"
          ]
        }
      }
    }
      ]
  }
```

For more information about elements involved in policy syntax, see Overview.

For more information about bucket policies, see Configure bucket policies to authorize other users to access OSS resources.

6.6.10. Can I recover an OSS object after the object is deleted or overwritten?

The redundancy mechanism of the OSS backend is used to recover data when server or hardware failures occur. Alibaba Cloud cannot recover OSS data that is manually deleted, overwritten, or automatically deleted by configuration rules.

Data deletion and overwriting

OSSThe following items are the description about user data management in Service Terms and Security Center Service Level Agreement (SLA).

- The following items are the description about user data management in Service Terms:
  - You can delete, change, and manage your business data. If you manually release services or delete data, Alibaba Cloud does not retain the data.
  - If the user business data is deleted, the data cannot be recovered. You assume the consequences and responsibilities that result from the deletion of such data on your own. You understand and agree that Alibaba Cloud has no obligation to continue to retain, export, or return the user business data.
- The following content is the description about data destructibility in SLA:

  If you manually delete data or want to delete data when your services expire, Alibaba Cloud automatically deletes disk data and clears memory on the corresponding physical server. The data cannot be recovered.

Operations that may cause data to be deleted or overwritten

Your data may be deleted or overwritten when you use the following methods. Proceed with caution.

- Delete objects by using the OSS console, ossutil, ossbrowser, or SDKs. For more information, see Delete objects.
- Upload an object that has the same name as an existing object to OSS by using the OSS console, ossutil, ossbrowser, or SDKs. The existing object is overwritten by the uploaded object.
- Configure lifecycle rules to delete objects on a regular basis. OSS automatically deletes objects based on the lifecycle rules. For more information, see Configure lifecycle rules.
- Configure cross-region replication (CRR) rules for your bucket and set **Add/Delete/Change** to synchronize data from the source bucket to your bucket. If objects in the source bucket are modified or deleted, the changes are synchronized to the destination bucket. For more information, see Configure CRR.
- Other users delete or overwrite your objects because access permissions on the bucket are improperly configured. For more information about access permissions, see

Overview.

How do I avoid accidental operations on my objects?

You can use one of the following methods to prevent your objects from being deleted or overwritten:

- Enable versioning

  When you enable versioning for a bucket, the objects that are deleted or overwritten are stored as previous versions in the bucket. You can recover an object to a previous version at any time. For more information, see Overview.

- Use CRR to back up objects

  You can configure CRR rules for your bucket and set **Add/Delete/Change** to synchronize data from your bucket to another bucket. For more information, see Configure CRR.

- Configure scheduled backup

  You can use Hybrid Backup Recovery (HBR) to back up your objects. This allows you to recover your objects when the objects are lost. For more information, see Configure scheduled backup.

- Configure appropriate access permissions

  Take note of the following principles to grant appropriate access permissions to other users who access your bucket:

  - Do not use your Alibaba Cloud account to access OSS.
  - Grant read and write permissions to different RAM users. Use a RAM user that has only read permissions or an Security Token Service (STS) temporary access credential to access read-only data.
  - We recommend that you provide an STS temporary access credential to the users who need to only temporarily access your data.
  - Grant the least but sufficient access permissions on OSS data for different businesses.
  - Use a secure location to store credentials for data access such as the password of your Alibaba Cloud account and the access credentials of a RAM user.

  For more information, see Overview.

6.6.11. Why anonymous users cannot access public-read objects?

After the access control lists (ACLs) of your objects are set to public read, all users can access your objects. However, the following configurations can cause anonymous users to fail to access public-read objects in a bucket:

Pay-by-requester

After you have pay-by-requester enabled, requesters must pay fees for requests and traffic generated by reading data in the bucket. The bucket owner is charged only the storage fees of the bucket. Therefore, requesters must provide identity information for authentication so that Object Storage Service (OSS) can identify and charge requesters for request and traffic fees. Anonymous users do not carry identity information for authentication when they access your buckets, so OSS denies requests from anonymous users. For more information, see Enable pay-by-requester.

Solution:

- The bucket owner can generate a signed URL for the required object and provide the anonymous user with the signed URL. For more information, see How do I obtain the URL of an uploaded object?.
- The bucket owner can disable pay-by-requester. For more information, see Enable pay-by-requester.

Bucket policy

Bucket policies are used to authorize other users to access your OSS resources. Therefore, some bucket policies that you configure for a bucket may cause anonymous users failed access to the bucket. For more information about bucket policies, see Configure bucket policies to authorize other users to access OSS resources.

Solution:

View the bucket policies that you configure for the bucket. Modify or delete the bucket policies that prevent anonymous users from accessing the bucket.

6.6.12. Why can an expired signed object URL be used to access object?

In most cases, you cannot access an object by using an expired signed object URL. However, if the URL is opened in your local browser, you can access the object by using the browser cache while the browser cache is valid.

You can modify the object metadata to specify that a signed object URL becomes invalid when it expires. You can set Cache-Control to `no-cache` or set Expires to a value smaller than the expiration time of the signed object URL. For more information, see Manage object metadata.

> **Notice** After you set the Cache-Control or Expires parameter, the browser stops caching OSS objects or the validity periods of caches decrease. Consequently, the request count and outbound traffic over Internet may increase.

6.6.13. What are the operations that affect the LastModified attribute of OSS objects?

LastModified (last modified time) is an important attribute of OSS objects. It is used in billing, incremental migration, and lifecycle rules. When you perform some operations on objects, the LastModified values of the objects are updated. The following table lists the common operations on objects.

Common operation	API operation	Whether the LastModified value of the object is updated
Modify object ACLs	CopyObject	Yes
	PutObjectACL	Yes
Modify the user metadata of an object	CopyObject	Yes
Modify the storage class of an object	CopyObject	Yes

Common operation	API operation	Whether the LastModified value of the object is updated
Modify the storage class of an object	CommitTransition	No
Modify the encryption algorithm for an object	CopyObject	Yes
Overwrite an object	PutObject	Yes
	CopyObject	Yes
Add or modify object tags	PutObjectTagging	No
Delete an object tag	DeleteObjectTagging	No
Restore an Archive or Cold Archive object	RestoreObject	No

Notice

- When you modify the storage class of an object by using the OSS console, ossutil, ossbrowser, or SDKs, OSS generates a new object of the specified storage class to overwrite the original object.
- The last modified time of an object is updated when the object is overwritten or the storage class and encryption method are changed by using CopyObject. If the object whose last modified time is updated is an IA, Archive, or Cold Archive object and is stored for a period less than the minimum storage duration, you are charged for the minimum storage duration. For more information, see Storage fees.

  For example, if you change the storage class of an object from IA to Archive by using the OSS console after the object is stored for 12 days, the last modified time of the object is updated. In addition, you are charged for the minimum storage duration of an IA object, which is 30 days.

6.6.14. Can I modify the expiration time of a signed URL?

After a signed URL is generated, it cannot be modified. If you want to modify the validity period of a signed URL, generate a signed URL. For more information, see Generate a signed URL.

6.6.15. How do I modify, update, and edit objects?

OSS does not provide native support for you to edit objects (except for data uploaded by append upload). To modify an uploaded object, you must call the PutObject operation to upload a new object.

6.6.16. Can OSS buckets be renamed or OSS objects be migrated?

No, OSS buckets cannot be renamed, and OSS objects cannot be migrated. To use another bucket name, we recommend that you create a bucket, migrate the objects from the source object to the destination (new) bucket, and delete the source bucket.

If the number of objects in the source bucket is small, you can copy the objects to migrate your data. For more information, see Copy objects.

If the number of objects in the source bucket is large, use the following methods to migrate your data:

- Use cross-region replication (CRR). For more information, see Cross-region replication.
- Use Data Online Migration. For more information, see Architectures and configurations.
- Use ossimport. For more information, see ossimport.

6.6.17. What do I do if "TypeError: Failed to fetch" is reported when I use a browser to access a bucket or an object?

Cause

- The network connection is abnormal.
- The ad blocker installed in your browser filters the name of the bucket or object because the name of the bucket or object contains characters ad such as adtest or aadb.

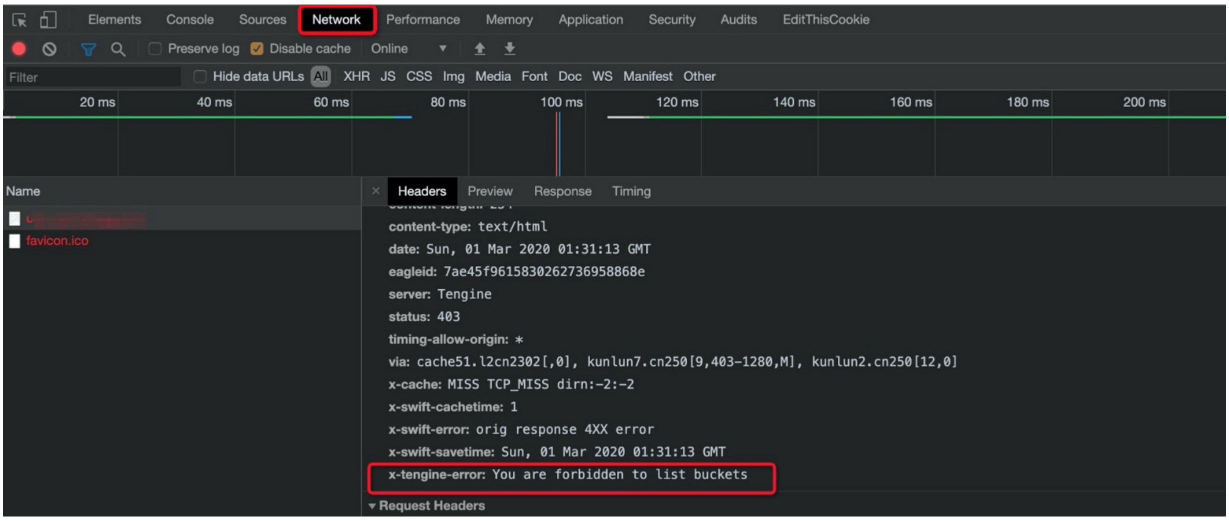
Solution

- Network exception: Check your network and try again after the exception is resolved.
- Ad blocker:
  - Disable the ad blocker of your browser, or add the Object Storage Service (OSS) domain name to the whitelist.
  - Do not include the characters ad in the name of your buckets or objects.

6.6.18. What do I do if "You are forbidden to list buckets" is returned when I access the accelerated domain name that is mapped to a bucket?

This topic describes the cause of the "You are forbidden to list buckets" error returned when you access the accelerated domain name that is mapped to a bucket and the solution to the error.

Problem: When you access the accelerated domain name that is mapped to a private bucket for which CDN back-to-origin is enabled, the `You are forbidden to list buckets` error is returned.



[redacted]
favicon.ico

```
content-type: text/html
date: Sun, 01 Mar 2020 01:31:13 GMT
eagleid: 7ae45f9615830262736958868e
server: Tengine
status: 403
timing-allow-origin: *
via: cache51.l2cn2302[,0], kunlun7.cn250[9,403-1280,M], kunlun2.cn250[12,0]
x-cache: MISS TCP_MISS dirn:-2:-2
x-swift-cachetime: 1
x-swift-error: orig response 4XX error
x-swift-savetime: Sun, 01 Mar 2020 01:31:13 GMT
x-tengine-error: You are forbidden to list buckets
```

Cause: AGetBucket (ListObejcts) request is sent to CDN. By default, this request is denied by CDN.

Solution: Rewrite the URL of the accelerated domain name in the CDN console to redirect the URL to a file in the path specified by the URL. For example, if the URL of the accelerated domain name that is mapped to your bucket is `example.aliyundoc.com`, rewrite the URL to `example.aliyundoc.com/index.html`. For more information about how to rewrite URLs, see Create a URI rewrite rule.

6.6.19. What do I do if my data is lost?

OSS is a distributed storage service that ensures data durability based on automatic backup for redundancy. To prevent data loss, OSS keeps data secure and intact.

However, data may be deleted in the following situations:

- Lifecycle rules

  If you have configured lifecycle rules to automatically delete objects, the system deletes data based on the lifecycle configurations. We recommend that you configure lifecycle rules based on actual requirements. For more information, see Lifecycle rules based on the last modified time.

- Bucket ACL set to public read/write
  - Bucket ACL set to public read/write: After you set the ACL of a bucket to public read/write, all users can read and write the objects in the bucket. We recommend that you do not set the bucket ACL to public read/write unless necessary. For more information, see Configure the ACL of a bucket.
  - Bucket policy that allows all users to write and read objects in a bucket: After you set the bucket ACL to public read, all users can read and write the objects in the bucket. We recommend that you do not set the bucket ACL to public read unless necessary. For more information, see Configure bucket policies to authorize other users to access OSS resources.

- Leak of accounts that have the management permissions on buckets: After an account and password that have the management permissions on a bucket are leaked, users who have the account can perform operations on the objects in your bucket. We recommend that you use RAM users to access resources during routine management, and grant fine-grained permissions to the RAM users. If you find that the account is leaked, change the password of the RAM user and disable the AccessKey pair to minimize impacts. For more information, see Overview of RAM users.
- Accidental deletes by administrators: After objects in OSS are deleted, the deleted objects cannot be retrieved. To prevent data from being accidentally overwritten and deleted, we recommend that you use the following features:
  - CRR: Back up your bucket data to another bucket. Objects in the source bucket is deleted. You can find the objects in the backup bucket. For more information, see Cross-region replication.
  - Scheduled backup: After scheduled backup is enabled, OSS backs up your data to Hybrid Backup Recovery (HBR) on a regular basis. This way, objects can be recovered when they are lost. For more information, see Configure scheduled backup.
  - Versioning: After you enable versioning, overwritten or deleted objects are saved as previous versions and can be recovered if necessary. For more information, see Overview.
  - Retention policy: You can enable retention policy for important data. Data within the retention period cannot be overwritten or deleted. For more information, see Retention policy.

7.Data security

7.1. Access and control

7.1.1. Overview

By default, the access control list (ACL) of Object Storage Service (OSS) resources, including buckets and objects, is set to private to ensure data security. Only the owners of the resources and authorized users can access these resources. OSS allows you to configure a variety of policies to grant third-party users specific permissions to access or use your OSS resources.

The following table describes the access control policies that you can configure for objects stored in buckets.

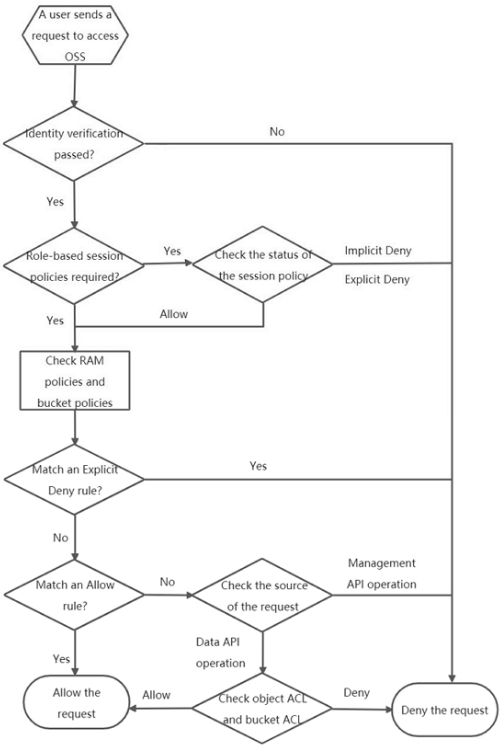
Policy	Description	Scenario
RAM Policy	Resource Access Management (RAM) is a service provided by Alibaba Cloud to manage access permissions on resources. RAM policies are configured based on users. You can configure RAM policies to manage your users, such as employees, systems, or applications, and control the permissions of the users on your resources. For example, you can configure a RAM policy to allow your users to only read one bucket.	• Grant the same permissions to the RAM users of the same Alibaba Cloud account. • Configure the same permissions required to access all OSS resources or multiple buckets. • Configure the permissions required to perform specific operations. For example, you can configure a RAM policy to specify the permission required to list all buckets that belong to the same Alibaba Cloud account. • Limit the permissions of temporary access credentials used to access OSS.
Bucket Policy	Bucket policies are configured based on resources. Compared with RAM policies, bucket policies can be easily configured on the graphical interface of the console. In addition, the owner of a bucket can configure bucket policies for the bucket without RAM permissions. You can configure bucket policies to grant permissions to the RAM users of other Alibaba Cloud accounts or anonymous users who access OSS by using the specified IP addresses.	• Grant different permissions to the RAM users of the same Alibaba Cloud account. • Grant permissions to the RAM users of other Alibaba Cloud accounts or anonymous users.
Object ACL	You can configure the ACL of a bucket when you create the bucket or modify the ACL of a created bucket. Only the owner of a bucket can configure or modify the ACL of the bucket. You can set the ACL of a bucket to one of the following values: *public-read-write*, *public-read*, and *private*.	Configure the same access permission for all objects in a bucket.
Object ACL	You can also configure the ACL of each object stored in OSS. You can configure the ACL of an object when you upload the object or modify the ACL of an uploaded object. You can set the ACL of an object to one of the following values: *Inherited from bucket*, *public-read-write*, *public-read*, and *private*.	Configure the access permission of a single object. For example, you configure RAM policies or bucket policies for a bucket to set the ACL of all objects in the bucket or objects whose names contain the specified prefix to private. In this case, if you want that an object in the bucket can be accessed by all anonymous users from the Internet, you can set the ACL of the object to *public-read*.

7.1.2. Authentication

By default, the access control list (ACL) of OSS resources including buckets and objects is set to private to ensure data security. Only the owners of the resources and authorized users can access these buckets and objects. OSS allows you to use a variety of policies to grant other users specific permissions to access or use your OSS resources. A user can access authorized OSS resources only after the user is authenticated by OSS based on all policies.

Overview

When OSS receives a request, OSS determines whether to allow or deny the request based on authentication. The authentication is performed based on identity verification, role-based session policies, Resource Access Management (RAM) policies, bucket policies, object ACLs, and bucket ACLs.


A user sends a request to access OSS
Identity verification passed?
No
Yes
Role-based session policies required?
Yes
Check the status of the session policy
Implicit Deny
Explicit Deny
Yes
Allow
Check RAM policies and bucket policies
Match an Explicit Deny rule?
Yes
No
Match an Allow rule?
No
Check the source of the request
Management API operation
Yes
Data API operation
Allow the request
Allow
Check object ACL and bucket ACL
Deny
Deny the request

In the preceding figure, the following states are used to indicate the authentication result of a request:

- Allow: The request matches an Allow rule specified in a policy. In this case, OSS allows the request.
- Explicit Deny: The request matches a Deny rule specified in a policy. In this case, OSS explicitly denies the request.
- Implicit Deny: The request does not match the Allow or Deny rules specified in a policy or the policy that OSS uses to authenticate the request does not exist. In this case, OSS implicitly denies the request.

Process

1. OSS checks whether the request passes identity verification.

   After OSS receives a request, OSS compares the signature contained in the request with the signature calculated by the OSS server.

   - If the two signatures are inconsistent, the request is denied.
   - If the two signatures are consistent, OSS checks whether the request needs to be authenticated based on role-based session policies.

2. OSS checks whether the request needs to be authenticated based on role-based session policies.

   If the request needs to be authenticated based on role-based session policies, OSS checks whether the request matches the session policies.

   - If the matching result of the request is Explicit Deny or Implicit Deny, the request is denied.
   - If the matching result of the request is Allow, OSS authenticates the request based on RAM policies and bucket policies.

   If the request does not need to be authenticated based on role-based session policies, OSS proceeds to authenticate the request based on RAM policies and bucket policies.

3. OSS checks whether the request matches RAM policies and bucket policies.

   RAM policies are configured based on users. You can configure RAM policies to control the resources that can be accessed by users. When OSS authenticates a request based on RAM policies, OSS determines whether to allow or deny the request based on the account used to sent the request.

   - If the request is sent by using the AccessKey pair of an Alibaba Cloud account, OSS implicitly denies the request.
   - If the request is sent by using the AccessKey pair of a RAM user or an STS credential to access a bucket that does not belong to the RAM user or the Alibaba Cloud account, OSS implicitly denies the request.
   - OSS calls the authentication operation provided by RAM to authenticate requests. Authentication based on accounts and the resource groups of buckets is supported. After the authentication, OSS determines whether to allow, explicitly deny, or implicitly deny the request.

   Bucket policies are resource-based authorization policies. The owner of a bucket can configure bucket policies to authorize RAM users or other Alibaba Cloud accounts to access the bucket or specific resources in the bucket.

   - If no bucket policy is configured for the bucket, OSS implicitly denies the request.
   - If bucket policies are configured for the bucket, OSS checks whether the request matches the bucket policies and then determines whether to allow, explicitly deny, or implicitly deny the request.

4. OSS checks whether the request matches an Explicit Deny rule specified in all policies described in the preceding three steps.

   If the request matches an Explicit Deny rule, OSS denies the request. If the request does not match an Explicit Deny rule, OSS checks whether the request matches an Allow rule specified in all policies described in the preceding three steps.

   i. OSS checks whether the request matches an Allow rule specified in RAM policies and bucket policies.

      If the request matches an Allow rule, OSS allows the request. If the request does not match an Allow rule, OSS checks the source of the request.

ii. OSS checks the source of the request.

If the request is sent by using a management API operation, OSS denies the requests. If the request is sent by using a data API operation, OSS checks the ACLs of the object and bucket that the request is sent to access.

Management API operations include service-related operations, bucket-related operations, and LiveChannel-related operations. Service-related operations include GetService (Listbuckets). Bucket-related operations include PutBucket and GetBucketLifecycle. LiveChannel-related operations include PutLiveChannel and DeleteLiveChannel.

Data API operations include object-related operations, such as PutObject and GetObject.

5. OSS authenticates the request based on the ACLs of the object and bucket that the request is sent to access.

   OSS authenticates the request based on the object ACL, whether the user that sends the request is the bucket owner, and whether the request is a read request or a write request.

   - If the authentication result is Allow, OSS allows the request.
   - If the authentication result is Deny, OSS denies the request.

   If the ACL of the object is inherited from the bucket, OSS authenticates the request based on the ACL of the bucket in which the object is stored.

   OSS authenticates the request based on the bucket ACL, and whether the user that sends the request is the bucket owner.

   - If the authentication result is Allow, OSS allows the request.
   - If the authentication result is Deny, OSS denies the request.

7.1.3. Configure the ACL of a bucket

Access control list (ACL) can be used to define the access permissions of users or user groups on data stored in Object Storage Service (OSS). After a request is sent to access data stored in OSS, OSS checks the ACL of the data and verifies whether the requester has required permissions. You can configure the ACL of a bucket when you create the bucket. You can also modify the ACL of an existing bucket based on your requirements. Only the owner of a bucket can configure or modify the ACL of the bucket.

Usage notes

- By default, if you do not specify the ACL of an object when you upload the object to a bucket, the ACL of the object inherits the ACL of the bucket.
- If you modify the ACL of a bucket, the ACLs of all objects that inherit the bucket ACL change accordingly.

Types of ACLs

The following table describes the three types of bucket ACLs.

ACL	Operation
public-read-write	Anyone, including anonymous users, can perform read and write operations on the objects in the bucket. **Warning** All users on the Internet can access objects in the bucket and write data to the bucket. This may result in unexpected access to the data in your bucket and unexpectedly high costs. If a user uploads prohibited data or information, your legitimate interests and rights may be infringed. Therefore, we recommend that you do not set your bucket ACL to public read/write except in special cases.
public-read	Only the owner of the bucket can write data to objects in the bucket. Other users, including anonymous users, can only read objects in the bucket. **Warning** All users on the Internet can access objects in the bucket. This may result in unexpected access to the data in your bucket and unexpectedly high costs. Exercise caution when you set your bucket ACL to this value.
private	Only the bucket owner can perform read and write operations on objects in the bucket. Other users cannot access the objects in the bucket. This is the default value.

Use the OSS console

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket that you want to manage.
3. In the left-side navigation pane, choose **Access Control > Access Control List (ACL)**.
4. In the **Access Control List (ACL)** section, click **Configure**. Then, modify the ACL of the bucket based on your requirements.
5. Click **Save**.

Use ossbrowser

ossbrowser supports the same operations related to buckets as the OSS console. You can follow the on-screen instructions in ossbrowser to modify the ACL of a bucket. For more information about how to use ossbrowser, see Use ossbrowser.

Use OSS SDKs

The following code provides examples on how to modify the ACL of a bucket by using OSS SDKs for common programming languages. For more information about how to modify the ACL of a bucket by using OSS SDKs for other programming languages, see Overview.

Python

```
import com.aliyun.oss.ClientException;
import com.aliyun.oss.OSS;
import com.aliyun.oss.OSSClientBuilder;
import com.aliyun.oss.OSSException;
import com.aliyun.oss.model.CannedAccessControlList;
public class Demo {
    public static void main(String[] args) throws Exception {
        // Set yourEndpoint to the endpoint of the region in which the bucket is located. For example, if the bucket is located in the China (Hangzhou) region, set yourEndpoint to https://oss-cn-hangzhou.aliyuncs.com.
        String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
        // Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API operations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
        String accessKeyId = "yourAccessKeyId";
        String accessKeySecret = "yourAccessKeySecret";
        // Specify the name of the bucket. Example: examplebucket.
        String bucketName = "examplebucket";
        // Create an OSSClient instance.
        OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
        try {
            // Configure the ACL of the bucket. For example, you can set the ACL of the examplebucket bucket to private.
            ossClient.setBucketAcl(bucketName, CannedAccessControlList.Private);
        } catch (OSSException oe) {
            System.out.println("Caught an OSSException, which means your request made it to OSS, "
                    + "but was rejected with an error response for some reason.");
            System.out.println("Error Message:" + oe.getErrorMessage());
            System.out.println("Error Code:" + oe.getErrorCode());
            System.out.println("Request ID:" + oe.getRequestId());
            System.out.println("Host ID:" + oe.getHostId());
        } catch (ClientException ce) {
            System.out.println("Caught an ClientException, which means the client encountered "
                    + "a serious internal problem while trying to communicate with OSS, "
                    + "such as not being able to access the network.");
            System.out.println("Error Message:" + ce.getMessage());
        } finally {
            if (ossClient != null) {
                ossClient.shutdown();
            }
        }
    }
}
```

Use ossutil

For more information about how to use ossutil to configure or modify the ACL of a bucket, see set-acl (configure or modify object or bucket ACLs).

Use the RESTful API

If your program requires more custom options to configure the ACL of a bucket, you can call RESTful API operations. In this case, you need to manually write code to calculate the signature. For more information, see PutBucketAcl.

References

In addition to bucket ACLs, OSS provides object ACLs, bucket policies, and RAM policies for you to control access to your buckets and objects in OSS. For more information, see Overview.

7.1.4. Object ACL

Access control lists (ACLs) can be used to define the access permissions of users or user groups on data stored in Object Storage Service (OSS). After a request is sent to access data stored in OSS, OSS checks the ACL of the data and verifies whether the requester has required permissions. You can configure the ACL of an object when you upload the object or modify the ACL of an uploaded object.

Usage notes

- If you do not set the object ACL, the object ACL is default. In that case, the ACL of the object is the same as that of the bucket in which the object is stored.
- If you set the object ACL to a value that is different from the bucket ACL, the object ACL takes precedence. For example, if the ACL of an object is set to public read, all authenticated and anonymous users can read the object regardless of the bucket ACL.

ACL types

The following table describes object ACL types.

ACL	Description
public-read-write	Public read/write: All users, including anonymous users, can perform read and write operations on objects in the bucket. **Warning** When you set the object ACL to this value, all users can access the object over the Internet and write data to the object. This may result in unexpected access to the data in your bucket and unexpectedly high fees. If a user uploads prohibited data or information, your legitimate interests and rights may be infringed. Therefore, we recommend that you do not set the object ACL to public read/write except in special cases.

ACL	Description
public-read	Only the bucket owner can write data to the object. Other users, including anonymous users, can only read the object. **Warning** All users can access the object over the Internet. This may result in unexpected access to the data in your bucket and unexpectedly high fees. Exercise caution when you set the object ACL to public read.
private	Private: Only the bucket owner is allowed to perform read and write permissions on the object. Other users cannot access the object. **Note** You can configure and send the object URL to share your private objects with your partners. For more information, see Add signatures to URLs.
default	Default value: The ACL of the object is the same as that of the bucket in which the object is stored.

Use the OSS console

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to modify the ACL.
3. In the left-side navigation pane, click **Files**.
4. Select the object for which you want to modify the ACL. Click the name of the object. In the **View Details** panel, Click **Set ACL**.

   Alternatively, move the pointer over **More** in the Actions column of the object and choose **Set ACL** from the shortcut menu.
5. In the **Set ACL** panel, modify the ACL of the object.
6. Click **OK**.

Use ossbrowser

ossbrowser supports the same object-wide operations as the OSS console. You can follow the on-screen instructions in ossbrowser to modify the ACL of an object. For more information about how to use ossbrowser, see Use ossbrowser.

Use OSS SDKs

The following code provides examples on how to modify the ACL of an object by using OSS SDKs for common programming languages. For more information about how to modify the ACL of an object by using OSS SDKs for other programming languages, see Overview.

[illegible]

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to log on to OSS, because the account has permissions on all AP
I operations. We recommend that you use a RAM user to call API operations or perform routine operations and maintenance. To create a RAM user, log on
to https://ram.console.aliyun.com.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Configure the ACL of the specified object to public read.
ossClient.setObjectAcl("<yourBucketName>", "<yourObjectName>", CannedAccessControlList.PublicRead);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Use ossutil

For more information about how to use ossutil to configure or modify the ACL of a bucket, see Configure or modify the ACLs of objects.

Use the RESTful API

If your program requires more custom options to configure the ACL of an object, you can call RESTful API operations. In this case, you must manually write code to calculate the signature. For more information, see PutObjectACL.

References

In addition to object ACLs, OSS provides bucket ACLs, bucket policies, and RAM policies for you to control access to your buckets and objects in OSS. For more information, see Overview.

7.1.5. Bucket Policy

7.1.5.1. Overview

You can configure bucket policies to authorize other users to access your specific Object Storage Service (OSS) resources.

Scenarios

Bucket policies can be used for access authorization in the following scenarios:

- You need to grant permissions to another Alibaba Cloud account or anonymous users to access or manage all resources or part of resources in a bucket.
- You need to grant different permissions such as read-only, read/write, or complete management to the RAM users of the same Alibaba Cloud to access or manage resources in your bucket.

Implementation methods

The following table describes the methods that you can use to configure bucket policies in different scenarios.

Implementation method	Description
Console	- Configure bucket policies by using graphical interfaces. - Configure bucket policies by specifying policy syntax. - Tutorial: Implement data sharing across departments based on bucket policies - Authorize a RAM user under another Alibaba Cloud account by adding a bucket policy
ossutil	bucket-policy
SDKs for different programming languages	- OSS SDK for Java - OSS SDK for PHP - OSS SDK for Node.js - OSS SDK for Python - OSS SDK for .NET - OSS SDK for Go - OSS SDK for C++

7.1.5.2. Examples

You can configure bucket policies to authorize other users to access specified Object Storage Service (OSS) resources. For example, you can configure bucket policies to grant different permissions, such as read-only or read/write, to anonymous users, Resource Access Management (RAM) users of the same Alibaba Cloud account, or different Alibaba Cloud accounts.

Introduction

The following examples show the bucket policies configured by the bucket owner whose UID is `174649585760xxxx` to grant different permissions to RAM users, such as the RAM user whose UID is `27737962156157xxxx` . Compared with RAM policies, bucket policies contain the Principal field used to specify the users to which you want to grant permissions. The syntax of other fields in bucket policies, such as Action and Condition, is the same as that of these fields in RAM policies. For more information about how to configure the fields in a RAM policy, see Overview.

Usage notes

When you configure a bucket policy, the bucket policy applies to all users only except the bucket owner if Principal is set to an asterisk (*) and Condition is not included. For more information, see Example 3.

When you configure a bucket policy, the bucket policy applies to all users if Principal is set to an asterisk (*) and Condition is included. For more information, see Example 4.

Example 1: Grant the specified RAM users permissions to read and write a bucket

The following bucket policy can be configured to grant the RAM users whose UIDs are `27737962156157xxxx` and `20214760404935xxxx` respectively permissions to read and write a bucket named examplebucket:

```
{
    "Version": "1",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "oss:GetObject",
            "oss:PutObject",
            "oss:GetObjectAcl",
            "oss:PutObjectAcl",
            "oss:ListObjects",
            "oss:AbortMultipartUpload",
            "oss:ListParts",
            "oss:RestoreObject",
            "oss:GetVodPlaylist",
            "oss:PostVodPlaylist",
            "oss:PublishRtmpStream",
            "oss:ListObjectVersions",
            "oss:GetObjectVersion",
            "oss:GetObjectVersionAcl",
            "oss:RestoreObjectVersion"
        ],
        "Principal": [
            "27737962156157xxxx",
            "20214760404935xxxx"
        ],
        "Resource": [
            "acs:oss:*:174649585760xxxx:examplebucket/*"
        ]
      }, {
        "Effect": "Allow",
        "Action": [
            "oss:ListObjects",
            "oss:GetObject"
        ],
        "Principal": [
            "27737962156157xxxx",
            "20214760404935xxxx"
        ],
        "Resource": [
            "acs:oss:*:174649585760xxxx:examplebucket"
        ],
        "Condition": {
            "StringLike": {
                "oss:Prefix": [
                    "*"
                ]
            }
        }
      }
    ]
}
```

Example 2: Grant a RAM user permissions only to read the specified directory of a bucket

The following bucket policy can be configured to grant a RAM user whose UID is `20214760404935xxxx` permissions only to read the `hangzhou/2020` and `shanghai/2015` directories of a bucket named examplebucket:

```
{
     "Version": "1",
    "Statement": [
        {
            "Action": [
                "oss:GetObject",
                "oss:GetObjectAcl",
                "oss:GetObjectVersion",
                "oss:GetObjectVersionAcl"
            ],
            "Effect": "Allow",
            "Principal": [
                "20214760404935xxxx"
            ],
            "Resource": [
                "acs:oss:*:174649585760xxxx:examplebucket/hangzhou/2020/*",
                "acs:oss:*:174649585760xxxx:examplebucket/shanghai/2015/*"
            ]
        },
        {
            "Action": [
                "oss:ListObjects",
                "oss:ListObjectVersions"
            ],
            "Condition": {
                "StringLike": {
                    "oss:Prefix": [
                        "hangzhou/2020/*",
                        "shanghai/2015/*"
                    ]
                }
            },
            "Effect": "Allow",
            "Principal": [
                "20214760404935xxxx"
            ],
            "Resource": [
                "acs:oss:*:174649585760xxxx:examplebucket"
            ]
        }
    ]
}
```

Example 3: Grant anonymous users permissions only to list all objects in a bucket

The following bucket policy can be configured to grant anonymous users permissions only to list all objects in a bucket named examplebucket:

```
{
    "Version": "1",
    "Statement": [
    {
        "Action": [
            "oss:ListObjects",
            "oss:ListObjectVersions"
        ],
        "Effect": "Allow",
        "Principal": [
            "*"
        ],
        "Resource": [
            "acs:oss:*:174649585760xxxx:examplebucket"
        ]
    }
  ]
}
```

Example 4: Prevent users who use IP addresses that are not in the specified CIDR block from performing operations on a bucket

The following bucket policies can be configured to prevent anonymous users who use IP addresses that are not in the CIDR block `192.168.0.0/16` from managing a bucket named examplebucket:

```
{
    "Version": "1",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "oss:*",
            "Principal": [
                "*"
            ],
            "Resource": [
                "acs:oss:*:174649585760xxxx:examplebucket"
            ],
            "Condition":{
                "NotIpAddress": {
                    "acs:SourceIp": ["192.168.0.0/16"]
                }
            }
        }
    ]
}
```

7.1.5.3. Tutorial: Use bucket policies to restrict access to OSS over the Internet

You can configure bucket policies to restrict access to your Object Storage Service (OSS) resources over the Internet.

Scenario

Enterprise A creates a bucket named examplebucket in the China (Hangzhou) region. A large amount of internal data is stored in the *examplefolder* directory of examplebucket. Enterprise A does not want specific partners to access resources in the *examplefolder* directory over the Internet by using RAM users.

To meet the preceding requirements of Enterprise A, you can configure a bucket policy by using the policy syntax.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click **examplebucket**.
3. In the left-side navigation pane, click **Files**. On the page that appears, click **Authorize**.
4. On the **Syntax** tab, click **Edit** and enter the following policy:

```
{
    "Version": "1",
    "Statement": [{
        "Effect": "Deny",
        "Action": [
            "oss:RestoreObject",
            "oss:ListObjects",
            "oss:AbortMultipartUpload",
            "oss:PutObjectAcl",
            "oss:GetObjectAcl",
            "oss:ListParts",
            "oss:DeleteObject",
            "oss:PutObject",
            "oss:GetObject",
            "oss:GetVodPlaylist",
            "oss:PostVodPlaylist",
            "oss:PublishRtmpStream",
            "oss:ListObjectVersions",
            "oss:GetObjectVersion",
            "oss:GetObjectVersionAcl",
            "oss:RestoreObjectVersion"
        ],
        "Principal": [
            <! -- The following sample IDs of RAM users are for your reference. -->
            "26642223584287****",
            "27658173539067****",
            "24430533117653****"
        ],
        "Resource": [
            <! -- 137918634953**** is the user ID of the owner of the examplebucket bucket. -->
            "acs:oss:*:137918634953****:examplebucket/examplefolder/*"
        ],
        "Condition": {
            "StringNotEquals": {
                "acs:SourceVpc": [
                    "vpc-*"
                ]
            }
        }
    }, {
        "Effect": "Deny",
        "Action": [
            "oss:ListObjects",
            "oss:GetObject"
        ],
        "Principal": [
            "26642223584287****",
            "27658173539067****",
            "24430533117653****"
        ],
        "Resource": [
            "acs:oss:*:137918634953****:examplebucket"
        ],
        "Condition": {
            "StringLike": {
                "oss:Prefix": [
                    "examplefolder/*"
                ]
            },
            "StringNotEquals": {
                "acs:SourceVpc": [
                    "vpc-*"
                ]
            }
        }
    }]
}
```

5. Click **Save**.

References

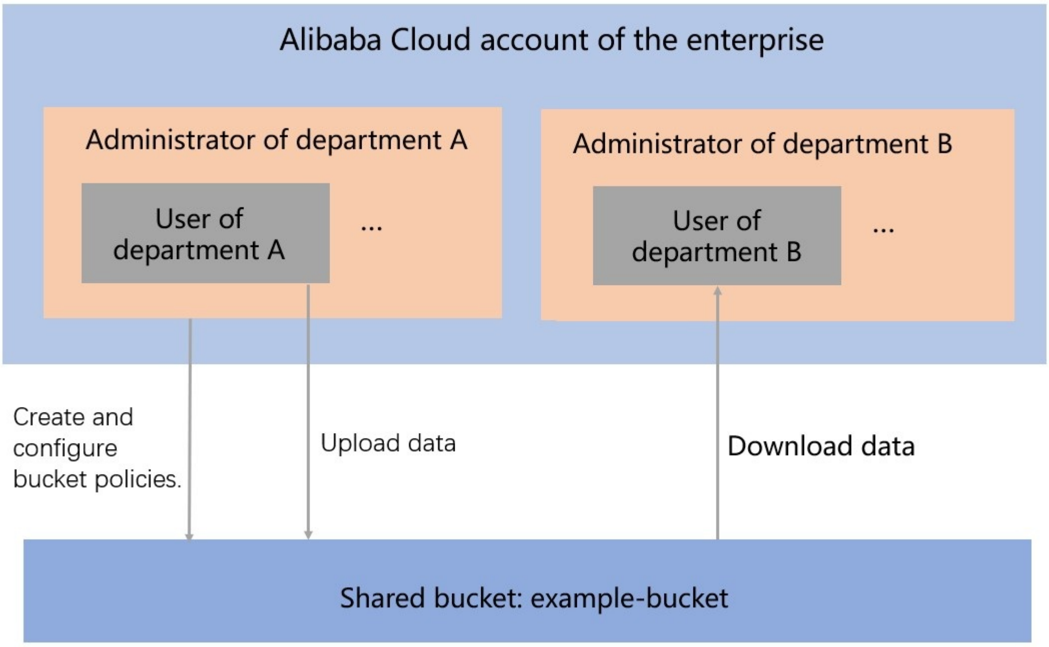
- Data must be shared across multiple departments or projects. You may want users from other departments to download the data that is shared by your department. However, you do not want the users to write or delete the shared data. In this case, you can implement data sharing across multiple departments based on bucket policies. For more information, see Tutorial: Implement data sharing across departments based on bucket policies.
- You can grant different permissions, such as read-only and read and write permissions, to anonymous users or RAM users in the same Alibaba Cloud account and across multiple Alibaba Cloud accounts to access or manage bucket resources. For more information, see Examples.

7.1.5.4. Tutorial: Implement data sharing across departments based on bucket policies

This topic describes how to implement data sharing across different departments or projects of an enterprise to allow that the data shared by a department can be only downloaded but cannot be written or deleted by users of other departments. This way, you can reduce the risks of accidental deletion and modification of the shared data.

Context

In this example, department A shares the data stored in a bucket named example-bucket with users of department B and allows these users to download the shared data. This example shows how to follow the principle of least privilege to perform access control on shared data. The following figure shows the relationship between the shared bucket and the administrators and users of department A and B.


Alibaba Cloud account of the enterprise
Administrator of department A
User of department A
...
Administrator of department B
User of department B
...
Create and configure bucket policies.
Upload data
Download data
Shared bucket: example-bucket

Procedure

In this example, the administrator of department A can configure bucket policies to allow users of department B to download but not write or delete the shared data. To achieve the goal, the following steps must be performed:

- Step 1: Create a bucket

  The administrator of department A creates a bucket named example-bucket to store shared data.
- Step 2: Grant permissions to upload shared data

  The administrator of department A configures a bucket policy for example-bucket to allow users of department A to upload shared data to the bucket.
- Step 3: Grant permissions to download but not write or delete shared data

  The administrator of department A configures a bucket policy for example-bucket to allow users of department B to download but not write or delete shared data.
- Step 4: Upload shared data

  Users of department A upload shared data to example-bucket.
- Step 5: Verify permissions

  Verify the permissions of the users of department B to ensure they can only download but cannot write or delete shared data.

Prerequisites

- RAM users for the administrators and users of department A and B are created by the Alibaba Cloud account of the enterprise.

  For more information about how to create RAM users, see Create a RAM user.
- The UIDs of the RAM users are obtained.

  For more information about how to view the basic information about a RAM user such as the UID, see View the basic information about a RAM user.
- Appropriate permissions are granted to the RAM users.

  In this example, the administrator of department A needs to create buckets and configure bucket policies. Therefore, the user group of the RAM users for administrators must have the AliyunOSSFullAccess permission. For more information about how to grant permissions to RAM users, see Grant permissions to a RAM user.

Step 1: Create a bucket

Perform the following steps to create a bucket as the administrator of department A:

1. Use the RAM user of the administrator of department A to log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the page that appears, click **Create Bucket**.
3. In the **Create Bucket** dialog box that appears, configure parameters for the bucket.

   In this example, set the bucket name to example-bucket. For more information about how to configure parameters to create a bucket, see Create buckets.
4. Click **OK**.

Step 2: Grant permissions to upload shared data

Perform the following steps to grant users of department A permissions to upload shared data to example-bucket as the administrator of department A:

1. Click example-bucket created in Step 1.
2. You can also choose **Access Control** > **Bucket Policy**. In the Bucket Policy section, click **Configure**.

3. In the **Authorize** dialog box that appears, click **Authorize**.
4. In the **Authorize** dialog box that appears, configure parameters for the bucket policy.

Parameter	Description
Applied To	Select **Whole Bucket**, which indicates that the policy applies to the whole bucket.
Accounts	Select **RAM Users**. You can select the RAM users to which you can grant permissions to upload shared data from the drop-down list. You can also enter a keyword in the search box to search for specific RAM users. Fuzzy matching is supported.
Authorized Operation	Select **Read/Write**. This option indicates that authorized users can perform read and write operations on the specified resources.

5. Click **OK**.

   A bucket policy is created to allow users of department A to upload shared data.

Step 3: Grant permissions to download but not write or delete shared data

Perform the following steps to grant users of department B permissions to download shared data from example-bucket as the administrator of department A:

1. Click example-bucket created in Step 1.
2. You can also choose **Access Control > Bucket Policy**. In the Bucket Policy section, click **Configure**.
3. In the **Authorize** dialog box that appears, click **Authorize**.
4. In the **Authorize** dialog box that appears, configure parameters for the bucket policy.

Parameter	Description
Applied To	Select **Whole Bucket**, which indicates that the authorization policy applies to the whole bucket.
Accounts	Select **Other Accounts**. Enter the UIDs of the RAM users to which you want to grant permissions to download shared data.
Authorized Operation	Select **Read Only**. This option indicates that authorized users can only view, list, and download but cannot write or delete the data stored in example-bucket.

5. Click **OK**.

   A bucket policy is created to allow users of department B to download but not write or delete shared data.

Step 4: Upload shared data

Perform the following steps to upload data to example-bucket as a user of department A:

1. Use the RAM user of a user of department A to log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the page that appears, click example-bucket.
3. Choose **Files > Upload**.
4. In the **Upload** dialog box that appears, configure parameters to upload shared data.

   Select Current for Upload To. For more information about how to configure the ACL and upload method of the object to upload, see Upload objects.
5. After the upload is complete, close the **Upload Tasks** dialog box.

   The shared data is uploaded to example-bucket.

Step 5: Verify permissions

Perform the following steps in the OSS console to verify that users of department B can only download but cannot write or delete shared data:

1. Use the RAM user of a user of department B to log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the page that appears, click example-bucket.
3. On the page that appears, click the **Files**.
4. Verify permissions.
   i. Verify the download permissions of users of department B on shared data.

      In the Actions column corresponding to any object, choose **More > Download**.
      - If the download fails, the download permissions are not correctly configured. Check whether the bucket policy is correctly configured.
      - If the download is successful, the download permissions are correctly configured.

   ii. Verify the upload permissions of users of department B on shared data.

      Follow Step 4 to upload data to example-bucket.
      - If the upload fails, the upload permissions are correctly configured.
      - If the upload is successful, the upload permissions are not correctly configured. Check whether the bucket policy is correctly configured.

   iii. Verify the deletion permissions of users of department B on shared data.

      In the Actions column corresponding to any object in example-bucket, choose **More > Completely Delete**.
      - If the object cannot be deleted, the deletion permissions are correctly configured.
      - If the object is deleted, the deletion permissions are not correctly configured. Check whether the bucket policy is correctly configured.

7.1.5.5. Tutorial: Authorize a RAM user under another Alibaba Cloud account by adding a bucket policy

By default, access to OSS resources is restricted to the owner. To authorize another user to access your OSS resources, you can grant permissions for the user to access your bucket by adding a bucket policy.

Context

Example: Company A wants to authorize Company B to access their OSS resources. However, Company A does not want to provide Company B with a RAM user. In this case, Company A can allow Company B to access their bucket by adding a bucket policy. After Company A adds a bucket policy that authorizes Company B to access their bucket, Company B can access the OSS bucket owned by Company A by adding the path of the bucket in the OSS console.

Add the bucket policy

- Use the Alibaba Cloud account of Company B to perform the following steps:
  i. Log on to the RAM console to create a RAM user (herein referred to as RAM User B).

     For more information about how to add a resource group, see Create a RAM user.

  ii. In the left-side navigation pane, click **Users**.
  iii. Click the username of the created RAM user to view and record the UID of the RAM user.
- Use the Alibaba Cloud account of Company A to perform the following steps:
  i. Log on to the OSS console.
  ii. Click **Buckets**, and then click the name of the target bucket.
  iii. Choose **Files > Authorize > Authorize**.

     > ? **Note** You can also choose **Access Control > Bucket Policy**. In the Bucket Policy section, click **Configure**. The **Authorize** dialog box appears.

  iv. In the **Authorize** dialog box that appears, configure the required parameters. Set **Accounts** to **Other Accounts**, and enter the UID of RAM User B. For more information about other parameters, see Use bucket policies to authorize other users to access OSS resources.
  v. Click **OK**.

Log on to the OSS console as RAM User B and add the access path

After the bucket policy is added, you must log on to the OSS console as RAM User B and add the access path of the bucket of Company A. To add the access path, perform the following steps:

1. Log on to the OSS console as RAM User B through the RAM User logon link.
2. Go to the OSS console.
3. In the left-side navigation pane, click the + icon next to **My OSS Paths**. In the Add Authorized OSS Path dialog box that appears, add the access path of the authorized bucket.
   - **Region**: Select the region of the bucket of Company A from the drop-down list.
   - **Bucket**: Enter the name of the bucket of Company A.
   - **File Path**: Enter the path that Company A authorizes Company B to access. Example: To allow Company B to access only the *test* folder in the *abc* bucket, enter *abc/test* for File Path.

You can also create an AccessKey pair for the RAM user, and log on to ossutil or ossbrowser with the AccessKey pair to access the authorized bucket.

7.1.6. RAM Policy

7.1.6.1. Overview

Resource Access Management (RAM) policies are authorization policies configured based on users. You can configure RAM policies to manage user access to your resources stored in Object Storage Service (OSS).

Background information

- Syntax and structure of RAM policies

  A RAM policy contains a version number and a statement. Each statement contains the following elements: Effect, Action, Resource, and Condition. The Condition element is optional. For more information about the syntax and structure of RAM policies, see Policy structure and syntax.

  You can use the Version, Statement, and Effect elements in RAM policies for OSS in the same manner as in policies for RAM. For more information about how to use the Action, Resource, and Condition elements in RAM policies for OSS, see the following sections:

  - Action element in RAM policies for OSS
  - Resource element in RAM policies for OSS
  - Condition element in RAM policies for OSS
- Common RAM policies for OSS
  - AliyunOSSFullAccess: grants a RAM user the permissions to manage OSS resources.
  - AliyunOSSReadOnlyAccess: grants a RAM user the read-only permissions on OSS resources.
- Access control methods supported by OSS

  For more information about access control methods supported by OSS, see Overview.

Action element in RAM policies for OSS

RAM policies for OSS support service-related actions, bucket-related actions, and object-related actions.

- Service-related actions

API operation	Action	Description
GetService (ListBuckets)	oss:ListBuckets	Lists all buckets owned by the requester.

- Bucket-related actions

API operation	Action	Description
PutBucket	oss:PutBucket	Creates a bucket.
GetBucket (ListObjects)	oss:ListObjects	Lists all objects in a bucket.
GetBucketInfo	oss:GetBucketInfo	Queries the information about a bucket.
GetBucketLocation	oss:GetBucketLocation	Queries the location information about a bucket.
PutBucketVersioning	oss:PutBucketVersioning	Specifies the versioning status of a bucket.
GetBucketVersioning	oss:GetBucketVersioning	Queries the versioning status of a bucket.
GetBucketVersions(ListObjectVersions)	oss:ListObjectVersions	Lists the versions of all objects including delete markers in a bucket.
PutBucketAcl	oss:PutBucketAcl	Configures or modifies the access control list (ACL) of a bucket.
GetBucketAcl	oss:GetBucketAcl	Configures or modifies the ACL of a bucket.
DeleteBucket	oss:DeleteBucket	Deletes a bucket.
PutBucketLogging	oss:PutBucketLogging	Enables logging for a bucket.
GetBucketLogging	oss:GetBucketLogging	Queries the logging configurations of a bucket.
DeleteBucketLogging	oss:DeleteBucketLogging	Disables logging for a bucket.
PutBucketWebsite	oss:PutBucketWebsite	Enables static website hosting for a bucket and configures redirection rules for the bucket.
GetBucketWebsite	oss:GetBucketWebsite	Queries the static website hosting status of a bucket and the redirection rules configured for the bucket.
DeleteBucketWebsite	oss:DeleteBucketWebsite	Disables static website hosting for a bucket and deletes the redirection rules configured for the bucket.
PutBucketReferer	oss:PutBucketReferer	Configures hotlink protection for a bucket.
GetBucketReferer	oss:GetBucketReferer	Queries the hotlink protection configurations of a bucket.
PutBucketLifecycle	oss:PutBucketLifecycle	Configures lifecycle rules for a bucket.
GetBucketLifecycle	oss:GetBucketLifecycle	Queries the lifecycle rules configured for a bucket.
DeleteBucketLifecycle	oss:DeleteBucketLifecycle	Deletes the lifecycle rules configured for a bucket.
ListMultipartUploads	oss:ListMultipartUploads	Lists all ongoing multipart upload tasks, which include tasks that have been initiated but are not completed or canceled.
PutBucketCors	oss:PutBucketCors	Configures cross-origin resource sharing (CORS) rules for a bucket.
GetBucketCors	oss:GetBucketCors	Queries the CORS rules configured for a bucket.
DeleteBucketCors	oss:DeleteBucketCors	Disables the CORS feature and deletes all CORS rules configured for a bucket.
PutBucketPolicy	oss:PutBucketPolicy	Configures policies for a bucket.
GetBucketPolicy	oss:GetBucketPolicy	Queries the policies configured for a bucket.
DeleteBucketPolicy	oss:DeleteBucketPolicy	Deletes the policies configured for a bucket.
PutBucketTags	oss:PutBucketTagging	Adds tags to or modifies the tags of a bucket.
GetBucketTags	oss:GetBucketTagging	Queries the tags of a bucket.
DeleteBucketTags	oss:DeleteBucketTagging	Deletes the tags of a bucket.
PutBucketEncryption	oss:PutBucketEncryption	Configures encryption rules for a bucket.
GetBucketEncryption	oss:GetBucketEncryption	Queries the encryption rules of a bucket.
DeleteBucketEncryption	oss:DeleteBucketEncryption	Deletes the encryption rules configured for a bucket.
PutBucketRequestPayment	oss:PutBucketRequestPayment	Configures the pay-by-requester mode for a bucket.
GetBucketRequestPayment	oss:GetBucketRequestPayment	Queries the pay-by-requester configurations of a bucket.
PutBucketReplication	oss:PutBucketReplication	Configures data replication rules for a bucket.
GetBucketReplication	oss:GetBucketReplication	Queries the data replication rules configured for a bucket.

API operation	Action	Description
DeleteBucketReplication	oss:DeleteBucketReplication	Stops the data replication tasks of a bucket and deletes the data replication configurations of the bucket.
GetBucketReplicationLocation	oss:GetBucketReplicationLocation	Queries the regions in which the destination bucket can be located.
GetBucketReplicationProgress	oss:GetBucketReplicationProgress	Queries the data replication progress of a bucket.
PutBucketInventory	oss:PutBucketInventory	Configures bucket inventories for a bucket.
GetBucketInventory	oss:GetBucketInventory	Queries the specified inventories configured for a bucket.
ListBucketInventory	oss:GetBucketInventory	Queries all the inventories configured for a bucket.
DeleteBucketInventory	oss:DeleteBucketInventory	Deletes a specified inventory configured for a bucket.
PutStyle	oss:PutStyle	Configures image styles.
GetStyle	oss:GetStyle	Queries image styles.
ListStyle	oss:ListStyle	Lists image styles.
DeleteStyle	oss:DeleteStyle	Deletes image styles.

- Object-related actions

API operation	Action	Description
PutObject	oss:PutObject	Uploads an object.
PostObject	oss:PutObject	Uploads an object to a specified bucket by using HTML form upload.
AppendObject	oss:PutObject	Uploads an object by appending the content of the object to an existing object.
InitiateMultipartUpload	oss:PutObject	Initiates a multipart upload task.
UploadPart	oss:PutObject	Uploads an object by part based on the specified object name and the upload ID.
CompleteMultipartUpload	oss:PutObject	Completes a multipart upload task.
AbortMultipartUpload	oss:AbortMultipartUpload	Cancels a multipart upload task and deletes uploaded parts.
PutSymlink	oss:PutObject	Creates a symbolic link for an object.
GetObject	oss:GetObject	Queries an object.
HeadObject	oss:GetObject	Queries the metadata of an object.
GetObjectMeta	oss:GetObject	Queries the metadata of an object, including the ETag, the object size, and the last modified time.
SelectObject	oss:GetObject	Executes SQL statements on an object. After the SQL statements are executed, execution results are returned.
GetSymlink	oss:GetObject	Queries the symbolic link of an object.
DeleteObject	oss:DeleteObject	Deletes an object.
DeleteMultipleObjects	oss:DeleteObject	Deletes multiple objects from a bucket.
CopyObject	oss:GetObject,oss:PutObject	Copies objects within the same bucket or across buckets in the same region.
UploadPartCopy	oss:GetObject,oss:PutObject	Copies data from an existing object to upload a part by adding the x-oss-copy-source request header to a UploadPart request to call UploadPartCopy.
ListParts	oss:ListParts	Lists all parts that are uploaded by using a specified upload ID.
PutObjectACL	oss:PutObjectAcl	Modifies the ACL of an object in a bucket.
GetObjectACL	oss:GetObjectAcl	Queries the ACL of an object in a bucket.
RestoreObject	oss:RestoreObject	Restores an object of the Archive or Cold Archive storage class.
PutObjectTagging	oss:PutObjectTagging	Adds tags to or modifies the tags of an object.
GetObjectTagging	oss:GetObjectTagging	Queries the tags of an object.
DeleteObjectTagging	oss:DeleteObjectTagging	Deletes the tags of an object.
GetObject (Specify the version ID of an object in the request)	oss:GetObjectVersion	Downloads a specified version of an object.
PutObjectACL (Specify the version ID of an object in the request)	oss:PutObjectAcl	Modifies the ACL of a specified version of an object.

API operation	Action	Description
GetObjectACL (Specify the version ID of an object in the request)	oss:GetObjectVersionAcl	Queries the ACL of a specified version of an object in a bucket.
RestoreObject (Specify the version ID of an object in the request)	oss:RestoreObjectVersion	Restores a specified version of an object of the Archive or Cold Archive storage class.
DeleteObject (Specify the version ID of an object in the request)	oss:DeleteObjectVersion	Deletes a specified version of an object.
PutObjectTagging (Specify the version ID of an object in the request)	oss:PutObjectVersionTagging	Adds tags to or modifies the tags of a specified version of an object.
GetObjectTagging (Specify the version ID of an object in the request)	oss:GetObjectVersionTagging	Queries the tags of a specified version of an object.
DeleteObjectTagging (Specify the version ID of an object in the request)	oss:DeleteObjectVersionTagging	Deletes the tags of a specified version of an object.
PutLiveChannel	oss:PutLiveChannel	Creates a LiveChannel before you upload audio and video data by using the RTMP protocol.
ListLiveChannel	oss:ListLiveChannel	Lists specified LiveChannels.
DeleteLiveChannel	oss:DeleteLiveChannel	Deletes a specified LiveChannel.
PutLiveChannelStatus	oss:PutLiveChannelStatus	Switches the status between enabled and disabled.
GetLiveChannelInfo	oss:GetLiveChannel	Queries the configurations of a specified LiveChannel.
GetLiveChannelStat	oss:GetLiveChannelStat	Queries the ingestion status of a specified LiveChannel.
GetLiveChannelHistory	oss:GetLiveChannelHistory	Queries the ingestion records of a specified LiveChannel.
PostVodPlaylist	oss:PostVodPlaylist	Generates a VOD playlist for a specified LiveChannel.
GetVodPlaylist	oss:GetVodPlaylist	Queries the playlist that is generated by the streams ingested to the specified LiveChannel within the specified time range.
ImgSaveAs	oss:PostProcessTask	Saves processed images to a specified bucket.

Resource element in RAM policies for OSS

In RAM policies for OSS, the Resource element indicates one or more specific resources. This element supports the asterisk (*) wildcard character. A RAM policy can contain multiple Resource elements.

The Resource element is specified in the following format: `acs:oss:{region}:{bucket_owner}:{bucket_name}/{object_name}` .

When you specify the Resource element in a RAM policy for a bucket, you do not need to add a forward slash (/) or `{object_name}` after `{bucket_name}` . In this case, you can specify the Resource element in the following format: `acs:oss:{region}:{bucket_owner}:{bucket_name}` . The region field can be set only to the asterisk (*) wildcard character.

Condition element in RAM policies for OSS

In RAM policies for OSS, the Condition element indicates the conditions for the RAM policies. The following table describes the conditions supported by OSS.

Condition	Description
acs:SourceIp	The CIDR block from which the requests originate. This condition supports the asterisk (*) wildcard character.
acs:UserAgent	The User-Agent header in the HTTP request. Type: string.
acs:CurrentTime	The time when the request arrives at the OSS server. Standard: ISO 8601.
acs:SecureTransport	The protocol of the request. If the protocol of the request is HTTP, set the value to HTTP. If the protocol of the request is HTTPS, set the value to HTTPS.
oss:Prefix	The prefix of the names of the objects that you want to list by calling the ListObjects operation.
oss:Delimiter	The character that is used to group the names of objects that you want to list by calling the ListObjects operation.
acs:AccessId	The AccessKey ID included in the request.
oss:BucketTag	The tag of the bucket. A single bucket tag can be used as a condition. To configure multiple bucket tags as multiple conditions, you must add `oss:BucketTag/` before each bucket tag.

Condition	Description
acs:MFAPresent	Specifies whether multi-factor authentication (MFA) is enabled. Valid values: • true: MFA is enabled. • false: MFA is disabled.
oss:ExistingObjectTag	Specifies that the requested object has tags. A single object tag can be used as a condition. To configure multiple object tags as multiple conditions, you must add `oss:ExistingObjectTag/` before each object tag. This condition applies to operations that are called to read objects, such as GetObject and HeadObject, and operations related to object tags, such as PutObjectTagging and GetObjectTagging.
oss:RequestObjectTag	The object tags included in the request. A single object tag can be used as a condition. To configure multiple object tags as multiple conditions, you must add `oss:RequestObjectTag/` before each object tag. This condition applies to operations that are called to write objects, such as PutObject and PostObject, and operations related to object tags, such as PutObjectTagging and GetObjectTagging.

Examples

You can use RAM policies to grant permissions to users in different scenarios. For more information, see Common examples of RAM policies.

7.1.6.2. Common examples of RAM policies

You can configure RAM policies to manage the permissions of users such as employees, systems, or applications and control the resources that can be accessed by users. For example, you can create a RAM policy to authorize users to list and read the objects stored in a specified bucket.

Attach a custom policy to a RAM user

1. Create a custom policy.

   You can refer to the examples described in this topic based on actual scenarios and create a custom RAM policy by using scripts. For more information about specific operations, see Create a custom policy.

   A RAM policy consists of the Version and Statement elements. A Statement contains the Effect, Action, Resource, and Condition fields, in which the Condition field is optional. For more information, see Overview.

   Notice In OSS, you can set Resource to an asterisk (*) to specify resources of a specific type. The format to specify the resources is `acs:oss:{region}:{bucket_owner}:{bucket_name}/{object_name}`. For example, if Resource is set to `acs:oss:*:*:mybucket/*`, all resources in mybucket are specified. If Resource is set to `acs:oss:*:*:mybucket/abc*.txt`, all .txt objects in mybucket that are prefixed with abc are specified.

2. Attach the custom policy to a RAM user.

   Attach the RAM policy created in Step 1 to a RAM user. For more information, see Grant permissions to a RAM user.

Example 1: Authorize a RAM user to completely control a bucket

The following RAM policy authorizes a RAM user to completely control a bucket named `mybucket`.

Warning We recommend that you do not authorize RAM users to completely control a bucket used by mobile apps because it is highly risky.

```
{
    "Version": "1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "oss:*",
            "Resource": [
                "acs:oss:*:*:mybucket",
                "acs:oss:*:*:mybucket/*"
            ]
        }
    ]
}
```

Example 2: Prohibit a RAM user from deleting multiple objects in a bucket

The following code provides an example on how to prohibit RAM users from deleting all .txt objects in `mybucket` that are prefixed with abc:

```
{
  "Version": "1",
  "Statement": [
        {
         "Effect": "Deny",
         "Action": [
           "oss:DeleteObject"
         ],
         "Resource": [
           "acs:oss:*:*:mybucket/abc*.txt"
         ]
     }
   ]
}
```

Example 3: Authorize a RAM user to list and read objects in a bucket

- Authorize a RAM user to list and read objects in a bucket by using OSS SDKs or ossutil

  The following RAM policy authorizes a RAM user to list and read objects in a bucket named `mybucket` by using OSS SDKs or ossutil:

```
{
    "Version": "1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "oss:ListObjects",
            "Resource": "acs:oss:*:*:mybucket"
        },
        {
            "Effect": "Allow",
            "Action": "oss:GetObject",
            "Resource": "acs:oss:*:*:mybucket/*"
        }
    ]
}
```

- Authorize a RAM user to list and read objects in a bucket in the OSS console

  The following code provides an example on how to authorize RAM users to list and read all resources in a bucket named `mybucket` in the OSS console:

```
{
    "Version": "1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                      "oss:ListBuckets",
                      "oss:GetBucketStat",
                      "oss:GetBucketInfo",
                      "oss:GetBucketTagging",
                      "oss:GetBucketLifecycle",
                      "oss:GetBucketWorm",
                      "oss:GetBucketVersioning",
                      "oss:GetBucketAcl"
                      ],
            "Resource": "acs:oss:*:*:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oss:ListObjects",
                "oss:GetBucketAcl"
            ],
            "Resource": "acs:oss:*:*:mybucket"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oss:GetObject",
                "oss:GetObjectAcl"
            ],
            "Resource": "acs:oss:*:*:mybucket/*"
        }
    ]
}
```

Example 4: Prohibit RAM users from deleting a bucket

The following code provides an example on how to prohibit RAM users from deleting resources in a bucket named `mybucket`:

```
{
  "Version": "1",
  "Statement": [
      {
          "Effect": "Allow",
          "Action": "oss:*",
          "Resource": [
              "acs:oss:*:*:mybucket",
              "acs:oss:*:*:mybucket/*"
          ]
      },
        {
         "Effect": "Deny",
         "Action": [
           "oss:DeleteBucket"
         ],
         "Resource": [
           "acs:oss:*:*:mybucket"
         ]
     }
   ]
}
```

Example 5: Authorize a RAM user to access multiple folders in a bucket

In this example, a bucket named `mybucket` is used to store photos. The bucket contains multiple folders that are named based on the locations where the photos were captured. Each folder contains subfolders that are named based on the years when the photos were captured.

```
mybucket[Bucket]
  ├── beijing
  │   ├── 2014
  │   └── 2015
  ├── hangzhou
  │   ├── 2013
  │   ├── 2014
  │   └── 2015
  └── qingdao
      ├── 2014
      └── 2015
```

In this example, RAM policies are created to grant a RAM user read-only permissions on the `mybucket/hangzhou/2014/` and `mybucket/hangzhou/2015/` folders. Authorization based on folders is an advanced feature of RAM policies. The complexity of RAM policies is different based on scenarios. You can refer to the RAM policies in the following scenarios to grant permissions to users:

- Authorize a RAM user to only read objects in the `mybucket/hangzhou/2014/` and `mybucket/hangzhou/2015/` folders

  In this scenario, the RAM user knows the full path of the object to be accessed. Therefore, we recommend that you configure the RAM policy to allow the RAM user to access the object by using the full path of the object.

  ```
  {
      "Version": "1",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": [
                  "oss:GetObject"
              ],
              "Resource": [
                  "acs:oss:*:*:mybucket/hangzhou/2014/*",
                  "acs:oss:*:*:mybucket/hangzhou/2015/*"
              ]
          }
      ]
  }
  ```

- Authorize a RAM user to access the `mybucket/hangzhou/2014/` and `mybucket/hangzhou/2015/` folders and list the objects in the folders by using ossutil

  In this scenario, the RAM user does not know the objects in the folders and can use ossutil or call API operations to obtain the information about the objects in the folders. In this case, the permission to perform `ListObjects` must be specified in the policy.

```
{
    "Version": "1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "oss:GetObject"
            ],
            "Resource": [
                "acs:oss:*:*:mybucket/hangzhou/2014/*",
                "acs:oss:*:*:mybucket/hangzhou/2015/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "oss:ListObjects"
            ],
            "Resource": [
                "acs:oss:*:*:mybucket"
            ],
            "Condition":{
                "StringLike":{
                    "oss:Prefix": [
                        "hangzhou/2014/*",
                    "hangzhou/2015/*"
                     ]
                }
            }
        }
    ]
}
```

- Authorize RAM users to access directories in the OSS console

  In this scenario, the RAM user can use the OSS console to access the `mybucket/hangzhou/2014/` and `mybucket/hangzhou/2015/` folders from the root folder by level.

```
{
    "Version": "1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                      "oss:ListBuckets",
                      "oss:GetBucketStat",
                      "oss:GetBucketInfo",
                      "oss:GetBucketTagging",
                      "oss:GetBucketLifecycle",
                      "oss:GetBucketWorm",
                      "oss:GetBucketVersioning",
                      "oss:GetBucketAcl"
                      ],
            "Resource": [
                "acs:oss:*:*:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "oss:GetObject",
                "oss:GetObjectAcl"
            ],
            "Resource": [
                "acs:oss:*:*:mybucket/hangzhou/2014/*",
                "acs:oss:*:*:mybucket/hangzhou/2015/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "oss:ListObjects"
            ],
            "Resource": [
                "acs:oss:*:*:mybucket"
            ],
            "Condition": {
                "StringLike": {
                    "oss:Delimiter": "/",
                    "oss:Prefix": [
                        "",
                        "hangzhou/",
                        "hangzhou/2014/*",
                        "hangzhou/2015/*"
                    ]
                }
            }
        }
    ]
}
```

Example 6: Prohibit a RAM user from deleting an object in a bucket

The following RAM policy prohibits a RAM user from deleting an object in a bucket named `mybucket`:

```
{
  "Version": "1",
  "Statement": [
        {
         "Effect": "Deny",
         "Action": [
           "oss:DeleteObject"
         ],
         "Resource": [
           "acs:oss:*:*:mybucket/*"
         ]
     }
   ]
}
```

Example 7: Prohibit a RAM user from accessing objects with specified tags

The following RAM policy includes a Deny statement that prohibits a RAM user from accessing objects that are stored in the examplebucket bucket and have the status:ok and key1:value1 tags:

```
{
    "Version": "1",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "oss:GetObject"
            ],
            "Resource": [
                "acs:oss:*:1746495857602745:examplebucket/*"
            ],
            "Condition": {
                "StringEquals": {
                    "oss:ExistingObjectTag/status":"ok",
                    "oss:ExistingObjectTag/key1":"value1"
                }
            }
        }
    ]
}
```

Example 8: Authorize a RAM user to access OSS from specified IP addresses

- Add IP address conditions in the `Allow` statement

  The following RAM policy authorizes a RAM user to read objects in a bucket named `mybucket` from only IP addresses in the `192.168.0.0/16` and `172.12.0.0/16` CIDR blocks that are specified in the `Allow` statement:

  ```
  {
      "Version": "1",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": [
                        "oss:ListBuckets",
                        "oss:GetBucketStat",
                        "oss:GetBucketInfo",
                        "oss:GetBucketTagging",
                        "oss:GetBucketAcl"
                        ],
              "Resource": [
                  "acs:oss:*:*:*"
              ]
          },
          {
              "Effect": "Allow",
              "Action": [
                  "oss:ListObjects",
                  "oss:GetObject"
              ],
              "Resource": [
                  "acs:oss:*:*:mybucket",
                  "acs:oss:*:*:mybucket/*"
              ],
              "Condition":{
                  "IpAddress": {
                      "acs:SourceIp": ["192.168.0.0/16", "172.12.0.0/16"]
                  }
              }
          }
      ]
  }
  ```

- Add IP address conditions in the `Deny` statement

  The following RAM policy authorizes a RAM user to perform operations on OSS resources from only IP addresses in the `192.168.0.0/16` CIDR block that is specified in the `Deny` statement. Operations performed from other IP addresses are prohibited.

```
{
    "Version": "1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                      "oss:ListBuckets",
                      "oss:GetBucketStat",
                      "oss:GetBucketInfo",
                      "oss:GetBucketTagging",
                      "oss:GetBucketAcl"
                      ],
            "Resource": [
                "acs:oss:*:*:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "oss:ListObjects",
                "oss:GetObject"
            ],
            "Resource": [
                "acs:oss:*:*:mybucket",
                "acs:oss:*:*:mybucket/*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": "oss:*",
            "Resource": [
                "acs:oss:*:*:*"
            ],
            "Condition":{
                "NotIpAddress": {
                    "acs:SourceIp": ["192.168.0.0/16"]
                }
            }
        }
    ]
}
```

Note In a RAM policy, a Deny statement takes precedence over an Allow statement. Therefore, when a RAM user attempts to read data in the mybucket bucket from an IP address that is not in the `192.168.0.0/16` CIDR block, OSS notifies the RAM user of having no permissions.

Example 9: Use RAM or STS to authorize other users to access OSS resources

In this scenario, you can create a RAM policy to perform the following operations:

- Authorize specific users to access the bucket named `mybucket` and the objects prefixed with `mybucket/file*` .
- Authorize the users to perform the following operations: GetBucketAcl, GetBucket, PutObject, GetObject, and DeleteObject.
- In the Condition field, set UserAgent to java-sdk and the source IP address to `192.168.0.1` . Only users that meet these conditions can access specified OSS resources.
- Authorize the users to list only objects prefixed with foo.

The following RAM policy can meet the requirements of the preceding scenario:

```
{
    "Version": "1",
    "Statement": [
        {
            "Action": [
                "oss:GetBucketAcl",
                "oss:ListObjects"
            ],
            "Resource": [
                "acs:oss:*:177530505652XXXX:mybucket"
            ],
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "acs:UserAgent": "java-sdk",
                    "oss:Prefix": "foo"
                },
                "IpAddress": {
                    "acs:SourceIp": "192.168.0.1"
                }
            }
        },
        {
            "Action": [
                "oss:PutObject",
                "oss:GetObject",
                "oss:DeleteObject"
            ],
            "Resource": [
                "acs:oss:*:177530505652XXXX:mybucket/file*"
            ],
            "Effect": "Allow",
            "Condition": {
               "StringEquals": {
                    "acs:UserAgent": "java-sdk"
                },
                "IpAddress": {
                    "acs:SourceIp": "192.168.0.1"
                }
            }
        }
    ]
}
```

7.1.6.3. Tutorial: Use RAM policies to control access to OSS

This tutorial demonstrates how to use Resource Access Management (RAM) policies to control access to Object Storage Service (OSS) buckets, directories, and objects in the directories.

Background information

RAM policies are configured based on users. You can manage users by configuring RAM policies. For users such as employees, systems, or applications, you can control which resources are accessible. For example, you can create a RAM policy to grant users only read permissions on a bucket.

RAM policies are in the JSON format. A RAM policy includes the following fields:

- Statement: the authorization statement. A RAM policy can include multiple authorization statements.
- Effect: the effect of the policy. Valid values: Allow and Deny.

> **Note** If a RAM policy includes an Allow statement and a Deny statement at the same time, the Deny statement takes precedence over the Allow statement.

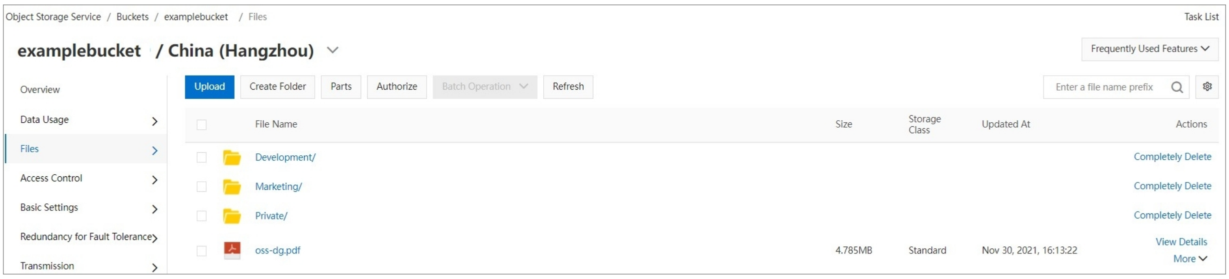
- Action: the authorized actions on resources.

If you use RAM policies, we recommend that you use RAM Policy Editor to generate RAM policies. For more information, see RAM Policy Editor.

Compared with RAM policies, bucket policies can be configured in the OSS console. A bucket owner can grant other users permissions to access OSS resources. For more information, see Configure bucket policies to authorize other users to access OSS resources.

Buckets and directories

Alibaba Cloud OSS uses a flat data model structure instead of a hierarchical one. All objects are stored in buckets. Therefore, OSS does not have directories and subdirectories that are used in hierarchical file systems. However, you can simulate a directory hierarchy in the OSS console to group, classify, and manage objects. The following figure shows some sample directories in the OSS console.

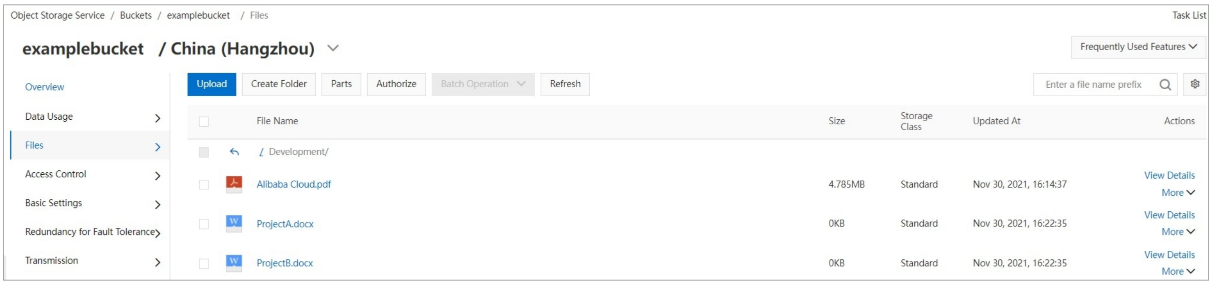


File Name	Size	Storage Class	Updated At	Actions
Development/				Completely Delete
Marketing/				Completely Delete
Private/				Completely Delete
oss-dg.pdf	4.785MB	Standard	Nov 30, 2021, 16:13:22	View Details More

OSS is a distributed object storage service in which objects are identified as key-value pairs. You can retrieve the content of an object based on the object name. For example, an object named *oss-dg.pdf* and the following three directories are stored in a bucket named *examplebucket*: *Development*, *Marketing*, and *Private*.

- When you create the *Development* directory, the OSS console creates an object whose key is `Development/`. A forward slash (`/`) is included in the key as a delimiter.
- When you upload an object named *ProjectA.docx* to the *Development* directory, the OSS console uploads the object and sets its key to `Development/ProjectA.docx`.

  In the key, `Development` is the prefix and the forward slash (`/`) is the delimiter. You can retrieve a list of all objects that share a common prefix and delimiter in the bucket. In the console, if you click the *Development* directory, the objects in the directory are listed. The following figure shows the objects in the Development directory.



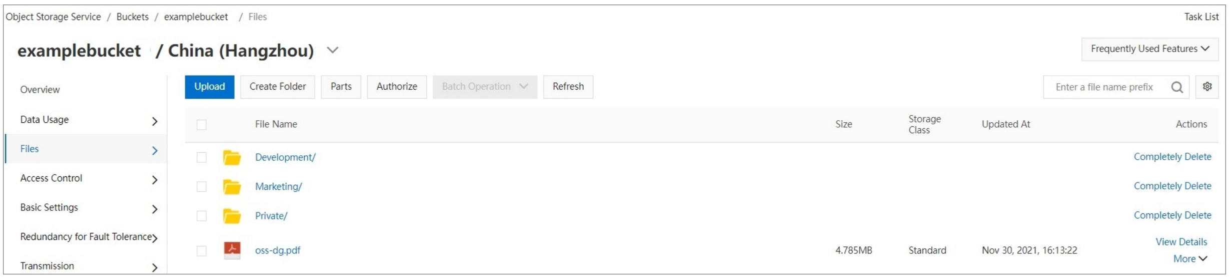
File Name	Size	Storage Class	Updated At	Actions
/ Development/				
Alibaba Cloud.pdf	4.785MB	Standard	Nov 30, 2021, 16:14:37	View Details More
ProjectA.docx	0KB	Standard	Nov 30, 2021, 16:22:35	View Details More
ProjectB.docx	0KB	Standard	Nov 30, 2021, 16:22:35	View Details More

Note To list objects in the *Development* directory of the examplebucket bucket, the console sends a request to OSS to list objects whose names include the specified prefix `Development` and a forward slash (`/`) as the delimiter. In the preceding example, three objects with the following keys are stored in the examplebucket bucket: `Development/Alibaba Cloud.pdf`, `Development/ProjectA.docx`, and `Development/ProjectB.docx`.

Before you start this tutorial, you must understand the concept of root-level bucket content. Assume that the *examplebucket* bucket contains the following objects:

- Development/Alibaba Cloud.pdf
- Development/ProjectA.docx
- Development/ProjectB.docx
- Marketing/data2020.xlsx
- Marketing/data2021.xlsx
- Private/2017/images.zip
- Private/2017/promote.pptx
- oss-dg.pdf

The keys of these objects determine a logical hierarchy with *Development*, *Marketing*, and *Private* as root-level directories and *oss-dg.pdf* as a root-level object. When you click the bucket name in the OSS console, the common prefix and delimiter shared by multiple objects (*Development/*, *Marketing/*, and *Private/*) are displayed as root-level directories. The *oss-dg.pdf* object does not have a prefix. Therefore, it is displayed as a root-level object.



File Name	Size	Storage Class	Updated At	Actions
Development/				Completely Delete
Marketing/				Completely Delete
Private/				Completely Delete
oss-dg.pdf	4.785MB	Standard	Nov 30, 2021, 16:13:22	View Details More

Requests and responses

Before you grant permissions to RAM users, you must understand how the OSS console interacts with OSS when you click a bucket name in the console.

- Send a request to access a bucket

  When you click the *examplebucket* bucket in the OSS console, the console sends a GetBucket (ListObjects) request to OSS.

- Sample request

  ```
  GET /?prefix=&delimiter=/ HTTP/1.1
  Host: examplebucket.oss-cn-hangzhou.aliyuncs.com
  Date: Fri, 24 Feb 2012 08:43:27 GMT
  Authorization: OSS qn6qrrqxo2oawuk53otf****:DNrnx7xHk3sgysx7I8U9I9IY****
  ```

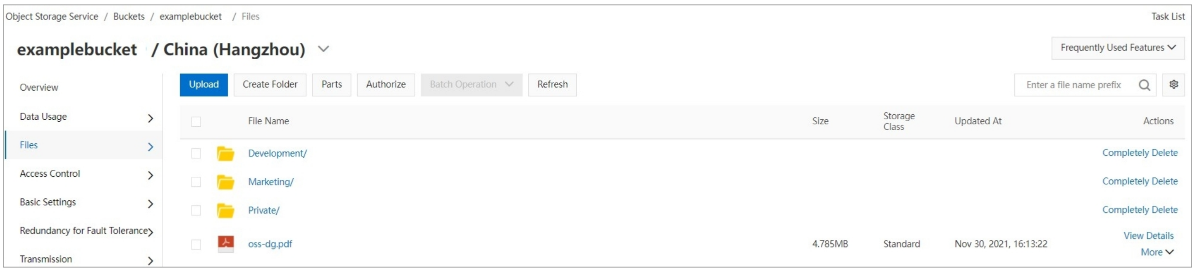
  In the preceding request, the value of the prefix parameter is empty and the value of the delimiter parameter is a forward slash (/).

- Sample response

  ```
  HTTP/1.1 200 OK
  x-oss-request-id: 534B371674E88A4D8906****
  Date: Fri, 7 Aug 2020 08:43:27 GMT
  Content-Type: application/xml
  Content-Length: 712
  Connection: keep-alive
  Server: AliyunOSS
  <?xml version="1.0" encoding="UTF-8"?>
  <ListBucketResult xmlns=¡±http://doc.oss-cn-hangzhou.aliyuncs.com¡±>
  <Name>examplebucket</Name>
  <Prefix></Prefix>
  <Marker></Marker>
  <MaxKeys>100</MaxKeys>
  <Delimiter>/</Delimiter>
      <IsTruncated>false</IsTruncated>
      <Contents>
          <Key>oss-dg.pdf</Key>
          ...
      </Contents>
     <CommonPrefixes>
          <Prefix>Development</Prefix>
     </CommonPrefixes>
        <CommonPrefixes>
          <Prefix>Marketing</Prefix>
     </CommonPrefixes>
        <CommonPrefixes>
          <Prefix>Private</Prefix>
     </CommonPrefixes>
  </ListBucketResult>
  ```

- Response parsing

  The console parses the response returned by OSS and displays the root-level objects and directories in the bucket.



□	File Name	Size	Storage Class	Updated At	Actions
□	Development/				Completely Delete
□	Marketing/				Completely Delete
□	Private/				Completely Delete
□	oss-dg.pdf	4.785MB	Standard	Nov 30, 2021, 16:13:22	View Details More

- Send a request to access a directory stored in the bucket

  When you click the *Development/* directory of the examplebucket bucket in the console, the console sends a GetBucket (ListObjects) request to OSS. The request includes the prefix and delimiter parameters.

  - Sample request

    ```
    GET /?prefix=Development/&delimiter=/ HTTP/1.1
    Host: examplebucket.oss-cn-hangzhou.aliyuncs.com
    Date: Fri, 24 Feb 2012 08:43:27 GMT
    Authorization: OSS qn6qrrqxo2oawuk53otf****:DNrnx7xHk3sgysx7I8U9I9IY****
    ```

    In the preceding request, the value of the prefix parameter is `Development/` and the value of the delimiter parameter is a forward slash (/).

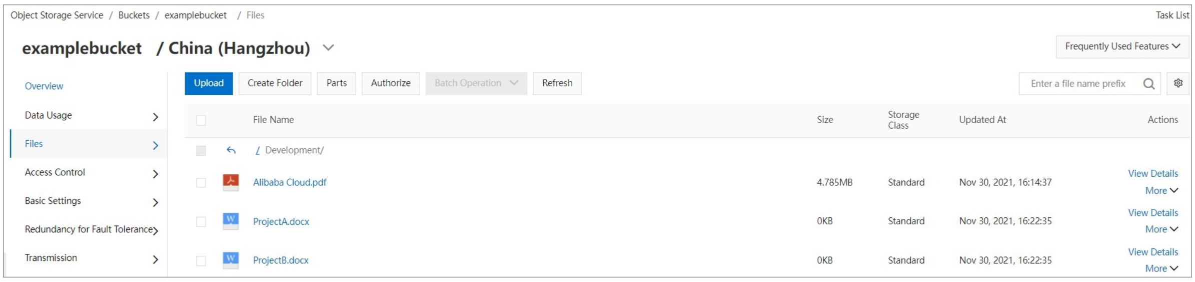
- Sample response

  In the response, OSS returns objects whose keys include the specified prefix.

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
Date: Fri, 7 Aug 2020 08:43:27 GMT
Content-Type: application/xml
Content-Length: 712
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns=¡±http://doc.oss-cn-hangzhou.aliyuncs.com¡±>
<Name>examplebucket</Name>
<Prefix>Development/</Prefix>
<Marker></Marker>
<MaxKeys>100</MaxKeys>
<Delimiter>/</Delimiter>
    <IsTruncated>false</IsTruncated>
    <Contents>
        <Key>ProjectA.docx</Key>
        ...
    </Contents>
    <Contents>
        <Key>ProjectB.docx</Key>
        ...
    </Contents>
    <Contents>
        <Key>Alibaba Cloud.pdf</Key>
        ...
    </Contents>
</ListBucketResult>
```

- Response parsing

  The console parses the response returned by OSS and displays the objects in the Development/ directory.



File Name	Size	Storage Class	Updated At	Actions
/ Development/				
Alibaba Cloud.pdf	4.785MB	Standard	Nov 30, 2021, 16:14:37	View Details More
ProjectA.docx	0KB	Standard	Nov 30, 2021, 16:22:35	View Details More
ProjectB.docx	0KB	Standard	Nov 30, 2021, 16:22:35	View Details More

Scenarios

Assume that you are the owner of the `examplebucket` bucket, and the access control list (ACL) of every object and directory in the bucket is private by default. You want to grant read and write permissions on the `Development` directory stored in the bucket and its subdirectories and objects to RAM user Anne, read-only permissions on the `Marketing` directory and its subdirectories and objects to RAM user Leo. In addition, you want to prevent all RAM users within the current Alibaba Cloud account from accessing the `Private` directory.

Step 1: Create a bucket and upload objects to the bucket.

1. Create a bucket named *examplebucket*.
   i. Log on to the OSS console by using your Alibaba Cloud account.
   ii. Create a bucket named *examplebucket*. For more information, see Create buckets.
2. Create the following directories in the bucket: *Development*, *Marketing*, and *Private*. For more information, see Create directories.
3. Upload objects to specified paths based on the following requirements:
   - Upload the oss-dg.pdf object to the root directory of the examplebucket bucket.
   - Upload the Alibaba Cloud.pdf, ProjectA.docx, and ProjectB.docx objects to the Development directory.
   - Upload the data2020.xlsx and data2021.xlsx objects to the Marketing directory.
   - Upload the images.zip and promote.pptx objects to the Private directory.

   For more information, see Upload objects.

Step 2: Create RAM users Anne and Leo.

Create RAM users Anne and Leo by using the RAM console. For more information about how to create a RAM user, see Create a RAM user.

Step 3: Grant read and write permissions on the Development directory to RAM user Anne.

1. Create a custom policy named *AllowAnneToReadAndWriteFolderDevelopment* and grant RAM user Anne read and write permissions on the Development directory and all objects stored in it.
   i. In the left-side navigation pane, choose **Permissions** > **Policies**.
   ii. On the Policies page, click **Create Policy**.

iii. On the **Create Custom Policy** page, click **JSON**, and then configure the policy content based on the following configurations. Then, click Next Step. On the page that appears, set **Name** to *AllowAnneToReadAndWriteFolderDevelopment*.

```
{
    "Version":"1",
    "Statement":[
        {
            "Effect":"Allow",
            "Action":[
                "oss:ListObjects"
            ],
            "Resource":[
                "acs:oss:*:*:examplebucket"
            ],
            "Condition":{
                "StringEquals":{
                    "oss:Prefix":[
                        "Development",
                        "Development/*"
                    ]
                }
            }
        },
        {
            "Effect":"Allow",
            "Action":[
                "oss:GetObject",
                "oss:PutObject",
                "oss:GetObjectAcl"
            ],
            "Resource":[
                "acs:oss:*:*:examplebucket/Development/*"
            ]
        }
    ]
}
```

iv. Click **OK**.

2. Attach the *AllowAnneToReadAndWriteFolderDevelopment* policy to RAM user Anne. For more information, see Grant permissions to a RAM user.

Step 4: Grant RAM user Leo read-only permissions on the Marketing directory.

Refer to Step 3 to create a custom policy named *AllowLeoToReadAndWriteFolderMarketing* and grant RAM user Leo read-only permissions on the Marketing directory and all objects stored in it. The policy content contains the following configurations:

```
{
    "Version":"1",
    "Statement":[
        {
            "Effect":"Allow",
            "Action":[
                "oss:ListObjects"
            ],
            "Resource":[
                "acs:oss:*:*:examplebucket"
            ],
            "Condition":{
                "StringEquals":{
                    "oss:Prefix":[
                        "Marketing",
                        "Marketing/*"
                    ]
                }
            }
        },
        {
            "Effect":"Allow",
            "Action":[
                "oss:GetObject",
                "oss:GetObjectAcl"
            ],
            "Resource":[
                "acs:oss:*:*:examplebucket/Marketing/*"
            ]
        }
    ]
}
```

Step 5: Deny all RAM users within the current Alibaba Cloud account access to the Private directory.

1. Create a user group and add members to it.

   For more information about how to create a user group, see Create a user group. After you create the user group, add all RAM users within the current Alibaba Cloud account to the group. For more information, see Add a RAM user to a RAM user group.

2. Create a custom policy named *DenyAllRamToAccessFolderPrivate* and deny all RAM users within the current Alibaba Cloud account access to the Private directory.

   i. In the left-side navigation pane, choose **Permissions** > **Policies**.

ii. On the Policies page, click **Create Policy**.

iii. On the **Create Custom Policy** page, click **JSON**, and then configure the policy content based on the following configurations. Then, click Next Step. On the page that appears, set **Policy Name** to *DenyAllRamToAccessFolderPrivate*.

```
{
    "Version":"1",
    "Statement":[
        {
            "Effect":"Deny",
            "Action":[
                "oss:*"
            ],
            "Resource":[
                "acs:oss:*:*:examplebucket/Private/*"
            ],
            "Condition":{
            }
        },
        {
            "Effect":"Deny",
            "Action":[
                "oss:ListObjects"
            ],
            "Resource":[
                "acs:oss:*:*:*"
            ],
            "Condition":{
                "StringEquals":{
                    "oss:Prefix":[
                        "Private/",
                        "Private/*"
                    ]
                }
            }
        }
    ]
}
```

iv. Click **OK**.

3. Attach the *DenyAllRamToAccessFolderPrivate* policy to the user group. For more information, see Grant permissions to a RAM user group.

   After you attach the policy to the user group, the RAM users in the group cannot access the `Private` directory of the `examplebucket` bucket. In addition, when the RAM users send a request to list the `Private/2017/images.zip` and `Private/2017/promote.pptx` objects in the `Private` directory, OSS returns an error response.

7.1.6.4. Tutorial: Authorize a RAM user of another Alibaba Cloud account by creating a RAM role

By default, OSS resources can be accessed only by their owners. To authorize another user to access your OSS resources, you can grant permissions to the user by creating a RAM role.

Context

Example: Company A wants to authorize Company B to access resources owned by Company A. However, Company A does not want to provide Company B with a RAM users' credentials. In this case, Company A can create a RAM role and grant the RAM role permissions to access the OSS resources of Company A. Company B can use a RAM user to assume the RAM role. This way, Company B can access the OSS resources owned by Company A.

Step 1: Company A creates a RAM role and grant the RAM role permissions to access the OSS resources of Company A

Company A must create a RAM role that has permissions to access the OSS resources owned by Company A.

1. Log on to the RAM console as Company A.
2. Choose **Identities > Roles**, and click **Create Role**.
3. In the **Create Role** pane, set Trusted entity type to **Alibaba Cloud Account** in the **Select Role Type** step. Click **Next**.
4. Configure the following parameters in the **Configure Role** step:
   - **RAM Role Name**: Enter a name for the RAM role. In this example, specify `admin-oss`.
   - Note: optional. Enter a description for the RAM role. In this example, the field is left empty.
   - **Select Trusted Alibaba Cloud Account**: Select **Other Alibaba Cloud Account** and enter the UID of an Alibaba Cloud account that belongs to Company B. In this example, specify `17464958576******`.
5. Click **OK**.
6. In the **Finish** step, click **Add Permissions to RAM Role**.
7. In the Add Permissions pane, select **System Policy** in the Select Policy section. Find and click AliyunOSSReadOnlyAccess, which grants only the read permissions on OSS resources. The AliyunOSSReadOnlyAccess policy is displayed in the **Selected** section on the right side. Click **OK**.

   AliyunOSSReadOnlyAccess allows your customers to access all your buckets in OSS. You can customize a policy to grant permissions to read only a part of your buckets or folders. For more information, see Overview.

If you want to specify that the RAM role can be assumed only by specified RAM users, you can modify the trusted entity of the RAM role. For more information, see Edit the trust policy of a RAM role.

Step 2: Company B creates a RAM user and grants the RAM user permissions to assume RAM roles

Company B must create a RAM user that has permissions to assume RAM roles. Company B can use the RAM user to assume a RAM role from Company A.

1. Log on to the RAM console as Company B.
2. In the left-side navigation pane, choose **Identities > Users**. On the Users page, click **Create User**.
3. On the **Create User** page, enter a logon name in the **Logon Name** field. Enter a display name in the **Display Name** field. Select **Console Password Logon** as the access mode and configure the logon information.
4. Click **OK**.
5. In the **Users** list, select the user you create and click **Add Permissions**.

   Save RAM user information to prevent loss.
6. In the Add Permissions pane, select **System Policy** in the Select Policy section. Find and click AliyunSTSAssumeRoleAccess, which grants the user permissions to call the AssumeRole operation in STS. The AliyunSTSAssumeRoleAccess policy is displayed in the **Selected** section on the right side. Click **OK**.

Step 3: Company B uses the created RAM user to log on to the Alibaba Cloud Management console and assume the RAM role created by Company A

Company B uses the created RAM user to log on to the Alibaba Cloud Management console and switches the identity to the RAM role created by Company A.

1. Log on to the Alibaba Cloud Management console as the RAM user of Company B. Switch Log on to the Alibaba Cloud Management Console as a RAM user.
2. Move the pointer over the profile picture in the upper-right corner of the console. Click **Switch Identity**.
3. On the **Switch Role** page, enter the information about the RAM role and click **Switch**.

   Enter the following information for the RAM role:

   - **Enterprise Alias/Default Domain Name**: Enter the alias or default domain name of Company A. For more information, see Terms.

     A default domain name is used in this example. Enter the default domain name `1178810717******.onaliyun.com`. `1178810717******` is the UID of an Alibaba Cloud account of Company A.
   - **Role Name**: Enter `admin-oss`, which is the name of the RAM role created by Company A.
4. Click OSS console to log on to the OSS console and manage the OSS resources owned by Company A.

References

You can also authorize a RAM user of another Alibaba Cloud account by adding a bucket policy. For more information, see Tutorial: Authorize a RAM user under another Alibaba Cloud account by adding a bucket policy.

7.2. Disaster recovery

7.2.1. ZRS

User data can be stored and distributed across three zones within the same region by using zone-redundant storage (ZRS). Even if one zone becomes unavailable, the data can still be accessed. Zone-redundant storage (ZRS) can provide data durability (designed for) of 99.9999999999% (twelve 9's) and service availability of 99.995%.

ZRS offers data center-level disaster recovery capabilities. When a data center is unavailable due to network disconnections, power outages, or other disaster events, OSS can continue to provide highly consistent services. This way, services are not interrupted and data is not lost during failovers. This meets the strict requirements of key business systems that the recovery time objective (RTO) and the recovery point objective (RPO) must be zero.

Implementation methods

You can configure ZRS for a bucket only when you create the bucket. For an existing bucket, you can use migration tools such as ossimport and Data Transport to migrate data from the existing bucket to another bucket that has ZRS enabled.

You can use the methods described in the following table to enable ZRS when you create a bucket.

Implementation method	Description
Console	A user-friendly and intuitive web application
ossbrowser	An easy-to-operate graphical tool
ossutil	A high-performance command-line tool
Java SDK	SDK demos for a variety of programming languages
Python SDK	
Go SDK	
C++ SDK	
Node.js SDK	

Usage notes

- Supported regions

  ZRS is supported in the following regions: China (Shenzhen), China (Beijing), China (Hangzhou), China (Shanghai), China (Hong Kong), Singapore, and Indonesia (Jakarta) regions.
- Billing information

  ZRS incurs more costs than locally redundant storage (LRS). For more information, see the OSS pricing page.

Supported storage classes

ZRS supports the Standard and Infrequent Access (IA) storage classes. The following table compares the two storage classes.

Item	Standard	Infrequent Access (IA)
Data durability (designed for)	99.9999999999% (twelve 9's)	99.9999999999% (twelve 9's)
Service availability	99.995%	99.50%
Minimum billable size of objects	None	64 KB
Minimum storage period	None	30 days
Data retrieval fee	None	Billed based on the size of retrieved data. Unit: GB.
Data access	Real-time access with low latency (within milliseconds)	Real-time access with low latency (within milliseconds)
Image Processing (IMG)	Supported	Supported

7.2.2. Cross-region replication

Cross-region replication (CRR) enables the automatic and asynchronous (near real-time) replication of objects across buckets in different OSS regions. Operations such as the creation, overwriting, and deletion of objects can be synchronized from a source bucket to a destination bucket.

Implementation methods

You can configure CRR in the OSS console or by using OSS SDK for Java:

- Console
- Java SDK

Scenarios

CRR can meet your requirements for cross-region disaster recovery (DR) and data replication. Objects in a destination bucket are exact replicas of those in a source bucket. They have the same object names, versioning information, object content, and object metadata such as the creation time, owner, user metadata, and object access control lists (ACLs). You can configure CRR rules in the following scenarios to meet your requirements:

- Compliance requirements

  OSS stores multiple replicas of objects in physical disks. However, to meet compliance requirements, the replicas of data must be stored at a geographical distance from each other. CRR allows you to replicate data between geographically distant data centers to meet compliance requirements.

- Minimum latency

  You have customers located in two geographical locations. To minimize the latency that occur when the customers access objects, you can maintain the replicas of objects in data centers that are geographically closer to the customers.

- Data backup and disaster recovery (BDR)

  You have strict requirements for data security and availability. You want all data written in one data center replicated to another data center. If one data center is damaged in a catastrophic event such as an earthquake or a tsunami, you can use data that is backed up in the other data center.

- Data replication

  For business reasons, you may need to migrate data from one OSS data center to another data center.

- Operational reasons

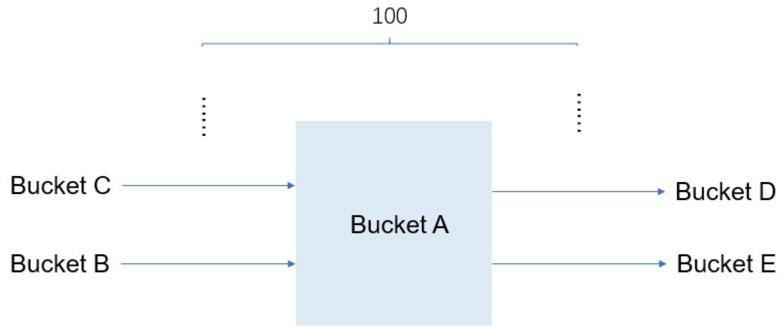
  You have compute clusters that are deployed in two data centers to analyze the same group of objects. You can maintain the replicas of the objects in the two regions.

Capabilities

CRR supports the following capabilities:

- Data synchronization between buckets in different regions

  You can configure CRR rules to synchronize data from a source bucket to multiple destination buckets. You can configure up to 100 CRR rules for a bucket. A bucket can be specified as a source bucket or a destination bucket.



  If your business requires more than 100 CRR rules for a bucket, contact the technical support.

- Real-time data synchronization

  You can monitor data that is added, deleted, or modified in real time and synchronize these changes to a destination bucket. Operations performed on objects that are smaller than 2 MB are synchronized within minutes to ensure data consistency between the source and the destination buckets.

- Historical data migration

Historical data can be synchronized from a source bucket to a destination bucket. This way, two identical data replicas are individually stored in the source bucket and destination bucket.

- Real-time display of the synchronization progress

  You can view the last synchronization time for real-time data synchronization and the percentage of synchronization for historical data migration.

- Versioning

  CRR ensures the consistency between the data in source and destination buckets for which versioning is enabled. If you configure a CRR rule to synchronize only added and modified data, delete operations performed on the specified version of an object in the source bucket are not synchronized to the destination bucket. However, delete markers created in the source bucket are synchronized to the destination bucket.

- Transfer acceleration

  You can use transfer acceleration to speed up data transfer when CRR tasks are performed across regions within mainland China and outside mainland China. For more information about transfer acceleration, see Transfer acceleration.

- Replication of encrypted data

  CRR allows you to replicate objects that are encrypted by using SSE-KMS or SSE-OSS at the server side. For more information, see Cross-region replication in specific scenarios.

- Event notification and real-time log query

  You can use the following methods to be notified when changes are made to objects in source and destination buckets during CRR. These changes include adding, modifying, deleting, and overwriting objects.

  - Set the event type to the following values in the event notification rule: `ObjectReplication:ObjectCreated` , `ObjectReplication:ObjectRemoved` , and `ObjectReplication:ObjectModified` . For more information, see Overview.
  - Enable real-time log query in the OSS console to obtain the statistics for operations performed on objects. For more information, see Query real-time logs.

Usage notes

When you use CRR, take note of the following items:

- Supported regions
  - You must enable transfer acceleration when you perform CRR between regions within mainland China and regions outside mainland China.
  - CRR rules based on object tags can be configured only in the following scenarios:
    - The source region is China (Hangzhou), and the destination region is a region except for China (Hangzhou).
    - The source region is Australia (Sydney), and the destination region can be a different region outside mainland China.
- Billing
  - After you configure a CRR rule between two buckets, you are charged for the traffic generated to replicate objects from the source bucket to the destination. For more information, see Traffic fees.
  - Each time an object is synchronized, OSS counts the number of requests and charges you on a pay-as-you-go basis. For more information, see .
  - If you enable transfer acceleration, you are charged for this feature. For more information, see Transfer acceleration fees.
- Replication time

  In CRR, data is asynchronously replicated in near real-time. It takes several minutes to several hours to copy data from a source bucket to a destination bucket based on the size of the data.

- Limits
  - You can configure CRR between two unversioned buckets or versioned buckets.
  - The versioning status of two buckets between which a CRR rule is configured cannot be changed.
  - You can manage two buckets between which a CRR rule is configured at the same time. Therefore, the object replicated from the source bucket may overwrite the object that has the same name in the destination bucket.

7.2.3. SRR

Same-region replication (SRR) enables the automatic and asynchronous (near real-time) replication of objects across buckets in the same Object Storage Service (OSS) region. Operations such as the creation, overwriting, and deletion of objects can be synchronized from a source bucket to a destination bucket.

Scenarios

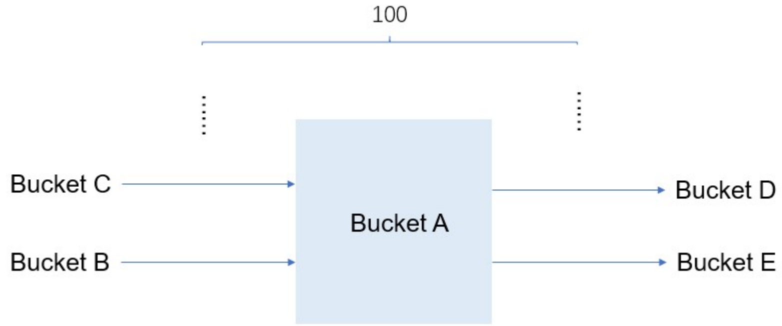
If your data cannot be transferred out of your country or region based on the compliance requirements of local laws and regulations, you can configure SRR rules to store the replicas of data in the source bucket in multiple destination buckets within the same region. Objects in a destination bucket are exact replicas of those in a source bucket. They have the same object names, versioning information, object content, and object metadata such as the creation time, owner, user metadata, and object access control lists (ACLs).

Features

SRR provides the following features:

- Data synchronization between buckets in the same region

You can configure SRR rules to synchronize data from a source bucket to multiple destination buckets in the same region. By default, you can configure up to 100 SRR rules for a bucket. The bucket can be specified as a source bucket or a destination bucket.



If you want to configure more than 100 SRR rules for a bucket,contact technical support.

- Real-time data synchronization

  You can monitor data that is added, removed, or modified in real time and synchronize these changes to a destination bucket. Operations performed on objects that are smaller than 2 MB are synchronized within minutes to ensure data consistency between the source and destination buckets.

- Historical data migration

  Historical data can be synchronized from a source bucket to a destination bucket. This way, two identical data replicas are individually stored in the source bucket and destination bucket.

- Real-time display of the synchronization progress

  You can view the last synchronization time for real-time data synchronization and the percentage of synchronization for historical data migration.

- Versioning

  SRR ensures eventual consistency between the data in source and destination buckets for which versioning is enabled. If you configure an SRR rule to synchronize only added and modified data, delete operations performed on the specified version of an object in the source bucket are not synchronized to the destination bucket. However, delete markers created in the source bucket are synchronized to the destination bucket.

Usage notes

- Billing

  After SRR is enabled, you are not charged for the traffic that is generated when you use SRR to replicate objects in OSS. Each time an object is synchronized, OSS accumulates the number of requests. However, you are not charged for the requests.

- Replication time

  In SRR, data is replicated asynchronously. Depending on the amount of data, it can take a few minutes to several hours to replicate data to the destination bucket.

Limits

- You can configure SRR between two unversioned buckets or two versioned buckets.
- The versioning status of two buckets between which an SRR rule is configured cannot be changed.
- You can manage two buckets between which an SRR rule is configured at the same time. Therefore, the object replicated from the source bucket may overwrite the object that has the same name in the destination bucket.

Use the OSS console

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the page that appears, click the name of the bucket for which you want to enable SRR.
3. In the left-side navigation pane, choose **Redundancy for Fault Tolerance > Same-Region Replication**.
4. In the **Same-Region Replication** section, click **Configure**.
5. Click **Same-Region Replication**.
6. In the **Same-Region Replication** panel, configure the parameters described in the following table.

Parameter	Description
Source Region	The region in which the current bucket is located.
Source Bucket	The name of the current bucket.
Destination Bucket	Select the destination bucket to which you want to synchronize data.
Applied To	Select the source data that you want to synchronize. ○ **All Files in Source Bucket**: OSS synchronizes all objects from the source bucket to the destination bucket. ○ **Files with Specified Prefix**: OSS synchronizes the objects whose names contain a specified prefix from the source bucket to the destination bucket. You can specify up to 10 prefixes.

Parameter	Description
Object Tagging	The tags of objects that you want to synchronize to the destination bucket. Objects that have the specified tags are synchronized to the destination bucket. Select **Configure Rules** and add tags in key-value pairs. You can add up to 10 tags. To configure this parameter, make sure that the following conditions are met: ○ Tags are configured for objects. For more information, see Configure object tagging. ○ Versioning is enabled for the source bucket and the destination bucket. ○ The Operations parameter is set to **Add/Change**.
Operations	Select the operations to synchronize. ○ **Add/Change**: OSS synchronizes only the added or changed data from the source bucket to the destination bucket. ○ **Add/Delete/Change**: OSS synchronizes all data changes including the create, overwrite, and delete operations on objects from the source bucket to the destination bucket.
Replicate Historical Data	Specifies whether to synchronize historical data in the source bucket before you enable SRR for the source bucket. ○ **Yes**: OSS synchronizes historical data to the destination bucket. **Notice** When historical data is synchronized, objects in the source bucket may overwrite objects that have the same names in the destination bucket. To avoid data loss, we recommend that you enable versioning for the source and destination buckets. ○ **No**: OSS synchronizes only objects that are uploaded or updated after the SRR rule takes effect to the destination bucket.
KMS-based Encryption	If KMS-based encryption is configured for the source objects or destination bucket, you must select **KMS-based Encryption** and configure the following parameters: ○ **CMK ID**: The customer master key (CMK) that is used to encrypt the destination object. If you want to use a CMK to encrypt objects, you must create a CMK in the same region as the destination bucket in the Key Management Service (KMS) console. For more information, see Create a CMK. ○ **RAM Role Name**: The RAM role that is authorized to perform KMS-based encryption on the destination object. ■ **New RAM Role**: A RAM role is created to perform KMS-based encryption on the destination object. The name of the RAM role is in the following format: `kms-replication-source bucket name-destination bucket name` . ■ **AliyunOSSRole**: The AliyunOSSRole role is used to perform KMS-based encryption on the destination object. If the AliyunOSSRole role does not exist, OSS automatically creates the AliyunOSSRole role when you select this option. **Note** You can use HeadObject to query the encryption status of the source object and use GetBucketEncryption to query the encryption state of the destination bucket.

7. Click **OK**.
   - An SRR rule cannot be edited or deleted after it is created.
   - The synchronization starts immediately after an SRR rule is configured. You can view the synchronization progress on the **Same-Region Replication** page.
   - It can take several minutes to several hours for the data to be synchronized to the destination bucket based on the amount of data.

Use the RESTful API

If your program requires more custom options to enable SRR, you can call RESTful API operations. In this case, you must manually write code to calculate the signature. For more information, see PutBucketReplication.

7.2.4. CRR in specific scenarios

This topic describes how cross-region replication (CRR) works when it is used with versioning, lifecycle rules, server-side encryption, and retention policies.

Use CRR with versioning

Take note of the following limits when you use CRR with versioning:

- You can enable CRR only between two buckets that are both versioned or unversioned. The versioning state of the source bucket and the destination bucket cannot be changed.
- Versioning cannot be suspended for the source bucket or destination bucket during data replication. To suspend versioning for the source bucket and destination bucket, you must first delete the CRR rule configured for the buckets first.

The following table describes the results of operations performed by Object Storage Service (OSS) in CRR when an object is deleted from the versioned source bucket.

Request method	Operation	Result
Send a DeleteObject request in which the version ID of the object is not specified.	Add/Change	A delete marker is created for the object in the source bucket and is synchronized to the destination bucket.
	Add/Delete/Change	A delete marker is created for the object in the source bucket and is synchronized to the destination bucket.
	Add/Change	The deletion is not synchronized to the destination bucket.

Request method	Operation	Result
Send a DeleteObject request in which the version ID of the object is specified.	Add/Delete/Change	The deletion is synchronized to the destination bucket.

For more information about how to configure data synchronization policies for versioned buckets, see Configure CRR.

Use CRR with lifecycle rules

When you use CRR with versioning, multiple previous object versions are synchronized to the destination bucket and incur additional storage costs. To reduce the costs, we recommend that you configure lifecycle rules for buckets to control storage costs and retain required data. For more information, see Lifecycle rules based on the last modified time.

Take note of the following items when you use CRR with lifecycle rules:

- In CRR, only the operations performed based on the lifecycle rules but not the lifecycle rules are synchronized to the destination bucket. To apply the same lifecycle rules as the source bucket on the objects in the destination buckets, configure the same lifecycle rules for the destination bucket.
- If a lifecycle rule is configured for the destination bucket, note that the created time of an object replicated to the destination bucket is the time when the object is created in the source bucket but not the time when it is replicated to the destination bucket.
- If an object is deleted from the source bucket based on a lifecycle rule while the object is being replicated to the destination bucket, the replication may continue, and the replicated object in the destination bucket is retained.

Use CRR with server-side encryption

CRR supports unencrypted objects and objects encrypted by using SSE-KMS and SSE-OSS. For more information, see Server-side encryption.

The following table describes the encryption status of the destination object when CRR is used with server-side encryption.

Encryption status of the source object	Encryption status of the destination bucket	Whether SSE-KMS is used to encrypt the destination object	Encryption status of the destination object
Unencrypted	Unencrypted	N/A	Unencrypted
	SSE-OSS	N/A	SSE-OSS
	SSE-KMS without a specified CMK ID	N/A	SSE-KMS without a specified CMK ID
	SSE-KMS with a specified CMK ID	Yes A SyncRole and a CMK ID are configured.	SSE-KMS with a specified CMK ID
		No	N/A. The source object cannot be replicated to the destination bucket.
SSE-OSS	Unrestricted	N/A	SSE-OSS
SSE-KMS without a specified CMK ID	Unrestricted	Yes A SyncRole and a CMK ID are configured.	SSE-KMS with a specified CMK ID
		No	SSE-KMS without a specified CMK ID
SSE-KMS with a specified CMK ID	Unrestricted	Yes A SyncRole and a CMK ID are configured.	SSE-KMS with a specified CMK ID
		No	N/A. The source object cannot be replicated to the destination bucket.

For more information about how to use SSE-KMS to encrypt objects when you configure CRR rules, see Configure CRR.

Use CRR with retention policies

After a retention policy configured for a bucket is locked, you can read objects from or upload objects to the bucket. However, the objects in the bucket cannot be overwritten or deleted within the retention period.

For more information about retention policies, see Retention policy.

The following table describes whether the source object can be synchronized to the destination bucket when CRR is used with retention policies.

Whether the source object is in the retention period	Allowed operation in the source bucket	Whether the destination object in the retention period	Whether the source object is synchronized to the destination bucket
	Create an object	Yes	No

Whether the source object is in the retention period No	Allowed operation in the source bucket	Whether the destination object in the retention period	Whether the source object is synchronized to the destination bucket
	Overwrite an object	Yes	No
	Delete an object	Yes	No
No	Create an object	No	Yes
	Overwrite an object	No	Yes
	Delete an object	No	Yes
Yes	Create an object	N/A	Yes

7.2.5. Data replication troubleshooting

After you configure a cross-region replication (CRR) rule for a source bucket and a destination bucket, if the objects in the source bucket are not replicated to the destination bucket, check the following reasons to locate and resolve the problem.

- Duration

  In CRR, data is asynchronously replicated in near real time. It takes several minutes to several hours to copy data from the source bucket to the destination bucket based on the size of the data. If objects in the source bucket are large in size, wait a moment and check whether the objects are replicated to the destination bucket.

- Source bucket configurations
  - Check whether the status of the data synchronization task is Enabled.
  - Check whether the prefix of the objects to replicate is correctly configured.
    - To synchronize objects whose names contained a specific prefix from the source bucket to the destination bucket, set the Prefix parameter to the prefix when you configure the data replication rule. For example, if you set the Prefix parameter to log, only objects whose names contain the log prefix, such as log/date1.txt and log/date2.txt, are replicated. Objects whose names do not contain the log prefix, such as date3.txt, are not replicated.
    - To synchronize all objects from the source bucket to the destination bucket, set the Prefix parameter to empty.

- Replication mechanism

  If an object in the source bucket is replicated from a bucket rather than the destination bucket based on another data replication rule, the object is not replicated to the destination bucket. For example, if you configure a data replication rule to replicate objects from Bucket A to Bucket B and another replication rule to replicate objects from Bucket B to Bucket C, objects that are replicated from Bucket A to Bucket B are not replicated to Bucket C.

- Versioning status

  Both the source bucket and destination bucket must be versioned or unversioned.

7.3. Data encryption

7.3.1. Server-side encryption

Object Storage Service (OSS) supports server-side encryption. When you upload objects, OSS encrypts and stores the data. When you download objects, OSS decrypts the data and returns the decrypted data. The returned HTTP request header indicates that the data is encrypted on the server side.

> **Notice** Server-side encryption cannot automatically encrypt data retrieved by using mirroring-based back-to-origin.

Encryption methods

OSS protects static data by using server-side encryption. You can use this method in scenarios in which additional security or compliance is required, such as the storage of deep learning samples and online collaborative documents.

Only one server-side encryption method can be used for an object at a time. OSS provides the following server-side encryption methods that you can use in different scenarios:

- Server-side encryption by using Key Management Service (SSE-KMS)

  You can use the default customer master key (CMK) or specify a CMK to encrypt or decrypt large amounts of data. This method is cost-effective because you do not need to send user data to the KMS server over networks to encrypt and decrypt data.

  > **Notice**
  > - You are charged when you call API operations to encrypt or decrypt data by using CMKs. For more information about the fees, see Billing.
  > - The key used to encrypt the object is also encrypted and written into the metadata of the object.
  > - Server-side encryption that uses the default CMK (SSE-KMS) only encrypts the data in the object. The metadata of the object is not encrypted.

- Server-side encryption by using OSS-managed keys (SSE-OSS)

  You can use SSE-OSS to encrypt each object. To improve security, OSS uses master keys that are rotated on a regular basis to encrypt data keys. You can use this method to encrypt and decrypt multiple objects at a time.

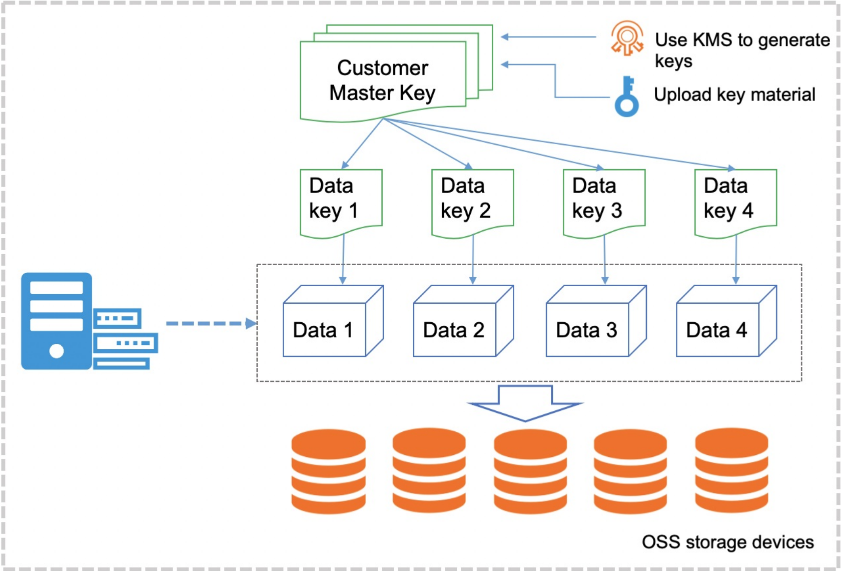
Implementation methods

Implementation method	Description
Console	A user-friendly and intuitive web application
ossutil - bucket-encryption - cp	A high-performance command-line tool
Java SDK	SDK demos for a variety of programming languages
Python SDK	
Go SDK	

Server-side encryption by using CMKs stored in KMS

You can use a CMK stored in KMS to generate CMK encrypted data. The envelope encryption mechanism further prevents unauthorized data access. KMS eliminates the need to manually maintain the security, integrity, and availability of your keys. You need only to focus on data encryption, data decryption, and digital signature generation and verification based on your business requirements.

The following figure shows the logic of server-side encryption based on SSE-KMS.



- Multi-level key envelope encryption
- Support for encryption and decryption by using BYOK
- Keys managed by KMS
- A random key generated for each object

When you use SSE-KMS to encrypt data, you can use the following keys:

- Use CMKs stored in KMS

  For this method, OSS generates different keys to encrypt different objects by using the default CMK stored in KMS, and automatically decrypts an object when the object is downloaded. The first time you use SSE-OSS, OSS creates a CMK on the KMS platform.

  You can use the following configuration methods:

  - Configure the default server-side encryption method for a bucket

    Set the default server-side encryption method for a bucket to KMS, but do not specify a CMK ID. Objects uploaded to this bucket are encrypted.

  - Configure an encryption method for a specified object

    When you upload an object or modify the metadata of an object, include the `x-oss-server-side-encryption` parameter in the request and set the parameter value to `KMS`. In this case, OSS uses the default CMK stored in KMS and uses the AES-256 encryption algorithm to encrypt the object. For more information, see PutObject.

- Use Bring Your Own Key (BYOK)

  After you use the BYOK material in the KMS console to generate a CMK, the keys generated by a specified CMK stored in KMS are used to encrypt different objects and the specified CMK ID is recorded in the metadata of the encrypted object. Objects are decrypted only when they are downloaded by users who have the permissions to decrypt the objects.

  You may obtain your BYOK material from one of the following sources:

  - BYOK material provided by Alibaba Cloud: When you create a key on KMS, you can select **Alibaba Cloud KMS** as the source of the key material.
  - BYOK material provided by the user: When you create a key on KMS, you can select **external** as the source of the key material and import the external key material. For more information about how to import the key material, see Import key material.

  You can use the following configuration methods:

  - Configure the default server-side encryption method for a bucket

    Set the default server-side encryption method to SSE-KMS, and specify the CMK ID. Objects uploaded to this bucket are encrypted.

  - Configure an encryption method for the requested object

    When you upload an object or modify the metadata of an object, include the `x-oss-server-side-encryption` parameter in the request and set the value of the parameter to `KMS`. In addition, include the `x-oss-server-side-encryption-key-id` parameter in the request and set the parameter value to a specified CMK ID. In this case, OSS uses the specified CMK stored in KMS and the AES-256 encryption algorithm to encrypt the object. For more information, see PutObject.

Server-side encryption and decryption by using OSS-managed keys

OSS generates and manages the keys used to encrypt data, and provides strong and multi-factor security measures to protect data. OSS server-side encryption uses AES-256, which is one of the advanced encryption standard algorithms to encrypt your data.

You can use the following configuration methods:

- Configure the default server-side encryption method for a bucket

  Set the default encryption method to SSE-OSS and specify the encryption algorithm as AES-256. This way, all objects uploaded to this bucket are encrypted by default.

- Configure an encryption method for the requested object

  When you upload an object or modify the metadata of an object, include the `x-oss-server-side-encryption` parameter in the request and set the parameter value to `AES256`. In this case, the requested object is encrypted by using an OSS-managed key. For more information, see PutObject.

Required permissions

To use server-side encryption by using the credentials of a Resource Access Management (RAM) user in the following scenarios, you must have the following permissions.

- To configure the default encryption method for a bucket, you must have the following permissions:
  - The permissions to manage the bucket.
  - The permissions to call `PutBucketEncryption` and `GetBucketEncryption` operations.
  - The permissions to call the `ListKeys` , `Listalias` , `ListAliasesByKeyId` , and `DescribeKeys` operations when you set the encryption method to SSE-KMS and use a specified CMK ID to encrypt data. To grant a RAM user the preceding permissions, configure a RAM policy based on the following example in the RAM console:

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:List*",
        "kms:DescribeKey"
      ],
      "Resource": [
        "acs:kms:*:1416614965936597:*" // In this example, the user is allowed to use all CMKs that belong to the account. If you want only one C
MK to be used by the RAM user, specify the CMK ID of the CMK.
      ]
    }
  ]
}
```

- To upload an object to a bucket that has the default encryption method configured, you must have the following permissions:
  - The permissions to upload objects to the bucket.
  - The permissions to call the `ListKeys` , `Listalias` , `ListAliasesByKeyId` , `DescribeKeys` , `GenerateDataKey` , and `kms:Decrypt` operations when you set the encryption method to KMS and use a specified CMK ID to encrypt data. To grant a RAM user the preceding permissions, configure a RAM policy based on the following example in the RAM console:

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:List*",
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": [
        "acs:kms:*:1416614965936597:*" // In this example, the user is allowed to use all CMKs that belong to the account. If you want only one C
MK to be used by the RAM user, specify the CMK ID of the CMK.
      ]
    }
  ]
}
```

- To download an object from a bucket that has the default encryption method configured, you must have the following permissions:
  - The permissions to access objects in the bucket.

- The permissions to call the `Decrypt` operation when you set the encryption method to KMS and use a specified CMK ID to encrypt data. To grant a RAM user the preceding permissions, configure a RAM policy based on the following example in the RAM console:

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
    "kms:Decrypt"
      ],
      "Resource": [
        "acs:kms:*:1416614965936597:*" // In this example, the RAM user has the permissions to use all CMKs to decrypt data. If you want only one
CMK to be used by the RAM user, specify the CMK ID.
      ]
    }
  ]
}
```

FAQ

Does OSS encrypt existing objects in a bucket after I configure server-side encryption for the bucket?

No, OSS encrypts only objects that are uploaded after server-side encryption is configured for the bucket and does not encrypt existing objects in the bucket. If you want to encrypt existing objects in a bucket, you can call the CopyObject operation to overwrite the existing objects.

7.3.2. Client-side encryption

If client-side encryption is performed, objects are encrypted on the local client before they are uploaded to OSS. This topic describes how to perform client-side encryption.

Disclaimer

- When you use client-side encryption, you must make sure the integrity and validity of the customer master key (CMK). If the CMK is used incorrectly or lost due to improper maintenance, you are liable for any losses or consequences caused by decryption failures.
- When you copy or migrate encrypted data, you must make sure the integrity and validity of the object metadata related to client-side encryption. If the object metadata related to client-side encryption is used incorrectly or lost due to improper maintenance, you are liable for any losses or consequences caused by decryption failures.

Implementation modes

You can use the following SDKs to implement client-side encryption:

- Java SDK
- Python SDK
- Go SDK
- C++ SDK

Background information

In client-side encryption, a random data key is generated for each object to perform symmetric encryption on the object. The client uses a CMK to encrypt the random data key. The encrypted data key is uploaded as a part of the object metadata and stored in the OSS server. When an encrypted object is downloaded, the client uses the CMK to decrypt the random data key and then uses the data key to decrypt the object. The CMK is used only on the client and is not transmitted over the network or stored in the server. This ensures data security.

> Notice
>
> - Client-side encryption supports multipart upload for objects larger than 5 GB. When you use multipart upload to upload an object, you must specify the total size of the object and the size of each part. The size of each part except for the last part must be the same and be a multiple of 16 bytes.
> - After you upload objects encrypted on the client, object metadata related to client-side encryption is protected and you cannot call CopyObject to modify object metadata related to client-side encryption.

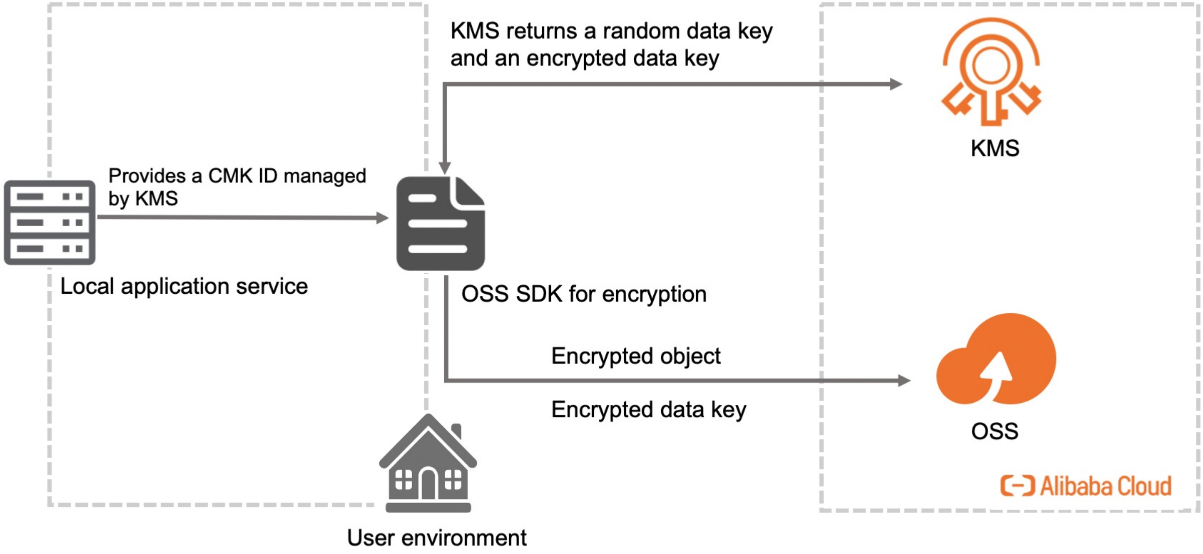
You can use CMKs managed in one of the following ways:

- Use KMS-managed CMKs
- Use customer-managed CMK

For the complete sample code, visit GitHub.

Use KMS-managed CMKs

If you use KMS-managed CMKs for client-side encryption, you need only to specify the CMK ID when you upload objects instead of providing the client with a data key. The following figure shows the encryption process in detail.


KMS returns a random data key and an encrypted data key
KMS
Provides a CMK ID managed by KMS
Local application service
OSS SDK for encryption
Encrypted object
Encrypted data key
OSS
Alibaba Cloud
User environment

- Encrypt and upload an object
  1. Obtain a data key.

     The client uses a specified CMK ID to request a data key used to encrypt the object from KMS. KMS returns a random data key and an encrypted data key.
  2. Encrypt the object and upload it to OSS.

     The client uses the returned data key to encrypt the object and uploads the encrypted object and encrypted data key to OSS.
- Download and decrypt an object
  1. Download an object.

     The client downloads an encrypted object. The encrypted data key is included in the metadata of the object.
  2. Decrypt the object.

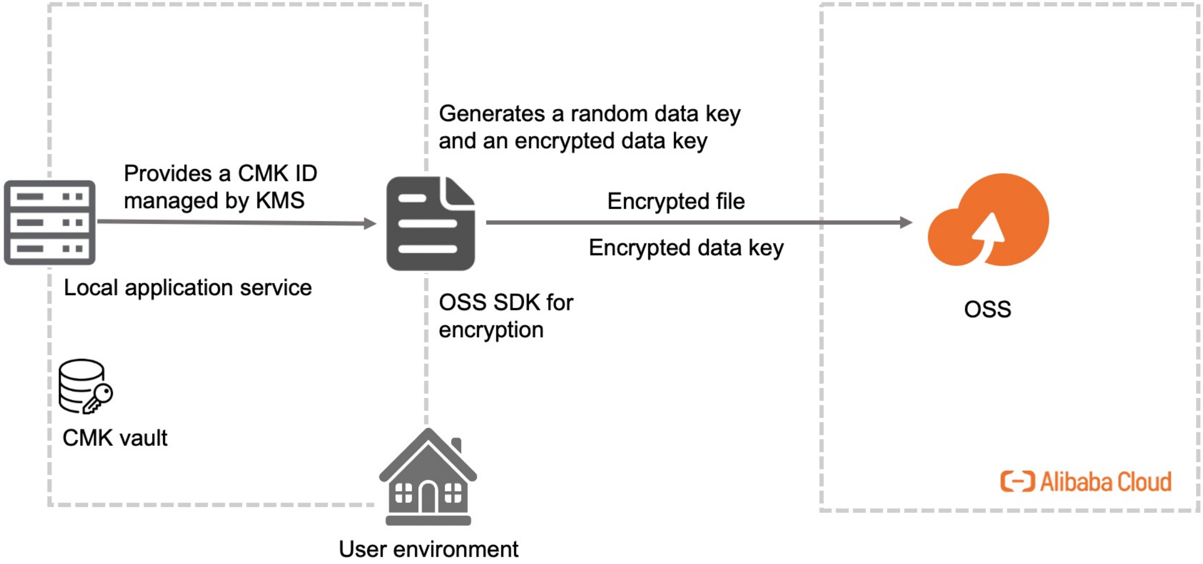
     The client sends the encrypted data key and the corresponding CMK ID to KMS. KMS uses the CMK sent by the client to decrypt the encrypted data key and returns the decrypted data key to the client.

Note

- The client obtains a unique data key for each object to upload.
- To ensure data security, we recommend that you rotate or update the CMK regularly.
- You must maintain the mapping relationship between the CMKs and the encrypted objects.

Use customer-managed CMK

To use this method for client-side encryption, you must generate and manage CMKs by yourself. When you implement client-side encryption on an object to upload, you must upload a symmetric or asymmetric CMK to the client. The following figure shows the encryption process in detail.



- Encrypt and upload an object
    i. You must provide the client with a symmetric or asymmetric CMK.
    ii. The client uses the CMK to generate a one-time-use symmetric data key that is used only to encrypt the current object to upload. The client generates a random and unique data key for each object to upload.
    iii. The client uses the data key to encrypt the object to upload and uses the CMK to encrypt the data key.
    iv. The encrypted data key is included in the metadata of the uploaded object.
- Download and decrypt an object
    i. The client downloads an encrypted object. The encrypted data key is included in the metadata of the object.
    ii. By using the materials in the object metadata, the client determines the CMK used to generate the data key and uses this CMK to decrypt the encrypted data key. Then, the client uses the decrypted data key to decrypt the object.

Notice

- CMKs and unencrypted data are not sent to OSS. Therefore, keep your CMKs secure. If a CMK is lost, objects encrypted by using the data keys generated by using this CMK cannot be decrypted.
- Data keys are randomly generated by the client.

7.4. Versioning

7.4.1. Overview

Object Storage Service (OSS) allows you to configure versioning to protect data at the bucket level. After you enable versioning for a bucket, data that is overwritten or deleted in the bucket is saved as a previous version. After you configure versioning for a bucket, you can recover objects in the bucket to a previous version to protect your data from being accidentally overwritten or deleted.

Usage notes

When you use versioning, take note of the following items:

- Billing

  If you enable versioning for a bucket, you are charged for the storage of previous versions of objects in the bucket. To prevent unnecessary storage costs, we recommend that you delete the previous versions of objects that you no longer need. If you download a previous version of an object or recover the previous version to the current version, fees are incurred by the requests and traffic. For more information, see Billable items and billing methods.

- Required permissions

  Only the bucket owner or Resource Access Management (RAM) users that have the PutBucketVersioning permission can configure versioning for a bucket.

- Feature conflicts
    - A bucket cannot have versioning and retention policies configured at the same time.
    - If versioning is enabled for a bucket, the `x-oss-forbid-overwrite` request header that is specified when an object is uploaded to the bucket does not take effect. For more information, see Request headers.

Implementation methods

The following table describes the methods that you can use to configure versioning for a bucket.

Implementation method	Description
Console	A user-friendly and intuitive web application

Implementation method	Description
ossutil	A high-performance command-line tool
Java SDK	SDK demos for various programming languages
Python SDK	
C++ SDK	
Go SDK	
.NET	
Node.js	

Versioning status

A bucket can be in one of the following versioning states: disabled, enabled, and suspended.

- By default, the versioning status of a bucket is disabled. After versioning is enabled for a bucket, the versioning status of the bucket cannot be set back to disabled. However, you can suspend versioning for a bucket that has versioning enabled.
- When an object is uploaded to a bucket for which versioning is enabled, OSS generates a random string as the globally unique version ID of the object. For more information about how to perform operations on objects in a versioned bucket, see Manage objects in a versioning-enabled bucket.
- When an object is uploaded to a bucket for which versioning is suspended, OSS generates a string null as the version ID of the object. For more information about how to perform operations on objects in a bucket for which versioning is suspended, see Manage objects in a versioning-suspended bucket.

Note In a versioned bucket, all versions of an object are stored, which consumes storage space and incurs storage fees. To reduce storage costs, we recommend that you configure lifecycle rules based on scenarios to delete unnecessary previous versions or convert the storage class of current or previous object versions to Infrequent Access (IA) or Archive. For more information, see Configure lifecycle rules to manage object versions.

Scenarios

To ensure data security, we recommend that you configure versioning in the following scenarios:

- Recover deleted data

  OSS does not provide the recycle bin feature. You can configure versioning to recover deleted data.

- Recover overwritten data

  Online collaborative documents and documents stored in online storage are frequently modified. In online office scenarios, a large number of temporary versions are generated when objects are edited. You can configure versioning to recover the data of a specified object at a point in time.

Data protection

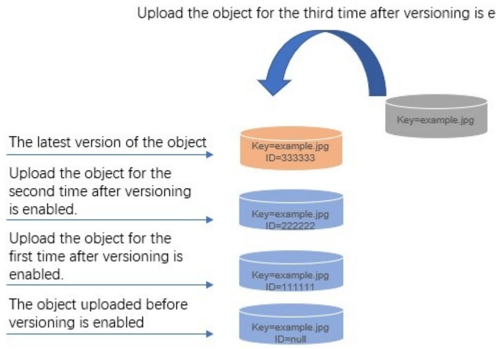
The following table describes how OSS processes deleted and overwritten data in buckets with different versioning states to help you understand the data protection mechanism of versioning.

Versioning status	Data overwritten	Object deletion
Disabled	The existing object is overwritten and cannot be recovered. Only the current object version can be accessed.	The object is deleted and cannot be accessed.
Enabled	A new version with a unique ID is generated for the object. The existing object is stored as a previous version.	A delete marker with a globally unique version ID is added to the object as the current version. The existing object is stored as a previous version.
Suspended	A new version with the version ID null is generated for the object. If a previous version or delete marker whose version ID is null already exists, the previous version or delete marker is overwritten. Other objects or delete markers whose version IDs are not null are not affected.	A delete marker with the version ID null is added for the object. If a previous version or delete marker whose version ID is null already exists, the previous version or delete marker is overwritten by the new delete marker. Other objects or delete markers whose version IDs are not null are not affected.

The following examples use figures to describe how OSS processes data when an object with the same name as that of an existing object is uploaded to or an object is deleted from a bucket for which versioning is enabled or suspended. For ease of viewing, all version IDs in the figures are in the simple format.

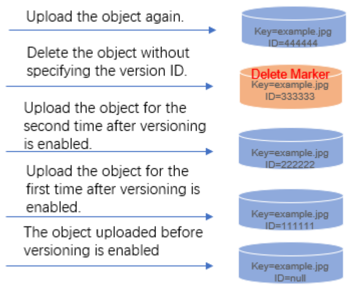
- Overwrite an object in a versioned bucket

  When you upload an object repeatedly to a versioned bucket, the object is overwritten multiple times. A version with a unique version ID is generated for the object each time when the object is overwritten.


Upload the object for the third time after versioning is enabled.
Key=example.jpg
The latest version of the object
Key=example.jpg ID=333333
Upload the object for the second time after versioning is enabled.
Key=example.jpg ID=222222
Upload the object for the first time after versioning is enabled.
Key=example.jpg ID=111111
The object uploaded before versioning is enabled
Key=example.jpg ID=null

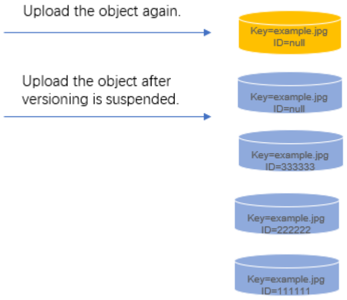
- Delete an object from a versioned bucket

  When you delete an object from a versioned bucket, the previous versions of the object is not deleted and a delete marker is added to the object as the current version to indicate that the object is deleted. If you upload an object with the same name after the delete marker is added, a new version with a unique version ID is added as the current version.


Upload the object again.
Key=example.jpg ID=444444
Delete the object without specifying the version ID.
Delete Marker
Key=example.jpg ID=333333
Upload the object for the second time after versioning is enabled.
Key=example.jpg ID=222222
Upload the object for the first time after versioning is enabled.
Key=example.jpg ID=111111
The object uploaded before versioning is enabled
Key=example.jpg ID=null

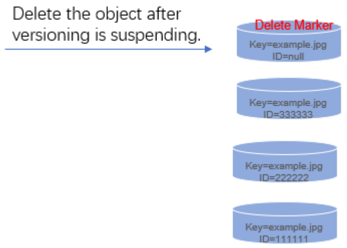
- Overwrite an object in a bucket for which versioning is suspended

  When you upload an object to a bucket for which versioning is suspended, a new version whose version ID is null is added and the previous versions of the object are retained. If you upload the object with the same name again, a new version whose version ID is null overwrites the current version whose version ID is null.


Upload the object again.
Key=example.jpg ID=null
Upload the object after versioning is suspended.
Key=example.jpg ID=null
Key=example.jpg ID=333333
Key=example.jpg ID=222222
Key=example.jpg ID=111111

- Delete an object from a bucket for which versioning is suspended

  When you delete an object from a bucket for which versioning is suspended, the previous versions of the object are not deleted and a delete marker is added to the object as the current version to indicate that the object is deleted.


Delete the object after versioning is suspending.
Delete Marker
Key=example.jpg ID=null
Key=example.jpg ID=333333
Key=example.jpg ID=222222
Key=example.jpg ID=111111

Therefore, deleted and overwritten data is stored as previous versions in a bucket whose versioning status is enabled. After you configure versioning for a bucket, you can recover objects in the bucket to a previous version to protect your data from being accidentally overwritten or deleted.

7.4.2. Manage objects in a versioning-enabled bucket

When versioning is enabled for a bucket, Object Storage Service (OSS) generates a unique ID for each version of all objects in the bucket. The content and access control list (ACL) of existing objects in the bucket remain unchanged. Versioning prevents your data from being accidentally overwritten or deleted and allows you to query or recover previous versions of objects.

Usage notes

Take note of the following items when you perform the following operations in versioned buckets: upload objects, list objects, download objects, delete objects, and recover objects.

- When versioning is enabled for a bucket, a current version and its previous versions are stored for each object in the bucket.
- The version ID of an object that is uploaded before versioning is enabled is set to null.
- For ease of viewing, all version IDs in the following figures are in the simple format.

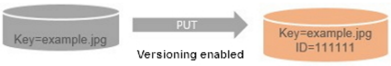
For more information about versioning, see Overview.

Upload objects

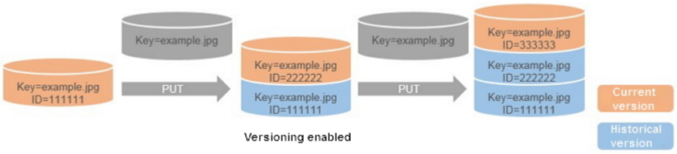
When you upload an object to a versioned bucket, OSS generates a unique version ID for the object.

> Note OSS generates unique version IDs for objects uploaded by using PutObject, PostObject, CopyObject, and MultipartUpload.

In the following figure, when you use the PutObject operation to upload an object whose key is example.jpg, OSS generates a unique version ID of 111111 for the object.



When you use the PutObject operation to upload an object that has the same key as the existing object example.jpg, OSS generates a new version with a unique version ID of 222222 for the object and stores the new version as the current version of the object. Version 111111 is stored as a previous version. When you use the PutObject operation to upload an object that has the same key again, OSS generates a new version with a unique version ID of 333333 for the object and stores the new version as the current version of the object. In the following figure, versions 111111 and 222222 are stored as previous versions.



You can use the cp command provided by ossutil and the following SDKs to upload objects to versioned buckets: OSS SDK for Java, OSS SDK for PHP, OSS SDK for Node.js, OSS SDK for Python,OSS SDK for .NET, OSS SDK for Go, and OSS SDK for C++.

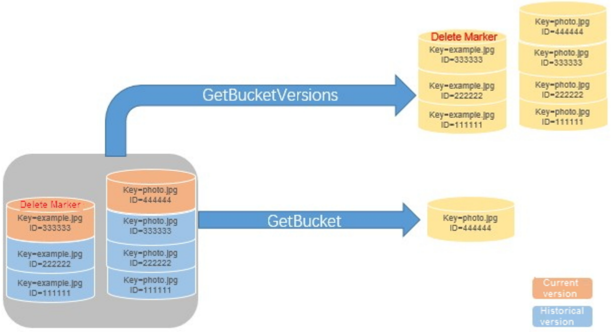
List objects

You can call the GetBucketVersions (ListObjectVersions) operation to obtain the information of all object versions and delete markers in a versioned bucket.

- Unlike GetBucketVersions (ListObjectVersions), the GetBucket (ListObject) operation returns only the current object versions that are not delete markers in a bucket.
- Up to 1,000 object versions can be returned for a single GetBucketVersions (ListObjectVersions) request. You can send multiple GetBucketVersions (ListObjectVersions) requests to obtain all object versions in a versioned bucket.

  For example, if a bucket contains two objects whose names are example.jpg and photo.jpg. The example.jpg object has 900 versions. The photo.jpg object has 500 versions. If you send a GetBucketVersions (ListObjectVersions) request, the 900 versions of the example.jpg object and 100 versions of the photo.jpg object are returned. Versions are returned in alphabetical order of object keys first and then in the order of time when the versions are generated.

In the following figure, all object versions including delete markers are returned when the GetBucketVersions (ListObjectVersions) operation is called. Only current object versions that are not delete markers are returned when the GetBucket(ListObject) operation is called. Therefore, only the current version of the photo.jpg, whose version ID is 444444, is returned.

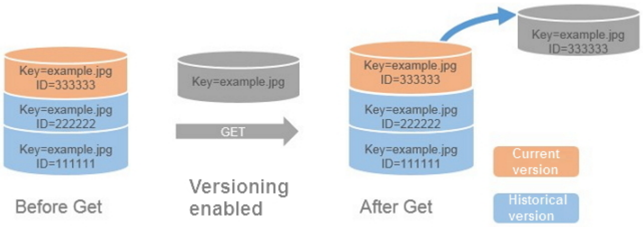


You can use the ls command provided by ossutil and the following OSS SDKs to list objects in a versioned bucket: OSS SDK for Java, OSS SDK for Node.js, OSS SDK for Python, and OSS SDK for Go.

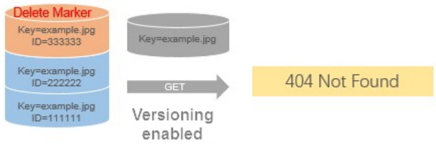
Download objects

You can download the current or a specified version of an object from a versioned bucket.

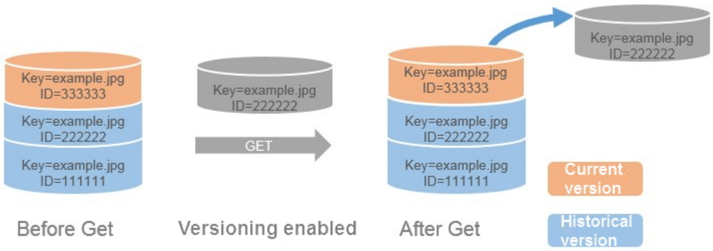
By default, the current version of an object is returned if a GetObject request in which no version ID is specified is sent to download the object. In the following figure, the current version of the object, whose version ID is 333333, is returned.



If the current version of the object to download is a delete marker, 404 Not Found is returned for the GetObject request.


Delete Marker
Key=example.jpg ID=333333
Key=example.jpg ID=222222
Key=example.jpg ID=111111
Key=example.jpg
GET
Versioning enabled
404 Not Found

To download the specified version of an object, you must specify the version ID in the GetObject request. In the following figure, a request in which a version ID is specified is sent to download the object version whose ID is 222222.


Key=example.jpg ID=333333
Key=example.jpg ID=222222
Key=example.jpg ID=111111
Before Get
Key=example.jpg ID=222222
GET
Versioning enabled
Key=example.jpg ID=333333
Key=example.jpg ID=222222
Key=example.jpg ID=111111
After Get
Key=example.jpg ID=222222
Current version
Historical version

You can use the cp command provided by ossutil and the following OSS SDKs to download objects from a versioned bucket: OSS SDK for Java, OSS SDK for PHP, OSS SDK for Node.js, OSS SDK for Python,OSS SDK for .NET, OSS SDK for Go, and OSS SDK for C++.

Delete objects

You can specify an object version ID in the DeleteObject request or configure lifecycle rules to permanently delete a version of an object in a versioned bucket. If you do not specify an object version ID in the DeleteObject request, OSS adds a delete marker as the current version of the object.

Note By default, if you do not specify a version ID in the DeleteObject request, the current version and previous versions of the object are not deleted.

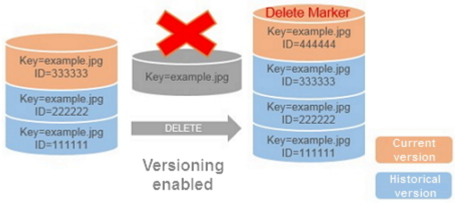
In addition, you can configure the Expiration element in lifecycle rules to delete the expired current version of objects in a versioned bucket. You can also configure the NoncurrentVersionExpiration element in lifecycle rules to permanently delete the expired previous versions of objects in a versioned bucket. The two elements have the following differences:

- The Expiration element is specified in lifecycle rules to delete the expired current version of objects. When an object is deleted based on a lifecycle rule with Expiration configured, OSS stores the current version of the object as a previous version and adds a delete marker as the new current version of the object.
- The NoncurrentVersionExpiration element is specified in lifecycle rules to delete expired previous versions of objects. A previous version deleted based on a lifecycle with NoncurrentVersionExpiration configured is permanently deleted and cannot be recovered.

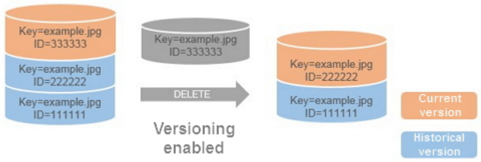
For more information about how to use versioning with lifecycle rules, see Configuration elements.

The following examples describe how OSS deletes an object when the version ID is specified and not specified in the DeleteObject request.

- If the version ID is not specified in the DeleteObject request, OSS adds a delete marker as the current version of the object to delete. The delete marker has a unique version ID but does not have data and ACL. In the following figure, a delete marker with the version ID of 444444 is added as the current version.


Key=example.jpg ID=333333
Key=example.jpg ID=222222
Key=example.jpg ID=111111
Key=example.jpg
DELETE
Versioning enabled
Delete Marker
Key=example.jpg ID=444444
Key=example.jpg ID=333333
Key=example.jpg ID=222222
Key=example.jpg ID=111111
Current version
Historical version

- If the version ID is specified in the DeleteObject request, the specified version is permanently deleted. In the following figure, the version whose ID is 333333 is permanently deleted.


Key=example.jpg ID=333333
Key=example.jpg ID=222222
Key=example.jpg ID=111111
Key=example.jpg ID=333333
DELETE
Versioning enabled
Key=example.jpg ID=222222
Key=example.jpg ID=111111
Current version
Historical version

You can use the rm command provided by ossutil and the following OSS SDKs to delete objects from a versioned bucket: OSS SDK for Java, OSS SDK for PHP, OSS SDK for Node.js, OSS SDK for Python,OSS SDK for .NET, OSS SDK for Go, and OSS SDK for C++.

Recover objects

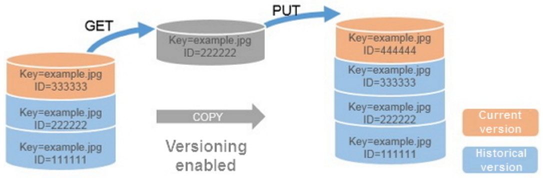
When versioning is enabled for a bucket, all versions of objects in the bucket are preserved. You can recover the specified previous version of an object as the current version.

You can recover a previous version of an object as the current version by using the following two methods:

- Use CopyObject to copy the previous version to the same bucket

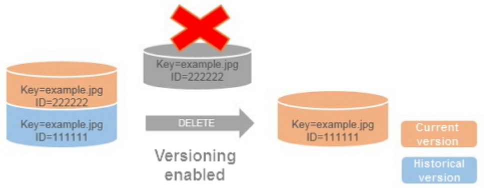
  The copied previous version becomes the current version of the object and all versions of the object are preserved.

In the following figure, a previous version whose ID is 222222 is copied to the same bucket. OSS adds the copied version as a new version whose ID is 444444. The new version becomes the current version of the object. Therefore, the object has two versions that have the same content: 222222 and 444444, in which version 222222 is a previous version and version 444444 is the current version.


GET
PUT
Key=example.jpg ID=333333
Key=example.jpg ID=222222
Key=example.jpg ID=111111
Key=example.jpg ID=222222
COPY
Versioning enabled
Key=example.jpg ID=444444
Key=example.jpg ID=333333
Key=example.jpg ID=222222
Key=example.jpg ID=111111
Current version
Historical version

- Permanently delete the current version of the object

  In the following figure, the current version whose ID is 222222 is permanently deleted by a DeleteObject request in which the version ID is specified. The most recent previous version whose ID is 111111 becomes the current version of the object.


Key=example.jpg ID=222222
Key=example.jpg ID=111111
Key=example.jpg ID=222222
DELETE
Versioning enabled
Key=example.jpg ID=111111
Current version
Historical version

Notice We recommend that you use CopyObject to recover a previous version as the current version because the current version cannot be recovered after it is permanently deleted.

You can use the cp command provided by ossutil and the following OSS SDKs to recover previous versions in a versioned bucket: OSS SDK for Java, OSS SDK for PHP, OSS SDK for Node.js, OSS SDK for Python,OSS SDK for .NET, OSS SDK for Go, and OSS SDK for C++.

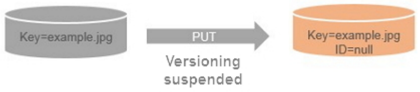
7.4.3. Manage objects in a versioning-suspended bucket

Object Storage Service (OSS) allows you to suspend versioning for a bucket so that no additional versions are generated for the objects in the bucket. When versioning is suspended for a bucket, you can upload objects to the bucket and download or delete previous versions of objects in the bucket by specifying the version IDs.

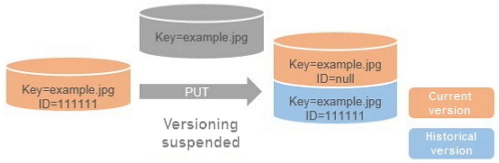
Upload objects

When you upload an object to a versioning-suspended bucket, OSS stores the uploaded object as a version whose ID is null. Each object in a versioning-suspended bucket has only one version whose ID is null.

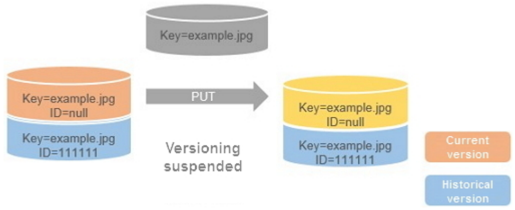
- In the following figure, when you use the PutObject operation to upload an object to a versioning-suspended bucket, OSS stores the uploaded object as a version whose ID is null.



- In the following figure, an object named example.jpg in a versioning-suspended bucket has a version whose ID is 111111. When you use the PutObject operation to upload an object that has the same name as the existing object, OSS stores the uploaded object as the current version of example.jpg and assigns the ID null for the current version. Version 111111 of the object is saved as a previous version.



- In the following figure, an object named example.jpg in a versioning-suspended bucket has a version whose ID is null. When you use the PutObject operation to upload an object that has the same name, the version whose ID is null is overwritten.



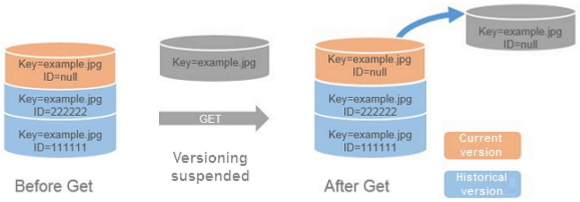
You can use the cp command provided by ossutil and the following OSS SDKs to upload objects to a versioning-suspended bucket: OSS SDK for Java, OSS SDK for PHP, OSS SDK for Node.js, OSS SDK for Python,OSS SDK for .NET, OSS SDK for Go, and OSS SDK for C++.

Download objects

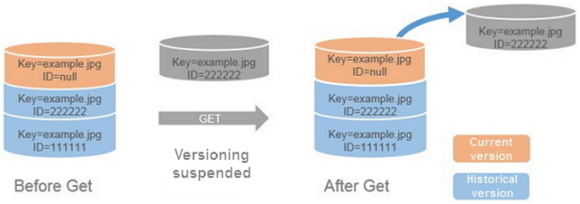
OSS allows you to download the current or specified version of an object from a versioning-suspended bucket.

The following examples describe how an object is downloaded from a versioning-suspended bucket when a version ID is specified and not specified in the GetObject request:

- By default, if no version ID is specified in the request, the current version of the object is returned. In the following figure, the current version of the object, whose ID is null, is returned.



- If you specify a version ID in the GetObject request, the specified version of the object is returned. In the following figure, the version whose ID is 222222 is returned.

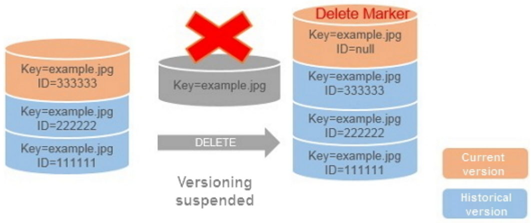

Key=example.jpg ID=null
Key=example.jpg ID=222222
Key=example.jpg ID=111111
Before Get
Key=example.jpg ID=222222
GET
Versioning suspended
Key=example.jpg ID=222222
Key=example.jpg ID=null
Key=example.jpg ID=222222
Key=example.jpg ID=111111
After Get
Current version
Historical version

You can use the cp command provided by ossutil and the following OSS SDKs to download objects from a versioning-suspended bucket: OSS SDK for Java, OSS SDK for PHP, OSS SDK for Node.js, OSS SDK for Python,OSS SDK for .NET, OSS SDK for Go, and OSS SDK for C++.

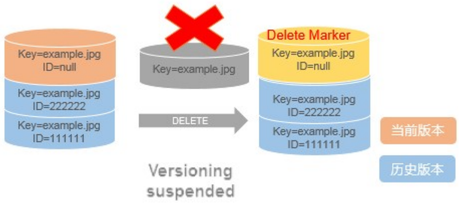
Delete objects

The following examples describe how an object is deleted from a versioning-suspended bucket in different scenarios:

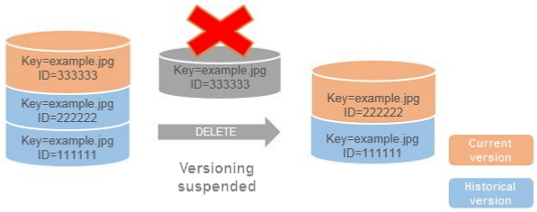
- When you use the DeleteObject operation to delete an object whose current version ID is not null, OSS adds a delete marker whose ID is null as the current version of the object.


Key=example.jpg ID=333333
Key=example.jpg ID=222222
Key=example.jpg ID=111111
Key=example.jpg
DELETE
Versioning suspended
Delete Marker
Key=example.jpg ID=null
Key=example.jpg ID=333333
Key=example.jpg ID=222222
Key=example.jpg ID=111111
Current version
Historical version

- When you use the DeleteObject operation to delete an object whose current version ID is null, OSS adds a delete marker whose ID is null as the current version of the object. An object can have only one version whose ID is null. Therefore, the original current version of the object is overwritten by the newly-added delete marker whose ID is null.



- If you specify a version ID in the DeleteObject request, the specified version of the object is permanently deleted. In the following figure, the version whose ID is 333333 is deleted.


Key=example.jpg ID=333333
Key=example.jpg ID=222222
Key=example.jpg ID=111111
Key=example.jpg ID=333333
DELETE
Versioning suspended
Key=example.jpg ID=222222
Key=example.jpg ID=111111
Current version
Historical version

You can use the rm command provided by ossutil and the following OSS SDKs to delete objects from a versioning-suspended bucket: OSS SDK for Java, OSS SDK for PHP, OSS SDK for Node.js, OSS SDK for Python,OSS SDK for .NET, OSS SDK for Go, and OSS SDK for C++.

7.4.4. Delete marker

A delete marker is a placeholder specified by the DeleteObject request. It is used to indicate that an object in a bucket that has versioning enabled or suspended is deleted.

Comparison with objects

A delete marker has an object name (or key) and version ID like an object, but differs from an object in the following aspects:

- A delete marker does not store data.
- A delete marker has no access control list (ACL) configured.
- If you initiate the GetObject request to a delete marker, no content can be returned because a delete marker does not store data. If you initiate the GetObject request to an object with a delete marker as its current version, 404 and the corresponding error message are returned.

- Users who have only the `oss:DeleteObjectVersion` permission can perform only delete operations on delete markers.

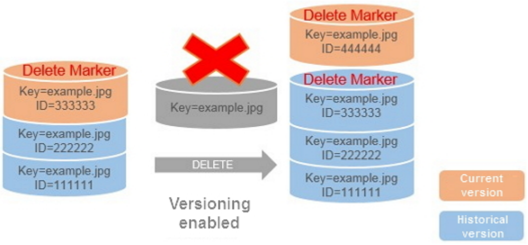
When you initiate the DeleteObject request to an object in a bucket that has versioning enabled or suspended, a delete marker is generated. If you do not specify the version ID of the object that you want to delete in the DeleteObject request, Object Storage Service (OSS) assigns a delete marker to the object as its current version instead of deleting the object.

Note You cannot delete an object stored in a bucket that has versioning enabled. However, you can consider an object with a delete marker as a deleted object.

Remove delete markers

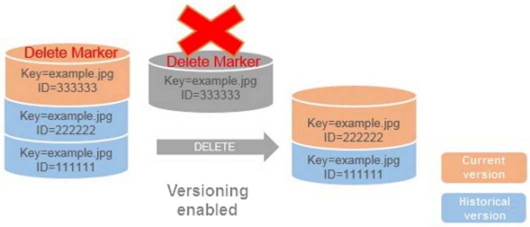
The following section describes how to remove delete markers for a versioning-enabled bucket:

If you do not specify the version ID of a delete marker of an object in the DeleteObject request, OSS does not remove the delete marker but assigns a new delete marker to the object as the current version of the object. The following figure shows that an object can have multiple delete markers.



Note In a versioning-enabled bucket, an object can have multiple delete markers, and a delete marker corresponds to a unique version ID.

To permanently remove a delete marker, you must include its version ID in the DeleteObject request. The following figure shows how the DeleteObject request permanently removes a delete marker with the version ID of 333333. Then, version 222222 becomes the current version of the object.



You can use the following OSS SDKs for different programming languages to delete a specific version of an object and the delete marker of the object: Java SDK, Python SDK, PHP SDK, Node.js SDK, .NET SDK, Go SDK, and C++ SDK.

7.4.5. FAQ

This topic describes problems you may encounter when you use the versioning feature provided by OSS and provides troubleshooting methods and solutions for the problems.

Storage cost

If you enable versioning for a bucket, you are charged for the storage of the current versions and previous versions of all objects in the bucket. The following example is used to describe the storage costs incurred in a month (30 days) for a versioning-enabled bucket.

- On the first day of the month, you call PutObject to upload an object 4 GB in size to a Standard (LRS) bucket.
- On the 16th day of the month, you call PutObject to upload an object with the same name as the previously uploaded object but 5 GB in size to the same bucket.

After you upload the object on the 16th day, the original object you upload on the first day is retained as a previous version. The object you upload on the first day is stored in the bucket for 30 days in total. The object you uploaded on the 16th day is stored in the bucket for 15 days as the current version.

Therefore, the storage cost for the object in the month can be calculated based on the pay-as-you-go billing method by using the following formula: 4 GB x 0.02 USD/GB/Month + (5 GB x 0.02 USD/GB/Month)/2 = 0.13 USD.

For more information about the storage cost, see Storage fees.

Low response speed

Question: Why does the response speed decrease significantly when the GetBucket (ListObjects) operation is called to list current object versions in a versioned bucket?

Cause: One or more objects in your bucket have a large number of previous versions or expired delete markers.

Troubleshooting:

- Call the GetBucketVersions (ListObjectVersions) operation to check whether the objects in your bucket have a large number of previous versions. For more information, see GetBucketVersions(ListObjectVersions).
- Use the bucket inventory feature to view the information about the objects in your bucket and check whether the objects have previous versions or expired delete markers. For more information, see Bucket inventory.

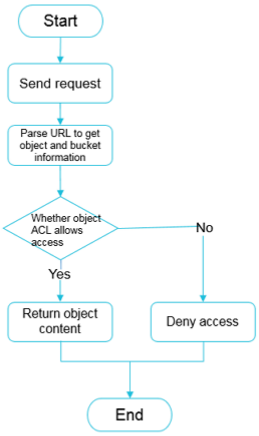
Solution: Configure lifecycle rules for your bucket and specify the NonCurrentVersionExpiration and ExpiredObjectDeleteMarker operations in the rules to delete expired previous object versions and delete markers. For more information, see Configuration elements.

7.5. Signature

7.5.1. OSS request process

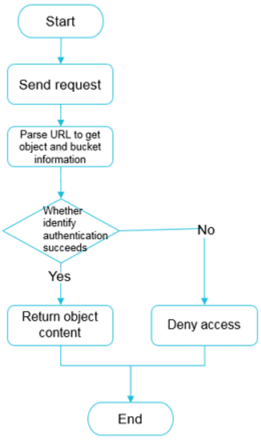
HTTP requests sent to Object Storage Service (OSS) are divided into two types based on whether authentication information is included: requests with authentication information and anonymous requests without authentication information. Compared with anonymous requests that do not include authentication information, a request with authentication information includes signature information in the request header or the request URL, which complies with OSS API Reference.

Use anonymous requests to access OSS


Start
Send request
Parse URL to get object and bucket information
Whether object ACL allows access
No
Yes
Return object content
Deny access
End

1. A user request is sent to the HTTP server of OSS.
2. OSS parses the URL of the request to obtain the requested bucket and object.
3. OSS checks whether the access control list (ACL) of the object is set to allow anonymous access.
   - If anonymous access is allowed, the object is returned to the user.
   - If anonymous access is not allowed, the request is denied.

Use requests with authentication information to access OSS


Start
Send request
Parse URL to get object and bucket information
Whether identify authentication succeeds
No
Yes
Return object content
Deny access
End

1. A user request is sent to the HTTP server of OSS.
2. OSS parses the URL of the request to obtain the requested bucket and object.
3. OSS obtains the identity information about the requester for authentication based on the AccessKey ID of the requester.
   - If the identity information is not obtained, the request is denied.
   - If the identity information is obtained, but the requester is not allowed to access the requested object, the request is denied.
   - If the identity information is obtained, but the signature calculated based on the HTTP parameters in the request does not match the signature contained in the request, the request is denied.
   - If the authentication succeeds, the object is returned to the user.

AccessKey pair types

Currently, the following three types of OSS AccessKey pairs are used to access OSS:

- The AccessKey pair of an Alibaba Cloud account

  The AccessKey pair of an Alibaba Cloud account indicates the AccessKey pair of the bucket owner. The AccessKey pair of an Alibaba Cloud account has full access to all resources in the account. Each Alibaba Cloud account can have up to five AccessKey pairs (AccessKey ID and AccessKey secret), and each AccessKey pair can be in either an active or inactive state.

  You can request to add or delete your AccessKey pairs in the Alibaba Cloud Management Console.

  Each AccessKey pair can be in either an active or inactive state.

- Active: indicates that the AccessKey pair can be used for authentication.
- Inactive: indicates that the AccessKey pair cannot be used for authentication.

> **Notice** We recommend that you do not use the AccessKey pair of your Alibaba Cloud account to manage your OSS resources for data security reasons. However, you can create an AccessKey pair for a Resource Access Management (RAM) user and grant permissions to the RAM user.

- The AccessKey pair of a RAM user

  Resource Access Management (RAM) is a service provided by Alibaba Cloud to manage access permissions on resources. The AccessKey pairs of RAM users are authorized in the RAM console. They can be used to access bucket resources only based on the rules that are defined in RAM. You can use RAM to manage users such as employees, systems, and applications, and control the permissions of users to access your resources. For example, you can create a RAM policy to grant users read-only permissions on one of your buckets. A RAM user belongs to the Alibaba Cloud account under which the RAM user was created. In addition, the RAM user does not actually own resources. All resources belong to the corresponding Alibaba Cloud account.

- The AccessKey pair of an STS account

  Security Token Service (STS) is an Alibaba Cloud service that provides temporary access credentials. An STS temporary AccessKey pair is issued by STS. The AccessKey pair can be used only to access OSS buckets in accordance with the rules defined in STS.

Authentication implementation methods

Currently, authentication is implemented in the following three methods:

- AccessKey pair authentication
- RAM authentication
- STS authentication

When a user sends a request to OSS as an individual identity, authentication is performed in the following procedure:

1. A signature string is generated in the format specified by OSS based on the request.
2. Use your AccessKey secret to encrypt the signature string so that a verification code is generated.
3. After OSS receives the request, OSS finds the AccessKey secret based on your AccessKey ID, and uses the AccessKey secret to extract the signature string and verification code.
   - If the verification code calculated by OSS is identical to the provided one, OSS considers the request valid.
   - Otherwise, OSS denies the request and returns the HTTP 403 error code.

How to access OSS when you use requests with authentication information

- Use the OSS console to access OSS: The authentication process runs in the background, and users do not need to worry about authentication configurations when they access OSS in the console. For more information, see Download objects.
- Use OSS SDKs to access OSS: OSS provides SDKs for multiple programming languages in which the signature algorithm is implemented. Therefore, users need only to input the AccessKey pair information to access OSS by using SDKs. For more information, see:
  - Java SDK
  - Python SDK
  - Go SDK
  - C++ SDK
  - PHP SDK
  - C SDK
  - .NET SDK
  - Android SDK
  - iOS SDK
  - Node.js
  - Browser.js
- Use OSS API operations to access OSS: To encapsulate and call RESTful API operations by using a specific programming language, you must implement a signature algorithm to calculate the signature. For more information, see Add signatures to the Authorization header and Add signatures to a URL in OSS API Reference.

7.5.2. Include signatures in the Authorization header

You can include the `Authorization` header in an HTTP request to carry signature information and indicate that the requester is authorized.

Sign requests when you use OSS SDKs

Requests initiated by using Object Storage Service (OSS) SDKs are automatically signed. You do not need to manually add signatures to requests. For more information about how requests are signed when you use OSS SDKs for different programming languages, see the sample code of OSS SDKs. The following table describes the sample code used to sign requests initiated by using OSS SDKs for different programming languages.

SDK	Sample code
Java SDK	OSSRequestSigner.java
Python SDK	auth.py
.Net SDK	OssRequestSigner.cs
PHP SDK	OssClient.php
C SDK	oss_auth.c
JavaScript SDK	client.js
Go SDK	auth.go

SDK	Sample code
Ruby SDK	util.rb
iOS SDK	OSSModel.m
Android SDK	OSSUtils.java

Calculation of the Authorization header

- Calculation method

```
Authorization = "OSS " + AccessKeyId + ":" + Signature
Signature = base64(hmac-sha1(AccessKeySecret,
            VERB + "\n"
            + Content-MD5 + "\n"
            + Content-Type + "\n"
            + Date + "\n"
            + CanonicalizedOSSHeaders
            + CanonicalizedResource))
```

- Parameters

Parameter	Required	Example	Description
AccessKeySecret	Yes	OtxrzxIsfpFjA7Sw******8Bw21TLhquhboDYROV	The AccessKey secret that is used to sign the request.
VERB	Yes	PUT	The method of the HTTP request, such as PUT, GET, POST, HEAD, or DELETE.
\n	No	\n	A line feed.
Content-MD5	No	eB5eJF1ptWaXm4bijSPyxw==	The Content-MD5 is the MD5 hash of requested content. The message content that excludes the header is calculated to obtain an MD5 hash, which is a 128-bit number. This number is encoded with Base64 into a Content-MD5 value. For more information, visit RFC 2616 Content-MD5. Optional. This request header can be used to check the message validity. The message content is valid if the received message content is the same as the content that is sent. For more information about how to calculate the value of Content-MD5, see Calculation of Content-MD5.
Content-Type	No	application/octet-stream	Optional. The type of the request content.
Date	Yes	Sun, 22 Nov 2015 08:16:38 GMT	Required. The time when this operation is performed. The value of this parameter must be in GMT. **Notice** If the difference between the time specified by the Date header in a request and the time on the server when the request is received is greater than 15 minutes, OSS rejects the request and returns HTTP status code 403.
CanonicalizedOSSHeaders	No	x-oss-meta-a:a\nx-oss-meta-b:b\nx-oss-meta-c:c\n	Optional. The HTTP headers that are prefixed with x-oss-. The HTTP headers are sorted in alphabetical order. ◦ CanonicalizedOSSHeaders can be left empty. In this case, the `\n` delimiter at the end can be removed. ◦ If CanonicalizedOSSHeaders includes only one header, the `\n` delimiter must be added at the end of the header. Example: `x-oss-meta-a\n` . ◦ If CanonicalizedOSSHeaders includes multiple headers, you must add the `\n` delimiter to each header. Example: `x-oss-meta-a:a\nx-oss-meta-b:b\nx-oss-meta-c:c\n` . For more information about how to construct CanonicalizedOSSHeaders, see Creation of CanonicalizedOSSHeaders.
CanonicalizedResource	Yes	/examplebucket/	Required. The OSS resource you want to access. For more information about how to construct CanonicalizedResource, see Creation of CanonicalizedResource.

- Signature examples

Request	Formula	Signature string
PUT /nelson HTTP/1.0 Content-MD5: eB5eJF1ptWaXm4bijSPyxw== Content-Type: text/html Date: Thu, 17 Nov 2005 18:49:58 GMT Host: oss-example.oss-cn-hangzhou.aliyuncs.com x-oss-meta-author: foo@example.com x-oss-meta-magic: abracadabra	Signature = base64(hmac-sha1(AccessKeySecret,VERB + "\n" + Content-MD5 + "\n"+ Content-Type + "\n" + Date + "\n" + CanonicalizedOSSHeaders+ CanonicalizedResource))	“PUT\n eB5eJF1ptWaXm4bijSPyxw==\n text/html\n Thu, 17 Nov 2005 18:49:58 GMT\n x-oss-meta-magic:abracadabra\nx-oss-meta-author:foo@example.com\n/oss-example/nelson

If AccessKey ID is set to "44CF959******252F707" and AccessKey Secret is set to "OtxrzxIsfpFjA7Sw******8Bw21TLhquhboDYROV", you can run the following Python code to calculate the signature:

```
import hmac
import hashlib
h = hmac.new(oss2.to_bytes("OtxrzxIsfpFjA7Sw******8Bw21TLhquhboDYROV"),
             oss2.to_bytes("PUT\nODBGOERFMDMzQTczRUY3NUE3NzA5QzdFNUYzMDQxNEM=\ntext/html\nThu, 17 Nov 2005 18:49:58 GMT\nx-oss-meta-magic:abracadab
ra\nx-oss-meta-author:foo@example.com\n/oss-example/nelson"), hashlib.sha1)
signature = oss2.utils.b64encode_as_string(h.digest())
print("Signature: %s" % signature)
```

The calculated signature is 26NBxoKd******Dv6inkoDft/yA=. Therefore, the value of the Authorization header is OSS 44CF95900***BF252F707:26NBxoKd******Dv6inkoDft/yA=, which is in the following format: "OSS"+ AccessKey ID + ":" + Signature. The following example shows the final request that includes the Authorization header:

```
PUT /nelson HTTP/1.0
Authorization:OSS 44CF95900***BF252F707:26NBxoKd******Dv6inkoDft/yA=
Content-Md5: eB5eJF1ptWaXm4bijSPyxw==
Content-Type: text/html
Date: Thu, 17 Nov 2005 18:49:58 GMT
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
x-oss-meta-author: foo@example.com
x-oss-meta-magic: abracadabra
```

Detail analysis:

- If the imported AccessKey ID does not exist or is not activated, 403 Forbidden and InvalidAccessKeyId are returned. If the imported AccessKey ID is activated but OSS determines that a signature error occurs in the request, 403 Forbidden is returned with the correct signature string in the response to verify the encryption. You can check whether the signature string is correct based on the response.

  Sample response

  ```
  <?xml version="1.0" ?>
  <Error>
   <Code>
       SignatureDoesNotMatch
   </Code>
   <Message>
       The request signature we calculated does not match the signature you provided. Check your key and signing method.
   </Message>
   <StringToSignBytes>
       47 45 54 0a 0a 0a 57 65 64 2c 20 31 31 20 4d 61 79 20 32 30 31 31 20 30 37 3a 35 39 3a 32 35 20 47 4d 54 0a 2f 75 73 72 65 61 6c 74 65 73 74 3
  f 61 63 6c
   </StringToSignBytes>
   <RequestId>
       1E446260FF9B****
   </RequestId>
   <HostId>
       oss-cn-hangzhou.aliyuncs.***
   </HostId>
   <SignatureProvided>
       y5H7yzPsA/tP4+0tH1HHvPEwUv8=
   </SignatureProvided>
   <StringToSign>
       GET
  Wed, 11 May 2011 07:59:25 GMT
  /oss-example?acl
   </StringToSign>
   <OSSAccessKeyId>
       AKIAIVAKMSMOY7VO****
   </OSSAccessKeyId>
  </Error>
  ```

- If the format of the Authorization value in the request is invalid, 400 Bad Request and InvalidArgument are returned.
- The date and time specified in all OSS requests must be in GMT defined in HTTP/1.1, in which date is in the following format:

  ```
  date1 = 2DIGIT SP month SP 4DIGIT; day month year (e.g., 02 Jun 1982)
  ```

  > **Note** In the preceding date format, "day" uses two digits. Therefore, "Jun 2", "2 Jun 1982", and "2-Jun-1982" are all invalid date formats.

  - If the Date header is not specified or is in an invalid format in a signed request, 403 Forbidden is returned with AccessDenied.
  - If the difference between the time specified by the Date header in a request and the time on the server when the request is received is greater than 15 minutes, 403 Forbidden and RequestTimeTooSkewed are returned.

Creation of CanonicalizedOSSHeaders

All HTTP headers prefixed with x-oss- are called CanonicalizedOSSHeaders. You can perform the following steps to create CanonicalizedOSSHeaders:

1. Convert the names of all HTTP request headers prefixed with x-oss- into lowercase letters. For example, convert `X-OSS-Meta-Name: TaoBao` into `x-oss-meta-name: TaoBao` .
2. If the request is sent by using an access credential obtained from STS, you must add the obtained security-token value to the signature string in the `x-oss-security-token:security-token` format.

   > Note For more information about how to configure Security Token Service (STS), see Use a temporary credential provided by STS to access OSS in OSS Developer Guide. You can call the AssumeRole operation or use STS SDKs for various programming languages to obtain a temporary access credential. For more information, see STS SDK overview. The temporary access credential contains a security token and a temporary AccessKey pair. The AccessKey pair consists of an AccessKey ID and an AccessKey secret.

3. Sort all HTTP request headers that are converted in Step 1 in alphabetical order.
4. Delete all spaces on each side of the delimiter between each header and value. For example, convert `x-oss-meta-name: TaoBao` into `x-oss-meta-name:TaoBao` .
5. Separate all headers with the `\n` delimiter to create CanonicalizedOSSHeaders.

Creation of CanonicalizedResource

The OSS resources that you send requests to access are called CanonicalizedResource. You can perform the following operations to create CanonicalizedResource:

1. Set CanonicalizedResource to an empty string `""` .
2. Specify the OSS resource you want to access in the following format: `/BucketName/ObjectName` .
   - If the resource you want to access is a bucket instead of an object, set CanonicalizedResource to "/BucketName/".
   - If the resource you want to access is not a bucket or an object, set CanonicalizedResource to "/".
   - If the resource that you want to access contains subresources, sort all subresources in alphabetical order and separate them with the ampersand (`&`) delimiter. Add a question mark (`?`) and the subresource string to the end of the CanonicalizedResource string. In this example, the created CanonicalizedResource is in the following format: `/BucketName/ObjectName?acl&uploadId=UploadId` .

     OSS supports the following subresources: acl, uploads, location, cors, logging, website, referer, lifecycle, delete, append, tagging, objectMeta, uploadId, partNumber, security-token, position, img, style, styleName, replication, replicationProgress, replicationLocation, cname, bucketInfo, comp, qos, live, status, vod, startTime, endTime, symlink, x-oss-process, response-content-type, response-content-language, response-expires, response-cache-control, response-content-disposition, and response-content-encoding.

     You can specify the following three types of subresources:

     - Resource identifiers such as acl, append, uploadId, and symlink subresources For more information, see Bucket operations and Object operations.
     - Subresources that specify the response header fields. Example: `response-***` . For more information, see `Request Parameters` in GetObject.
     - Object processing methods such as `x-oss-process` . For more information, see IMG.

Calculation of the signature

- The signature string used to calculate the signature must be encoded in `UTF-8` . A signature string that contains Chinese characters must be encoded in `UTF-8` . The encoded signature string is used together with `AccessKeySecret` to calculate the final signature.
- The HMAC-SHA1 method defined in RFC 2104 is used to calculate the signature. In this method, Key indicates AccessKeySecret.
- `Content-Type` and `Content-MD5` can be left unspecified in a request. However, if OSS needs to verify the signature of the request, the values of these two headers must be replaced by line feeds (`\n`).
- Non-standard HTTP headers prefixed with `x-oss-` must be added to the signature string. Other non-standard HTTP headers are ignored by OSS. For example, the x-oss-meta-magic header in the following example must be included in the signature string.

  > Note Headers prefixed with `x-oss-` in the signature string must comply with the following conventions:
  > - The names of headers must be in lowercase.
  > - The headers must be sorted in alphabetical order.
  > - No space exists before or after the colon (:) that separates each header name and value.
  > - Each header is followed by a line feed (\n). If no header is specified, CanonicalizedOSSHeaders is empty.

Calculation of Content-MD5

The following examples use a string "123456789" to show how to calculate the Content-MD5 value of the request content:

- Correct calculation

  i. Calculate the MD5 hash of the string, which is a 128-bit binary array.

  ii. Encode the binary array (instead of the 32-bit string) in Base64.

  The following Python code provides an example on how to calculate the Content-MD5 value:

  ```
  >>> import base64,hashlib
  >>> hash = hashlib.md5()
  >>> hash.update("0123456789")   // If you use Python 3, change the code to hash.update(b"0123456789").
  >>> base64.b64encode(hash.digest())
  'eB5eJF1ptWaXm4bijSPyxw=='
  ```

  Call hash.digest() to calculate the 128-bit binary array.

  ```
  >>> hash.digest()
  'x\x1e^$]i\xb5f\x97\x9b\x86\xe2\x8d#\xf2\xc7'
  ```

- Incorrect calculation

  > Note A common incorrect operation is to encode the calculated 32-bit string in Base64 to obtain the Content-MD5 value.

```
# Call hash.hexdigest() to obtain a 32-bit plaintext string.
>>> hash.hexdigest()
'781e5e245d69b566979b86e28d23f2c7'
# The following code provides an example on encoding the incorrect MD5 hash in Base64:
>>> base64.b64encode(hash.hexdigest())
'NzgxZTVlMjQ1ZDY5YjU2Njk3OWI4NmUyOGQyM2YyYzc='
```

7.5.3. Add signatures to a URL

In addition to adding signatures to the Authorization header of requests, you can also add signatures to the URL of an Object Storage Service (OSS) resource and share the URL to allow authorized third-party users to access the resource.

Usage notes

- If you use a signed URL to share data, the information about the data can be obtained by all users on the Internet within the validity period of the signed URL. We recommend that you evaluate the risks in advance.
- If a request contains a URL, signatures cannot be added to the Authorization header of the request and the URL at the same time.
- You can add signatures to URLs that are contained in PUT and GET requests.
- You can generate a presigned URL for a PUT request to check whether the content to upload is valid. When you use OSS SDKs to generate a presigned URL for a request, OSS SDKs calculate the MD5 hash of the request body and include the MD5 hash in the presigned URL. The MD5 hash of the uploaded content must be the same as the MD5 hash calculated by OSS SDKs. Otherwise, the request fails. To verify the MD5 hash of the uploaded content, add the Content-MD5 header in the request.

Implementation methods

- Example of a signed URL:

  ```
  http://oss-example.oss-cn-hangzhou.aliyuncs.com/oss-api.pdf?OSSAccessKeyId=nz2pc56s936****&Expires=1141889120&Signature=vjbyPxybdZaNmGa%2ByT272YEAiv****
  ```

  To use a temporary access credential provided by Security Token Service (STS) to generate a signed URL, you must add the `security-token` parameter in the signature. The following example shows a URL that is signed by using an access credential provided by STS:

  ```
  http://oss-example.oss-cn-hangzhou.aliyuncs.com/oss-api.pdf?OSSAccessKeyId=nz2pc56s936****&Expires=1141889120&Signature=vjbyPxybdZaNmGa%2ByT272YEAiv****&security-token=SecurityToken
  ```

- Parameters

<table>
<tr><th>Parameter</th><th>Required</th><th>Description</th></tr>
<tr><td>Expires</td><td>Yes</td><td>The time when the signed URL expires. The value of this parameter is a , which is the number of seconds that have elapsed since January 1, 1970 00:00:00 UTC. If the time when OSS receives the request that contains the URL is later than the value of this parameter that is included in the signature, a request timeout error is returned. For example, the current time is 1141889060. To create a URL that is valid within 60 seconds, you can set the value of this parameter to 1141889120.
Note For security reasons, the validity period of a signed URL that is generated by using the OSS console is 3,600 seconds by default. The maximum value of the validity period is 32,400 seconds.</td></tr>
<tr><td>OSSAccessKeyId</td><td>Yes</td><td>The AccessKey ID used to access OSS.</td></tr>
<tr><td>Signature</td><td>Yes</td><td>The signature information that you want to add to the URL.
The signature information is in the following format:<pre>
Signature = urlencode(base64(hmac-sha1(AccessKeySecret,
          VERB + "\n"
          + CONTENT-MD5 + "\n"
          + CONTENT-TYPE + "\n"
          + EXPIRES + "\n"
          + CanonicalizedOSSHeaders
          + CanonicalizedResource)))
</pre>The headers and algorithm used to calculate the signature that you want to add to a URL are similar to those used to calculate the signature that you want to add to the Authorization header of a request.
When you calculate the signature string that you want to add to a URL, the <code>CONTENT-TYPE</code> , <code>CONTENT-MD5</code> , and <code>CanonicalizedOSSHeaders</code> headers are the same as those used to calculate the signature that you add to the Authorization header. However, you must replace the Date header with Expire in the signature string. You can include the Date header in the request. For more information about the headers, see Include signatures in the Authorization header.
The signature string added to a URL must be URL-encoded. If the value of Signature, Expires, and OSSAccessKeyId is imported multiple times, the first imported value is used.
If a request contains a signed URL, OSS first checks whether the time when the request is received is later than the value of the Expires header, and then verifies the signature.</td></tr>
</table>

Parameter	Required	Description
security-token	No	The security token provided by STS. You must configure this parameter only when you use a temporary access credential to sign the URL. Note For more information about how to set up STS, see Use a temporary access credential provided by STS to access OSS. You can call the AssumeRole operation or use STS SDKs for various programming languages to obtain a temporary access credential. The temporary access credential contains a security token and a temporary AccessKey pair that consists of an AccessKey ID and an AccessKey secret.

- Use OSS SDKs to generate a signed URL

  The following code provides an example on how to generate a signed URL in Python:

```
import base64
import hmac
import hashlib
import urllib
h = hmac.new("OtxrzxIsfpFjA7SwPzILwy8Bw21TLhquhboDYROV",
             "GET\n\n\n1141889120\n/oss-example/oss-api.pdf",
             hashlib.sha1)
urllib.quote (base64.encodestring(h.digest()).strip())
```

  The following table describes the methods used by OSS SDKs to generate a signed URL and provides sample files.

SDK	Method	Sample file
Java SDK	OSSClient.generatePresignedUrl	OSSClient.java
Python SDK	Bucket.sign_url	api.py
.Net SDK	OssClient.GeneratePresignedUri	OssClient.cs
PHP SDK	OssClient.signUrl	OssClient.php
JavaScript SDK	signatureUrl	Object.js
C SDK	oss_gen_signed_url	oss_object.c
C++ SDK	OssClient::GeneratePresignedUrl	OssClient.cc

SDK

You can use OSS SDKs for the following programming languages to generate a signed URL for object upload and download:

- Java
- Python
- PHP
- Go
- C
- C++
- .NET
- Node.js
- Browser.js
- Android
- iOS

Error codes

Error code	Error message	Description
AccessDenied	403 Forbidden	The error message returned because one or more of the Signature, Expires, and OSSAccessKeyId parameters are missing. When a signature is added to a URL, the sequence of the Signature, Expires, and OSSAccessKeyId parameters can be swapped.
AccessDenied	403 Forbidden	The error message returned because the access time of the request is later than the value of Expires in the request, or the time format of the request is invalid.
InvalidArgument	400 Bad Request	The error message returned because one or more of the Signature, Expires, and OSSAccessKeyId parameters are included in the signed URL and in the request header at the same time.

7.5.4. FAQ

This topic provides answers to frequently asked questions about signatures when you use OSS.

Why is "The request signature we calculated does not match the signature you provided" displayed when OSS calculates a signature?

OSS allows you to include a signature in the Authorization header or in a URL. The following table lists the differences between the two methods.

Header	URL
expires is not supported.	expires is supported.
Methods: GET, POST, PUT, and DELETE.	Methods: GET and PUT.
The date is in GMT.	date is replaced with expires to specify timestamps.
Signatures are not URL-encoded.	Signatures are URL-encoded.

"The request signature we calculated does not match the signature you provided" is displayed when OSS calculates a signature from the Authorization header or a URL. The following code provides an example on how to upload an object to OSS when you use the self-signed mode and call API operations:

```
#! /us/bin/envy python
#Author: hanli
#Update: 2018-09-29
from optparse import OptionParser
import urllib, urllib2
import datetime
import base64
import hmac
import sha
import os
import sys
import time
class Main():
  # Initial input parse
  def __init__(self,options):
    self.ak = options.ak
    self.sk = options.sk
    self.ed = options.ed
    self.bk = options.bk
    self.fi = options.fi
    self.oj = options.objects
    self.left = '\033[1;31;40m'
    self.right = '\033[0m'
    self.types = "application/x-www-form-urlencoded"
    self.url = 'http://{0}.{1} /{2}'.format(self.bk,self.ed,self.oj)
  # Check client input parse
  def CheckParse(self):
    if (self.ak and self.sk and self.ed and self.bk and self.oj and self.fi) != None:
      if str(self.ak and self.sk and self.ed and self.bk and self.oj and self.fi):
        self.PutObject()
    else:
      self.ConsoleLog("error","Input parameters cannot be empty")
  # GET local GMT time
  def GetGMT(self):
    SRM = datetime.datetime.utcnow()
    GMT = SRM.strftime('%a, %d %b %Y %H:%M:%S GMT')
    return GMT
  # GET Signature
  def GetSignature(self):
    mac = hmac.new("{0}".format(self.sk),"PUT\n\n{0}\n{1}\n/{2}/{3}".format(self.types,self.GetGMT(),self.bk,self.oj), sha)
    Signature = base64.b64encode(mac.digest())
    return Signature
  # PutObject
  def PutObject(self):
    try:
      with open(self.fi) as fd:
        files = fd.read()
    except Exception as e:
      self.ConsoleLog("error",e)
    try:
      request = urllib2.Request(self.url, files)
      request.add_header('Host','{0}.{1} '.format(self.bk,self.ed))
      request.add_header('Date','{0}'.format(self.GetGMT()))
      request.add_header('Authorization','OSS {0}:{1}'.format(self.ak,self.GetSignature()))
      request.get_method = lambda:'PUT'
      response = urllib2.urlopen(request,timeout=10)
      fd.close()
      self.ConsoleLog(response.code,response.headers)
    except Exception,e:
      self.ConsoleLog("error",e)
  # output error log
  def ConsoleLog(self,level=None,mess=None):
    if level == "error":
      sys.exit('{0}[ERROR:]{1}{2}'.format(self.left,self.right,mess))
    else:
      sys.exit('\nHTTP/1.1 {0} OK\n{1}'.format(level,mess))
if __name__ == "__main__":
  parser = OptionParser()
  parser.add_option("-i",dest="ak",help="Must fill in Accesskey")
  parser.add_option("-k",dest="sk",help="Must fill in AccessKeySecrety")
  parser.add_option("-e",dest="ed",help="Must fill in endpoint")
  parser.add_option("-b",dest="bk",help="Must fill in bucket")
  parser.add_option("-o",dest="objects",help="File name uploaded to oss")
  parser.add_option("-f",dest="fi",help="Must fill localfile path")
  (options, args) = parser.parse_args()
  handler = Main(options)
  handler.CheckParse()
```

Request headers:

```
PUT /yuntest HTTP/1.1
Accept-Encoding: identity
Content-Length: 147
Connection: close
User-Agent: Python-urllib/2.7
Date: Sat, 22 Sep 2018 04:36:52 GMT
Host: yourBucket.oss-cn-shanghai.aliyuncs.com
Content-Type: application/x-www-form-urlencoded
Authorization: OSS B0g3mdt:lNCA4L0P43Ax
```

Response headers:

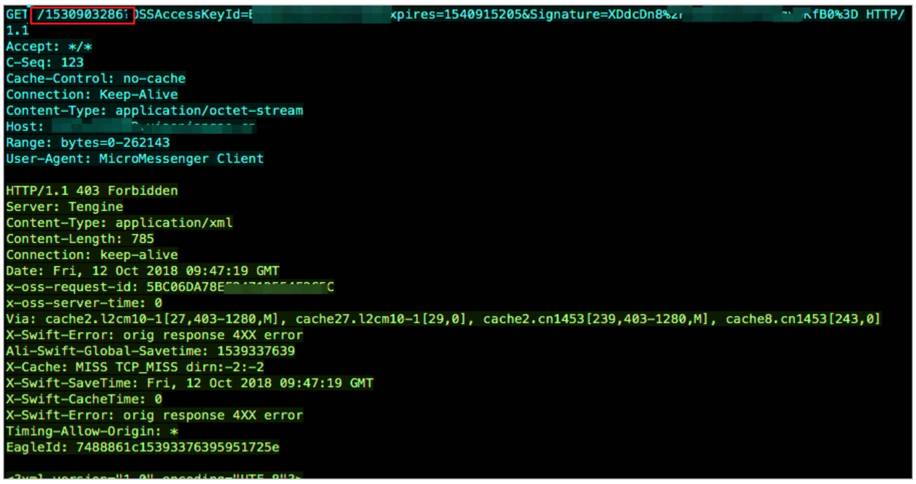
```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Sat, 22 Sep 2018 04:36:52 GMT
Content-Length: 0
Connection: close
x-oss-request-id: 5BA5C6E4059A3C2F
ETag: "D0CAA153941AAA1CBDA38AF"
x-oss-hash-crc64ecma: 8478734191999037841
Content-MD5: 0MqhU5QbIp3Ujqqhy9o4rw==
x-oss-server-time: 15
```

Note

- The parameters to sign are included in the Authorization header. The parameters for the header must be consistent with those for signatures. For more information about the header that carries the signature information, see Add a signature to the header.
- When you use the PUT method, you can set Content-Type to application/x-www-form-urlencoded to calculate signatures.
- You cannot set expires when you use the Authorization method to carry the signature information that authenticates requests. You can set expires only when you generate a signed URL by using OSS SDK or the OSS console.

What do I do if OSS returns a signature error when I use a WeChat mini program to send a request, but no signature errors occur when I use a browser?

The following figure shows the HTTP request captured when the request is sent using a browser.

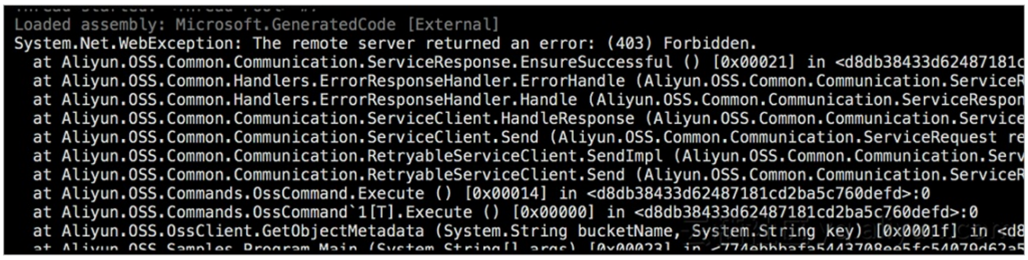


```
GET /1530903286[redacted]OSSAccessKeyId=[redacted]xpires=1540915205&Signature=XDdcDn8%2[redacted]kfB0%3D HTTP/
1.1
Accept: */*
C-Seq: 123
Cache-Control: no-cache
Connection: Keep-Alive
Content-Type: application/octet-stream
Host: [redacted]
Range: bytes=0-262143
User-Agent: MicroMessenger Client

HTTP/1.1 403 Forbidden
Server: Tengine
Content-Type: application/xml
Content-Length: 785
Connection: keep-alive
Date: Fri, 12 Oct 2018 09:47:19 GMT
x-oss-request-id: 5BC06DA78E[redacted]C
x-oss-server-time: 0
Via: cache2.l2cm10-1[27,403-1280,M], cache27.l2cm10-1[29,0], cache2.cn1453[239,403-1280,M], cache8.cn1453[243,0]
X-Swift-Error: orig response 4XX error
Ali-Swift-Global-Savetime: 1539337639
X-Cache: MISS TCP_MISS dirn:-2:-2
X-Swift-SaveTime: Fri, 12 Oct 2018 09:47:19 GMT
X-Swift-CacheTime: 0
X-Swift-Error: orig response 4XX error
Timing-Allow-Origin: *
EagleId: 7488861c15393376395951725e

<?xml version="1.0" encoding="UTF-8"?>
```

- The requests sent by the WeChat mini program and browser share the same URL, signature, and expires values. The difference is the request sent by the WeChat mini program contains Content-Type whereas the request sent by the browser does not.
- Content-Type is not counted when OSS calculates the signature for the request that is sent by the browser, whereas Content-Type is counted when OSS calculates the signature for the request that is sent by the WeChat mini program. Consequently, the calculation results are different.

To resolve this problem, we recommend that you capture packets. If the request header includes Content-Type, Content-Type is counted when OSS calculates the signature.

What do I do if HTTP status code 403 is returned when the client uses an accelerated domain name to calculate a signature and sends a HEAD request?



```
Loaded assembly: Microsoft.GeneratedCode [External]
System.Net.WebException: The remote server returned an error: (403) Forbidden.
  at Aliyun.OSS.Common.Communication.ServiceResponse.EnsureSuccessful () [0x00021] in <d8db38433d62487181c
  at Aliyun.OSS.Common.Handlers.ErrorResponseHandler.ErrorHandle (Aliyun.OSS.Common.Communication.ServiceR
  at Aliyun.OSS.Common.Handlers.ErrorResponseHandler.Handle (Aliyun.OSS.Common.Communication.ServiceRespon
  at Aliyun.OSS.Common.Communication.ServiceClient.HandleResponse (Aliyun.OSS.Common.Communication.Service
  at Aliyun.OSS.Common.Communication.ServiceClient.Send (Aliyun.OSS.Common.Communication.ServiceRequest re
  at Aliyun.OSS.Common.Communication.RetryableServiceClient.SendImpl (Aliyun.OSS.Common.Communication.Serv
  at Aliyun.OSS.Common.Communication.RetryableServiceClient.Send (Aliyun.OSS.Common.Communication.ServiceR
  at Aliyun.OSS.Commands.OssCommand.Execute () [0x00014] in <d8db38433d62487181cd2ba5c760defd>:0
  at Aliyun.OSS.Commands.OssCommand`1[T].Execute () [0x00000] in <d8db38433d62487181cd2ba5c760defd>:0
  at Aliyun.OSS.OssClient.GetObjectMetadata (System.String bucketName, System.String key) [0x0001f] in <d8
  at Aliyun.OSS.Samples.Program.Main (System.String[] args) [0x00023] in <774ebbbafa5443708ee5fc54079d62a5
```

Use tcpdump or Wireshar to capture packets. One possible cause is that when the client sends a HEAD request, a GET request is redirected to OSS using CDN. As a result, the signature calculated by the client is different from the signature that is calculated by OSS. To resolve this problem, upgrade Alibaba Cloud CDN.

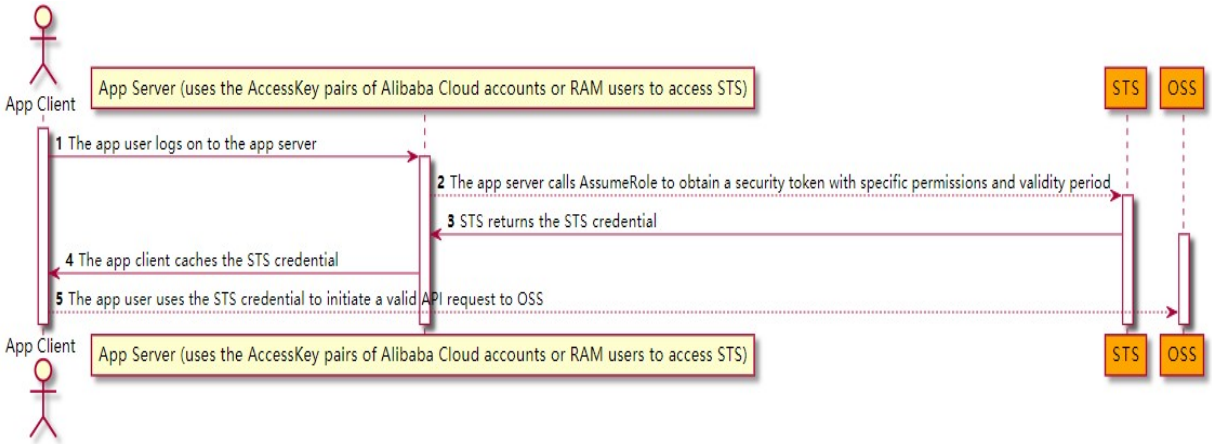
7.6. Use a temporary credential provided by STS to access OSS

You can use Security Token Service (STS) to generate a temporary credential to allow a user to access your Object Storage Service (OSS) resources within the specified period. This way, you do not need to share your AccessKey pair or risk the security of your OSS resources.

Scenarios

A mobile app developer plans to store the user data of an app in Alibaba Cloud OSS. The data of each user must be isolated to prevent users from obtaining the data of other users. In this case, you can use STS to authorize users to access their own data in your OSS buckets.

The following figure shows the process of how to use STS to authorize a user to access OSS.


App Client
App Server (uses the AccessKey pairs of Alibaba Cloud accounts or RAM users to access STS)
STS
OSS
1 The app user logs on to the app server
2 The app server calls AssumeRole to obtain a security token with specific permissions and validity period
3 STS returns the STS credential
4 The app client caches the STS credential
5 The app user uses the STS credential to initiate a valid API request to OSS
App Client
App Server (uses the AccessKey pairs of Alibaba Cloud accounts or RAM users to access STS)
STS
OSS

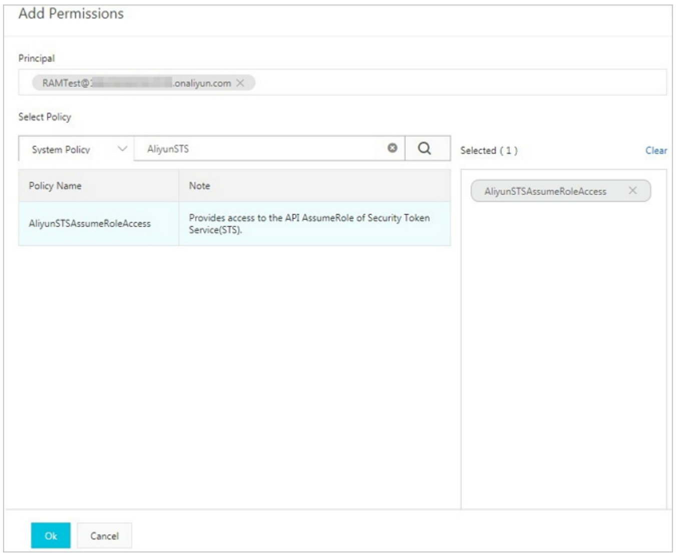
1. The app user logs on to the app server. The username and password used by the app user for logon are independent from the AcceessKey pair of the Alibaba Cloud account used to access OSS. The app server must be capable of defining the minimum access permissions for each valid app user.
2. The app server sends a request to STS to obtain a security token. Before the app server sends the request to STS, the app server must determine the minimum permissions required by the app user to access OSS and the validity period of the credential. You can configure a Resource Access Management (RAM) policy to customize the minimum permissions. Then, the app server calls AssumeRole to obtain a security token that indicates a role from STS.
3. STS returns a temporary access credential to the app server. The credential consists of a security token and a temporary AccessKey pair that can be used to access OSS within a validity period. A temporary AccessKey pair consists of an AccessKey ID and an AccessKey secret.
4. The app server returns the temporary access credential to the app client. The app client can cache the credential. After the credential expires, the app client must apply for a new temporary access credential from the app server. For example, if the temporary access credential is valid for one hour, the app client can send a request to the app server to update the credential every 30 minutes.
5. The app client uses the locally cached temporary access credential to initiate a request to call OSS API operations. After OSS receives the request, OSS uses STS to verify the access credential in the request and responds to the request.

Step 1: Create a RAM user

1. Log on to the RAM console.
2. In the left-side navigation pane, choose **Identities > Users**.
3. On the Users page, click **Create User**.
4. Specify the **Logon Name** and **Display Name** parameters.
5. In the **Access Mode** section, select **Open API Access** and click **OK**.
6. Click **Copy** to save the AccessKey pair of the RAM user.

Step 2: Grant the RAM user the AssumeRole permission

1. On the Users page, click **Add Permissions** in the Actions column that corresponds to the created RAM user.
2. In the **Add Permissions** panel, select the **AliyunSTSAssumeRoleAccess** policy from the policy list for System Policy.



Add Permissions

Principal

RAMTest@[redacted]onaliyun.com ×

Select Policy

System Policy | AliyunSTS

Policy Name	Note
AliyunSTSAssumeRoleAccess	Provides access to the API AssumeRole of Security Token Service(STS).

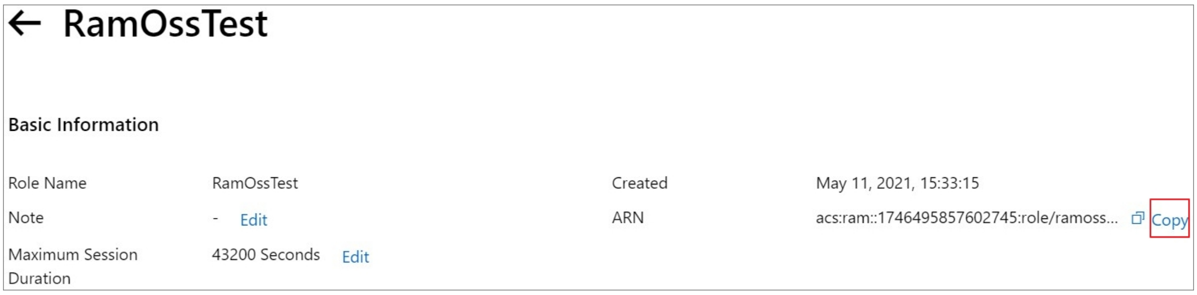
Selected (1) Clear

AliyunSTSAssumeRoleAccess ×

3. Click **OK**.

Step 3: Create a role used to obtain a temporary access credential from STS

1. In the left-side navigation pane, choose **Identities > Roles**.
2. Click **Create Role**. In the Create Role panel, set Select Trusted Entity to **Alibaba Cloud Account**, and then click **Next**.
3. Set **RAM Role Name** to *RamOssTest* and **Select Trusted Alibaba Cloud Account** to **Current Alibaba Cloud Account**.
4. Click **OK**. After the role is created, click **Close**.
5. On the **Roles** page, enter *RamOssTest* in the search box to search for the created role.
6. Click **Copy** to save the Alibaba Cloud Resource Name (ARN) of the role.



← RamOssTest

Basic Information

Role Name	RamOssTest	Created	May 11, 2021, 15:33:15
Note	- Edit	ARN	acs:ram::1746495857602745:role/ramoss... Copy
Maximum Session Duration	43200 Seconds Edit		

Step 4: Grant the role permissions to upload objects to OSS

1. Create a custom policy to grant permissions to upload objects.
   i. In the left-side navigation pane, choose **Permissions > Policies**.
   ii. On the Policies page, click **Create Policy**.
   iii. On the **Create Policy** page, click **JSON**. Then, edit the script in the policy editor to grant the role the permission to upload objects to the exampledir directory in the examplebucket bucket. The following script provides an example on how to edit the script in the policy editor.

   > **Warning** The following example is for reference only. You must configure fine-grained RAM policies based on your requirements to avoid granting excessive permissions to users. For more information about how to configure fine-grained RAM policies, see Common examples of RAM policies.

```
{
    "Version": "1",
    "Statement": [
     {
           "Effect": "Allow",
           "Action": [
             "oss:PutObject"
           ],
           "Resource": [
             "acs:oss:*:*:examplebucket/exampledir",
             "acs:oss:*:*:examplebucket/exampledir/*"
           ]
     }
    ]
}
```

   iv. Click **Next Step**.
   v. In the **Basic Information** section, set **Policy Name** to *RamTestPolicy*. Then, click **OK**.
2. Attach the custom policy to the *RamOssTest* to grant permissions to the role.

i. In the left-side navigation pane, choose **Identities > Roles**.

ii. On the **Roles** page, find the *RamOssTest* role.

iii. Click **Add Permissions** in the Actions column that corresponds to the *RamOssTest* role.

iv. In the **Add Permissions** panel, click the **Custom Policy** tab. Select the *RamTestPolicy* policy.

v. Click **OK**.

Step 5: Obtain a temporary access credential

You can call the AssumeRole operation or use STS SDKs for various programming languages to obtain a temporary access credential from STS.

The following code provides an example on how to obtain a temporary access credential:

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;
import com.aliyuncs.sts.model.v20150401.AssumeRoleRequest;
import com.aliyuncs.sts.model.v20150401.AssumeRoleResponse;
public class StsServiceSample {
    public static void main(String[] args) {
        // Specify the endpoint of STS. Example: sts.cn-hangzhou.aliyuncs.com.
        String endpoint = "<sts-endpoint>";
        // Specify the AccessKey pair generated in Step 1.
        String AccessKeyId = "<yourAccessKeyId>";
        String accessKeySecret = "<yourAccessKeySecret>";
        // Specify the role ARN obtained in Step 3.
        String roleArn = "<yourRoleArn>";
        // Specify a custom role session name to distinguish different tokens. Example: SessionTest.
        String roleSessionName = "<yourRoleSessionName>";
        // The following policy allows users to use only a temporary access credential to upload objects to the bucket named examplebucket.
        // The permission of the temporary access credential is the intersection of the role permission configured in Step 4 and the permission speci
fied by the RAM policy. Users can use the temporary access credential to upload objects only to the exampledir directory in the examplebucket bucket.

        String policy = "{\n" +
                "    \"Version\": \"1\", \n" +
                "    \"Statement\": [\n" +
                "        {\n" +
                "            \"Action\": [\n" +
                "                \"oss:PutObject\"\n" +
                "            ], \n" +
                "            \"Resource\": [\n" +
                "                \"acs:oss:*:*:examplebucket/*\" \n" +
                "            ], \n" +
                "            \"Effect\": \"Allow\"\n" +
                "        }\n" +
                "    ]\n" +
                "}";
        try {
            // regionId specifies the region ID of RAM. If RAM belongs to the China (Hangzhou) region, set regionId to cn-hangzhou. You can also reta
in the default setting, which is an empty string ("").
            String regionId = "";
            // Add the endpoint. This paramater can be configured by using SDK for Java V3.12.0 or later.
            DefaultProfile.addEndpoint(regionId, "Sts", endpoint);
            // Add the endpoint. This parameter can be configured by using SDK for Java that is earlier than version 3.12.0.
            // DefaultProfile.addEndpoint("",regionId, "Sts", endpoint);
            // Create a default profile.
            IClientProfile profile = DefaultProfile.getProfile(regionId, AccessKeyId, accessKeySecret);
            // Use the profile to create a client.
            DefaultAcsClient client = new DefaultAcsClient(profile);
            final AssumeRoleRequest request = new AssumeRoleRequest();
            // This parameter can be configured by using SDK for Java V3.12.0 or later.
            request.setSysMethod(MethodType.POST);
            // This parameter can be configured by using SDK for Java that is earlier than version 3.12.0.
            //request.setMethod(MethodType.POST);
            request.setRoleArn(roleArn);
            request.setRoleSessionName(roleSessionName);
            request.setPolicy(policy); // If the policy is not specified, the user obtains all permissions of the role.
            request.setDurationSeconds(3600L); // Set the validity period of the temporary access credential to 3600 seconds.
            final AssumeRoleResponse response = client.getAcsResponse(request);
            System.out.println("Expiration: " + response.getCredentials().getExpiration());
            System.out.println("Access Key Id: " + response.getCredentials().getAccessKeyId());
            System.out.println("Access Key Secret: " + response.getCredentials().getAccessKeySecret());
            System.out.println("Security Token: " + response.getCredentials().getSecurityToken());
            System.out.println("RequestId: " + response.getRequestId());
        } catch (ClientException e) {
            System.out.println("Failed: ");
            System.out.println("Error code: " + e.getErrCode());
            System.out.println("Error message: " + e.getErrMsg());
            System.out.println("RequestId: " + e.getRequestId());
        }
    }
}
```

Note

- The minimum validity period of a temporary access credential is 900 seconds. The maximum validity period of a temporary access credential is the maximum session duration specified for the current role. For more information, see Specify the maximum session duration for a RAM role.
- For more information about naming conventions for the `roleSessionName` role session name, see AssumeRole.

Step 6: Use the temporary access credential to upload objects to OSS

The following code provides an example on how to use OSS SDK for Java V3.12.0 to upload a file named exampletest.txt from a local path `D:\\localpath` to a directory named exampledir in a bucket named examplebucket:

```
import com.aliyun.oss.OSSClient;
import com.aliyun.oss.model.PutObjectRequest;
import java.io.File;
public class Upload {
    public static void main(String[] args) {
// In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
String endpoint = "oss-cn-hangzhou.aliyuncs.com";
// Specify the temporary AccessKey pair included in the access credential obtained from STS in Step 5.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Specify the security token included in the temporary access credential obtained from STS in Step 5.
String securityToken = "<yourSecurityToken>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret, securityToken);
// Upload a local file named exampletest.txt to the directory named exampledir in the bucket named examplebucket.
PutObjectRequest putObjectRequest = new PutObjectRequest("examplebucket", "exampledir/exampletest.txt", new File("D:\\localpath\\exampletest.txt"));
// The following code provides an example on how to specify the storage class and the access control list (ACL) of the object to upload.
// ObjectMetadata metadata = new ObjectMetadata();
// metadata.setHeader(OSSHeaders.OSS_STORAGE_CLASS, StorageClass.Standard.toString());
// metadata.setObjectAcl(CannedAccessControlList.Private);
// putObjectRequest.setMetadata(metadata);
// Upload the local file.
ossClient.putObject(putObjectRequest);
// Shut down the OSSClient instance.
ossClient.shutdown();
   }
}
```

For more information about the examples on OSS SDKs for other programming languages, see the following topics:

- Java SDK
- Python SDK
- Go SDK
- C++ SDK
- PHP SDK
- C SDK
- .NET SDK
- Node.js SDK
- Browser.js SDK
- Android SDK
- iOS SDK

For more information about how to obtain the URL after the object is uploaded, see How do I obtain the URL of an uploaded object?.

FAQ

- What do I do if the `The security token you provided is invalid.` error message is returned?

  Make sure that you specify the complete security token obtained in Step 5.

- What do I do if the `The OSS Access Key Id you provided does not exist in our records.` error message is returned?

  Use the temporary AccessKey pair to apply for a new temporary access credential from the app server because the current temporary access credential expired. For more information, see Step 5.

- What do I do if the `NoSuchBucket` error message is returned when I obtain an access credential from STS?

  Enter a valid STS `endpoint` based on your region. The error message is returned because the specified STS endpoint is invalid. For more information about the endpoints used to access STS in different regions, see Endpoints.

7.7. Hotlink protection

You can configure a Referer whitelist for a bucket to prevent your resources in the bucket from unauthorized access.

Background information

The hotlink protection feature allows you to configure a Referer whitelist for a bucket and select whether to allow the empty Referer field. This way, only requests from the domain names that are included in the Referer whitelist can access the data in the bucket. You can configure Referer whitelists based on the Referer header field in HTTP and HTTPS requests.

The following scenarios describe whether to use hotlink protection to verify access to Object Storage Service (OSS):

- Only anonymous requests and requests that contain signed URLs are verified.

- Requests that contain the `Authorization` header field are not verified.

OSS determines the source from which a request is sent based on the Referer header field in the request. When a browser sends a request to the web server, the Referer field is contained in the request to indicate the source from which the request is sent. OSS determines whether to allow or deny the request based on the Referer field contained in the request and the Referer whitelist configured for the specified bucket. If the Referer field in the request matches the Referer whitelist, the request is allowed. Otherwise, the request is denied. For example, a bucket has the Referer whitelist set to *https://10.10.10.10*:

- User A adds an image object named test.jpg to the *https://10.10.10.10* website. When a user accesses the image on the website, the browser sends a request in which the value of the Referer field is *https://10.10.10.10*. OSS allows the request because the Referer field in the request is included in the Referer whitelist.
- User B adds the URL of the image object to the *https://192.168.0.0* website without authorization. When a user accesses the image on the website, the browser sends a request in which the value of the Referer field is *https://192.168.0.0*. OSS denies the request because the Referer field in the request is not included in the Referer whitelist.

Referer configuration rules

- You can configure multiple Referers for a bucket. When you configure Referers in the OSS console, press the Enter key to use line feeds to separate Referers. When you call API operations to configure Referers, use commas (,) to separate Referers.
- You can use asterisks (*) and question marks (?) as wildcards in Referers.
    - An asterisk (*) can be used as a wildcard to indicate zero or multiple characters. For example, if you add **.aliyun.com* to the Referer whitelist of a bucket and turn off Allow Empty Referer, only HTTP or HTTPS requests in which the Referer field contains aliyun.com are allowed to access your resources, such as *help.aliyun.com* and *www.aliyun.com*. If you add **.aliyun.com* to the Referer whitelist and turn on Allow Empty Referer, requests in which the Referer field is empty are also allowed to access your resources.
    - A question mark (?) can be used as a wildcard to indicate a character. If you add a Referer that contains a question mark as a wildcard to the Referer whitelist and turn off Allow Empty Referer, only HTTP or HTTPS requests in which the Referer field is contained are allowed to access your resources. If you add a Referer that contains a question mark as a wildcard to the Referer whitelist and turn on Allow Empty Referer, requests in which the Referer field is empty are also allowed to access your resources.
- Referer supports for the use of backslashes(\), asterisks (*), and question marks (?) as escape characters.
- By default, query string are truncated in Referer. You can specify that query strings cannot be truncated.

  For example, if you add *http://www.example.com/?action=nop* to the Referer whitelist, OSS uses *http://www.example.com/* to compare the request URL because query strings are truncated by default. If you want OSS using *http://www.example.com/?action=nop* to compare the request URL, set AllowTruncateQueryString to *false* to disable query strings truncation.

  Note the following items when you disable query strings truncation:

    - Query strings are not decoded.

      If you use *http://www.example.com/?job_id=task$01* to access OSS, the request URL may be retained as *http://www.example.com/?job_id=task$01* or be URL-encoded into *http://www.example.com/?job_id=task%2401*. However, OSS does not decode the query string in the request URL when OSS compares Referers to the URL. Examples:

        - When the Referer is set to *http://www.example.com/?job_id=task%2401*, OSS denies the access from *http://www.example.com/?job_id=task$01*.
        - When the Referer is set to *http://www.example.com/?job_id=task$01*, OSS denies the access from *http://www.example.com/?job_id=task%2401*.

    - Fields contained in query strings are case-insensitive.

      When OSS compares Referers in the whitelist to the request URL, fields contained in query strings are case-insensitive. For example, when the Referer is set to *http://www.example.com/?action=nop*, OSS allows the access from *http://www.examplecom/?ACTION=NOP*.

    - Fields contained in query strings are not parsed.

      By default, browsers consider *http://example.com/?a=b&b=c* and *http://example.com/?b=c&a=b* as identical URLs. However, when OSS compares Referers in the whitelist to request URLs, fields contained in query strings are not parsed. Therefore, *http://example.com/?a=b&b=c* and *http://example.com/?b=c&a=b* are considered as different URLs. In other words, when the Referer is set to *http://example.com/?a=b&b=c*, OSS denies the access from *http://example.com/?b=c&a=b*.

Effects of Referer configurations

- If the Referer whitelist is empty and empty Referer fields are not allowed, all requests are denied.
- If the Referer whitelist is not empty and empty Referer fields are not allowed, only requests that contain the Referers specified in the whitelist are allowed.
- If the Referer whitelist is not empty and empty Referer fields are allowed, only requests in which the Referer field matches the whitelist or the Referer field is empty are allowed.

References

- To set conditions on users who can access part of or all of the resources in your bucket and who can perform certain operations on the resources, we recommend that you configure bucket policies. For example, you can configure a bucket policy to allow only users from specified IP addresses to access the bucket. For more information about how to configure bucket policies, see Configure bucket policies to authorize other users to access OSS resources.
- For more information about how to troubleshoot hotlink protection errors, see Referer.

7.8. Retention policy

Object Storage Service (OSS) supports the Write Once Read Many (WORM) strategy that prevents an object from being deleted or overwritten within a specific period of time. This strategy is applicable to business under the regulations of the U.S. Securities and Exchange Commission (SEC) and Financial Industry Regulatory Authority, Inc. (FINRA).

OSS provides strong compliant policies. You can configure time-based retention policies for buckets. After a retention policy is configured and locked for a bucket, you can read objects from or upload objects to the bucket. However, objects in the bucket or the retention policy cannot be deleted within the retention period that is provided by the retention policy. You can delete the objects only after the retention period expires. The WORM strategy is suitable for industries such as finance, insurance, health care, and securities. OSS provides the WORM strategy to allow you to build a compliant bucket in the cloud.

> Note
>
> OSS is accredited by Cohasset Associates in audit and meets specific requirements for electronic data storage. OSS buckets that are configured with retention policies can be used for the business that is subject to regulations such as SEC Rule 17a-4(f), CFTC Rule 1.31(c)-(d), and FINRA Rule 4511(c). For more information, see OSS Cohasset Assessment Report.

Usage notes

- In OSS, you can configure retention policies only for buckets.
- A bucket cannot have both versioning and retention policies configured at the same time. If versioning is enabled for a bucket, you cannot configure retention policies for the bucket. For more information about versioning, see Overview.A bucket cannot have both versioning and retention policies configured at the same time. If versioning is enabled for a bucket, you cannot configure retention policies for the bucket. For more information about versioning, see Overview.A bucket cannot have both versioning and retention policies configured at the same time. If versioning is enabled for a bucket, you cannot configure retention policies for the bucket. For more information about versioning, see Overview.
- During the protection period for objects in buckets, you can configure lifecycle rules to convert the storage class of the objects to minimize costs while ensuring compliance. For more information, see Lifecycle rules based on the last modified time.

Configuration method

Configure a retention policy in the OSS console. For more information, see Configure retention policies.

Rules

You can add only one time-based retention policy for a bucket. The policy provides a protection period that ranges from one day to 70 years.

Example: A bucket named examplebucket is created on June 1, 2013. The file1.txt, file2.txt, and file3.txt objects are uploaded to the bucket at different points in time. Then, a retention policy that provides a protection period of five years is created on July 1, 2014. The following table describes the upload and expiration dates of the objects.

Object	Upload date	Expiration date
file1.txt	June 1, 2013	May 31, 2018
file2.txt	July 1, 2014	June 30, 2019
file3.txt	September 30, 2018	September 29, 2023

- Implementation rules

  By default, a time-based retention policy is in the IN_PROGRESS state after the policy is created for a bucket. The state remains valid for 24 hours. Within the validity period, the retention policy protects the resources in the bucket.

  - In the 24-hour window after the retention policy is enabled: If the retention policy is not locked, the bucket owner and authorized users can delete this policy. If the retention policy is locked, the protection period of the policy cannot be decreased and the policy cannot be deleted. The protection period can only be increased.
  - 24 hours after the retention policy is enabled: If the retention policy is not locked, the policy becomes invalid.
  - If you attempt to delete or modify data in the protected bucket, the OSS API operation returns the `409 FileImmutable` error.

- Deletion rules
  - A time-based retention policy is a metadata attribute of a bucket. If a bucket is deleted, the retention policy and ACL of the bucket are also deleted. You can delete a bucket only when the bucket is empty.
  - If the retention policy is not locked within 24 hours after the policy is created, the bucket owner and authorized users can delete the policy.
  - If a bucket contains objects that are protected within the protection period, you cannot delete the bucket or the retention policy.

FAQ

- What are the advantages of a retention policy?

  A retention policy can be used to meet data security standards. Within the protection period of a retention policy, data cannot be modified or deleted. In contrast, the data that is protected by using RAM policies and bucket policies can be modified and deleted.

- What are the scenarios in which a retention policy can be used?

  You can use a retention policy if you want to store important data that is infrequently accessed and not allowed to be modified or deleted. This type of data includes medical records, technical documents, and contracts. You can store these archived objects in a specific bucket and configure a retention policy for the bucket.

- Does a retention policy take effect on individual objects?

  Retention policies can be configured only at the bucket level. OSS does not support retention policies at the directory or object level.

- How can I delete a bucket that is protected by a retention policy?
  - If the bucket contains no objects, the bucket can be deleted.
  - If the bucket contains objects, the bucket cannot be deleted even if the protection period expires. You must delete all objects in the bucket, Then, you can delete the bucket.
  - If the bucket contains objects that are protected within the protection period, the bucket cannot be deleted.

- Are objects that are protected within the protection period of the retention policy retained if my account has overdue OSS payments?

  Alibaba Cloud retains data for an account on which overdue payments exist based on the terms and conditions of your contract.

- Can an authorized RAM user configure a retention policy?

  All API operations related to the retention policy are available. These API operations support RAM policies. RAM users that are authorized by using RAM policies can create or delete the retention policy by using the OSS console, API operations, or SDKs.

7.9. OSS sandbox

When your Object Storage Service (OSS) bucket is under attack or is used to distribute illegal content, OSS automatically moves the bucket to the sandbox. A bucket in the sandbox can still respond to requests. However, the service quality is degraded and the performance of your application that uses the bucket may be degraded.

Usage notes

- If your bucket is under attack, OSS automatically moves the bucket to the sandbox. In this case, you must bear the cost incurred by the attack.
- If your user uses your bucket to distribute illegal content such as pornography and terrorism, OSS also moves the bucket to the sandbox. Users will be held liable for violations of the law.

Preventive measures against attacks

To prevent your bucket from being moved to the sandbox due to attacks such as DDoS attacks and Challenge Collapsar (CC) attacks, you can configure OSS DDoS protection for the bucket. You can also set up a reverse proxy by using an Elastic Computing Service (ECS) instance to access the bucket and configure Anti-DDoS Pro for the ECS instance. The following table compares the two solutions.

Solution	Description	Advantage	Disadvantage
Solution 1: Configure OSS DDoS protection	OSS DDoS protection is a proxy-based mitigation service that integrates OSS with Anti-DDoS Pro and Anti-DDoS Premium. When a bucket for which OSS DDoS protection is enabled suffers a DDoS attack, OSS DDoS protection diverts malicious traffic to an Anti-DDoS instance for scrubbing and then redirects normal traffic to the bucket. This way, your business can continue to function normally after a DDoS attack.	• Wide application scope: You can use this solution to protect bucket domain names and custom domain names that are mapped to the bucket. • Low costs: You are charged only for the number of Anti-DDoS instances that you configure for your bucket, the traffic generated by these instances, and the number of requests sent to your bucket. • Simple configurations: You can configure OSS DDoS in the graphical console.	Limited number of protected buckets: You can create only one Anti-DDoS instance within each region. Each instance can be attached to at most 10 buckets that are located in the same region.
Solution 2: Set up a reverse proxy by using an ECS instance to access the bucket and configure Anti-DDoS Pro for the ECS instance	For security reasons, the default domain name of a bucket is resolved to a random IP address each time when the bucket is accessed. If you want to use a static IP address to access the bucket, you can set up a reverse proxy by using an ECS instance to access the bucket. You can associate the Elastic IP address (EIP) of the ECS instance with Anti-DDoS Pro to protect the bucket against DDoS attacks and CC attacks.	You can use this solution to protect your bucket when you use a static IP address to access OSS.	• Complex configurations: You must set up an NGINX reverse proxy on your own. • High costs: You must purchase an ECS instance to set up an NGINX reverse proxy.

Implementation procedure

- Solution 1: Configure OSS DDoS protection

  Perform the following steps to configure OSS DDoS protection for a bucket in the OSS console:

  i. Step 1: Create an Anti-DDoS instance.

  ii. Step 2: Attach the bucket that you want to protect to the Anti-DDoS instance.

  iii. Step 3: If you want to use a custom domain name to access the bucket when the bucket suffers attacks, add the custom domain name in the OSS console.

  For more information about OSS DDoS protection, see Configure OSS DDoS protection.

- Solution 2: Set up a reverse proxy by using an ECS instance to access the bucket and configure Anti-DDoS Pro for the ECS instance

  Perform the following steps to set up a reverse proxy by using an ECS instance to access the bucket and configure Anti-DDoS Pro for the ECS instance:

  i. Step 1: Set up a reverse proxy by using an ECS instance to access your bucket.

  a. Create an ECS instance that runs CentOS or Ubuntu. For more information about how to create an ECS instance, see Create an instance by using the wizard.

  > Notice If the bucket encounters sporadic bursts of network traffic or access requests, upgrade hardware configurations of ECS or set up ECS clusters.

  b. Set up a reverse proxy by using an ECS instance to access the bucket. For more information, see Use an ECS instance that runs CentOS to configure a reverse proxy for access to OSS.

  ii. Step 2: Configure Anti-DDoS Pro for the ECS instance.

  a. Purchase Anti-DDoS Pro based on your requirements. For more information, visit the buy page of Anti-DDoS Pro.

  b. Configure Anti-DDoS Pro. Enter the endpoint of the bucket that you want to protect by using the ECS reverse proxy in **Domain**. Select **Origin Server IP** for Server IP and enter the public IP address of the ECS instance in the field. For more information about how to configure other parameters, see Add a website.

7.10. OSS DDoS protection

Object Storage Service (OSS) DDoS protection is a proxy-based mitigation service that integrates OSS with Anti-DDoS Pro and Anti-DDoS Premium. When a bucket with OSS DDoS protection enabled suffers DDoS attacks, OSS DDoS protection diverts malicious traffic to an Anti-DDoS instance for scrubbing and then redirects normal traffic to the bucket. This way, your business can continue to function normally after DDoS attacks.

> Note OSS DDoS protection is supported only in the China (Hangzhou), China (Shanghai), China (Qingdao), China (Beijing), China (Shenzhen), and China (Hong Kong) regions.

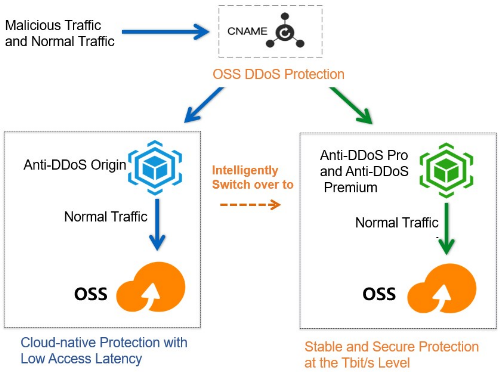
Scenarios

DDoS attacks are one of the most harmful attacks against enterprises in recent years. When an enterprise suffers DDoS attacks, its business may be interrupted. This affects business operations and causes damage to the corporate identity, customer attrition, and loss of profits.

To mitigate these problems, OSS is integrated with Anti-DDoS Pro and Anti-DDoS Premium to provide the highest DDoS attack mitigation capability at the Tbit/s level, millions of queries per second (QPS), and switchovers from Anti-DDoS Origin to Anti-DDoS Pro or Anti-DDoS Premium within a few seconds. These capabilities can protect your business from attacks such as SYN flood, ACK flood, Internet Control Message Protocol (ICMP) flood, UDP flood, NTP flood, Simple Service Discovery Protocol (SSDP) flood, Domain Name System (DNS) flood, and HTTP flood. OSS DDoS protection is suitable for scenarios where your business is vulnerable to attacks, ransom-driven attacks, click farming, and fraudulent traffic.

How does OSS DDoS protection work

The following figure shows how OSS DDoS protection works.


Malicious Traffic and Normal Traffic
CNAME
OSS DDoS Protection
Anti-DDoS Origin
Normal Traffic
OSS
Intelligently Switch over to
Anti-DDoS Pro and Anti-DDoS Premium
Normal Traffic
OSS
Cloud-native Protection with Low Access Latency
Stable and Secure Protection at the Tbit/s Level

By default, OSS uses Anti-DDoS Origin to protect your bucket from attacks. However, when the attack frequency exceeds the protection threshold of Anti-DDoS Origin, Anti-DDoS Origin cannot provide effective mitigation and your bucket may not be accessible.

After you enable OSS DDoS protection, when the attack frequency exceeds the protection threshold of Anti-DDoS Origin, OSS diverts all traffic to access the bucket to an Anti-DDoS instance. Malicious traffic is scrubbed in the scrubbing center of Anti-DDoS Pro or Anti-DDoS Premium. Only legitimate traffic is forwarded to the requested bucket by using the port protocol. This way, normal access to the bucket is ensured when the bucket suffers attacks.

After the attacks stop, OSS protects the attacked bucket by using Anti-DDoS Origin.

Usage notes

- An Anti-DDoS instance must be retained for at least seven days after the instance is created. If the instance is deleted within seven days, you are charged basic resource fees for the instance for a period of seven days.
- You can create only one Anti-DDoS instance within each region. Each instance can be attached to up to 10 buckets within the same region.
- After you attach the instance to a bucket, you cannot preview the resources in the bucket by using browsers. In addition, OSS does not protect the custom domain names mapped to the bucket by default. Therefore, when the bucket suffers attacks, you cannot access the bucket by using the custom domain names. If you want to access the bucket when it suffers attacks by using the custom domain names, add the custom domain names in the OSS console. You can add up to five custom domain names for each bucket.

  If the custom domain name of the bucket (such as `www.example.com`) that you want to protect matches an accurate domain name (`www.example.com`) or a wildcard domain name (`*.example.com`) that is specified in a forwarding rule of the Anti-DDoS instance, you must go to the Anti-DDoS Pro console to remove the mapping between the accurate domain name or the wildcard domain name and the instance. Otherwise, when the bucket is under attack, you cannot access the bucket by using the custom domain name.

  For more information about the forwarding rules of Anti-DDoS instances, see Add a website.

Configuration methods

To enable the protection of OSS Anti-DDoS Pro for a bucket, you need only to configure the protection in the OSS console. For more information about the configuration methods, see Configure OSS DDoS protection.

8.Manage logs

8.1. Logging

A large number of logs are generated when your Object Storage Service (OSS) buckets are accessed. After you enable and configure logging for a bucket, OSS generates logs every hour in accordance with predefined naming conventions and then stores the logs as objects in a specified bucket. You can use Alibaba Cloud Log Service or build a Spark cluster to analyze the logs.

Usage notes

- The source bucket for which logs are generated and the destination bucket in which the logs are stored can be the same or different. However, the destination bucket must belong to the same Alibaba Cloud account in the same region as the source bucket.
- OSS generates bucket access logs on an hourly basis. However, requests in an hour may be recorded in the log generated for the previous or subsequent hour.
- Before you disable logging, OSS keeps generating log objects. Delete log objects that you no longer need to reduce your storage costs.

  You can configure lifecycle rules to regularly delete log objects. For more information, see Lifecycle rules based on the last modified time
- More fields may be added to OSS logs. We recommend that developers consider potential compatibility issues when they develop log processing tools.

Naming conventions of log objects

The following naming conventions apply to log objects:

```
<TargetPrefix><SourceBucket>YYYY-mm-DD-HH-MM-SS-UniqueString
```

Field	Description
TargetPrefix	The prefix of the log object name.
SourceBucket	The name of the source bucket for which the access logs are generated.
YYYY-mm-DD-HH-MM-SS	The time when the log object was created. The items of this field indicate the year, month, day, hour, minute, and second in sequence.
UniqueString	The string generated by OSS to uniquely identify the log object.

Log formats and examples

- Log formats

  OSS access logs include the information about the requester and accessed resources in the following format:

```
RemoteIP Reserved Reserved Time "RequestURL" HTTPStatus SentBytes RequestTime "Referer" "UserAgent" "HostName" "RequestID" "LoggingFlag" "RequesterAliyunID" "Operation" "BucketName" "ObjectName" ObjectSize ServerCostTime "ErrorCode RequestLength "UserID" DeltaDataSize "SyncRequest" "StorageClass" "TargetStorageClass" "TransmissionAccelerationAccessPoint" "AccessKeyID"
```

Field	Example	Description
RemoteIP	192.168.0.1	The IP address of the requester.
Reserved	-	The reserved field. It is automatically set to -.
Reserved	-	The reserved field. It is automatically set to -.
Time	03/Jan/2021:14:59:49 +0800	The time when OSS received an access request.
RequestURL	GET /example.jpg HTTP/1.0	The request URL that contains a query string. OSS ignores the query string parameter that starts with `x-`. However, this parameter is recorded in logs. Therefore, you can tag a request by using a query string parameter that starts with `x-`. Then, you can use this tag to query the log that corresponds to the request.
HTTPStatus	200	The HTTP status code that OSS returns.
SentBytes	999131	The downstream traffic generated by the request. Unit: bytes.
RequestTime	127	The amount of time used to complete the request. Unit: milliseconds.
Referer	http://www.aliyun.com/product/oss	The Rerferer header in the HTTP request.
UserAgent	curl/7.15.5	The User-Agent header in the HTTP request.
HostName	examplebucket.oss-cn-hangzhou.aliyuncs.com	The destination domain name that the request attempts to access.
RequestID	5FF16B65F05BC932307A3C3C	The ID of the request.
LoggingFlag	true	Indicates whether logging is enabled. Valid values: ◦ *true*: Logging is enabled. ◦ *false*: Logging is disabled.
RequesterAliyunID	16571836914537****	The user ID of the requester. A value of - indicates anonymous access.
Operation	GetObject	The type of the request.

Field	Example	Description
BucketName	examplebucket	The name of the destination bucket that the request attempts to access.
ObjectName	example.jpg	The name of the destination object that the request attempts to access.
ObjectSize	999131	The size of the destination object. Unit: bytes.
ServerCostTime	88	The time that OSS takes to process the request. Unit: milliseconds.
ErrorCode	-	The error code that is returned by OSS. A value of - indicates that no error code is returned.
RequestLength	302	The length of the request. Unit: bytes.
UserID	16571836914537****	The ID of the bucket owner.
DeltaDataSize	-	The size change of an object. A value of - indicates that this request does not involve write operations on objects.
SyncRequest	cdn	Indicates whether the request is an Alibaba Cloud Content Delivery Network (CDN) back-to-origin request. Valid values: ○ *cdn*: The request is a CDN back-to-origin request. ○ -: The request is not a CDN back-to-origin request.
StorageClass	Standard	The storage class of the destination object. Valid values: ○ *Standard* ○ *IA* ○ *Archive* ○ *Cold Archive* ○ -: The storage class of the object is not obtained.
TargetStorageClass	-	Indicates whether the storage class of the object is converted based on a lifecycle rule or the CopyObject operation. Valid values: ○ *Standard* ○ *IA* ○ *Archive* ○ *Cold Archive* ○ -: The request does not involve operations to convert the storage class of the object.
TransmissionAccelerationAccessPoint	-	The accelerate endpoint used when transfer acceleration is used to access the destination bucket. For example, if the requester accesses the destination bucket by using an accelerate endpoint in the China (Hangzhou) region, the value of TransmissionAccelerationAccessPoint is *cn-hangzhou*. A value of - indicates that no accelerate endpoint is used or the accelerate endpoint is in the same region as the destination bucket.
AccessKeyID	LTAI4FrfJPUSoKm4JHb5****	The AccessKey ID of the requester. A value of - indicates anonymous requests.

- Log formats

```
192.168.0.1 - - [03/Jan/2021:14:59:49 +0800] "GET /example.jpg HTTP/1.0" 200 999131 127 "http://www.aliyun.com/product/oss" "curl/7.15.5" "exampleb
ucket.oss-cn-hangzhou.aliyuncs.com" "5FF16B65F05BC932307A3C3C" "true" "16571836914537****" "GetObject" "examplebucket" "example.jpg" 999131 88 "-"
302 "16571836914537****" - "cdn" "standard" "-" "-" "LTAI4FrfJPUSoKm4JHb5****"
```

After the log objects are stored in the specified bucket in OSS, you can use Alibaba Cloud Log Service to analyze the log objects. Before you analyze the log objects, you must import the log objects to Alibaba Cloud Log Service. For more information about how to import data, see Import data from OSS to Log Service. For more information about the analysis feature of Log Service, see Log analysis overview.

Use the OSS console

1. Log on to the OSS console.
2.
3. In the left-side navigation pane, choose **Logging > Logging**.
4. Click **Configure**. Set Destination Bucket and Log Prefix.
   - **Destination Bucket**: Select a bucket name from the Destination Bucket drop-down list. You can only select a bucket that is located in the same region as the bucket for which logging is enabled under the same Alibaba Cloud account.
   - **Log Prefix**: Enter the path and prefix of the logs. If you configure this parameter, the log objects are stored in the specified directory of the destination bucket. If you do not configure this parameter, the log objects are stored in the root directory of the destination bucket. For example, if you enter *log/* in Log Prefix, the logs are stored in the *log/* directory.

   > **Note** For more information about the format of access logs and log object naming conventions, see Logging.
5. Click **Save**.

Use OSS SDKs

The following code provides examples on how to enable logging by using OSS SDKs for common programming languages. For more information about the sample code used to enable logging by using OSS SDKs for other programming languages, see Overview.

Python

```
Failed to resolve content from t22302.dita#concept_32019_zh/codeblock_2hh_6bd_b4g
```

Use ossutil

For more information about how to enable logging by using ossutil, see Enable logging for a bucket.

Use the RESTful API

If your program requires more custom options to enable logging for a bucket, you can call RESTful API operations. In this case, you must manually write code to calculate the signature. For more information, see PutBucketLogging.

FAQ

Can I query interrupted requests in OSS access logs?

No, OSS does not record interrupted requests in access logs. If you send a request by using an SDK, you can identify the cause of request interruptions based on the returned value of the SDK.

8.2. Real-time log query

When you access OSS, a large number of access logs are generated. Real-time log query combines OSS with Log Service. This feature allows you to query and collect statistics for OSS access logs and audit access to OSS by using the OSS console, track exceptions, and troubleshoot problems. Real-time log query enables you to be more efficient and make informed decisions.

Comparison between real-time log query and logging

- Real-time log query:
  - Pushes logs to Log Service within three minutes and allows you to view real-time logs in the OSS console.
  - Provides the log analysis service and typical analysis reports so that you can easily query data.
  - Allows you to query and analyze raw logs in real time and filter logs by bucket, object name, API operation, or time.
- Logging:
  - Allows you to enable logging for a bucket, after which OSS generates log objects in accordance with a predefined naming convention. This way, hourly access logs are written to the specified bucket as objects.
  - Allows you to use Alibaba Cloud Data Lake Analytics or build a Spark cluster to analyze access logs.
  - Allows you to configure lifecycle rules for the specified destination bucket to convert the storage class of the log objects to Archive or Cold Archive. This way, these log objects can be retained for a long time.

Configuration method

Console: Configure real-time log query

Query method

The real-time log query feature provides the following query methods:

- Query raw logs

  You can specify the time range and query statement to query logs in real time and perform the following operations:

  - Analyze the distribution of a specified field such as an API operation within a specified time range.
  - Filter by field to view required access records. For example, filter the object deletion operations for the last day by bucket, object name, or API operation name, and query the deletion time and IP address.
  - Collect statistics for OSS access records such as the page view (PV), unique visitor (UV), or maximum latency of a bucket within a specified time range.
- Use Dashboard

  Dashboard allows you to view four immediately available reports.

  - **Access Center**: displays the overall operating status including the PV, UV, traffic, and distribution of access over the Internet.
  - **Audit Center**: displays statistics for object operations including read, write, and delete operations on objects.
  - **Operation Center**: displays statistics for access logs including the number of requests and distribution of failed operations.
  - **Performance Center**: displays statistics for performance including the performance of downloads and uploads over the Internet, the performance of transmission over different networks or with different object sizes, and the list of differences between stored and downloaded objects.
- Use the Log Service console

  You can view OSS access logs in the Log Service console. For more information, see OSS access logs.

Billing method

- Real-time log query allows you to query logs from the past seven days free of charge. If the log retention time that you set is longer than seven days, Log Service charges the fees for the excess days. Extra fees are charged when you read data from or write data to Log Service over the Internet.
- Log Service allows you to store 900 GB of logs (equivalent to 900 million 1-KB log entries) per day free of charge, and charges fees for excess logs.

For more information about the billing standards, see Pay-as-you-go.

9.Static website hosting

9.1. Overview

Static websites are websites in which all web pages consist of only static content, including scripts such as JavaScript code that can be run on the client. You can use the static website hosting feature to host your static website on an Object Storage Service (OSS) bucket and use the endpoint of the bucket to access the website.

Usage notes

When you configure static website hosting, you must specify the default homepage and the default 404 page for the website.

- The default homepage that appears when you use a browser to access the static website hosted on the OSS bucket.

  The object that you specify as the default homepage must be an object that is stored in the root directory of the bucket and allows anonymous access. If you have the subfolder homepage feature enabled, the object must also be stored in the subdirectory.

- The default 404 page is the error page returned by OSS. When you use a browser to access the static website hosted on an OSS bucket and a 404 error occurs, OSS returns the default 404 page.

  The object that you specify as the default 404 page must be an object that is stored in the root directory of the bucket and allows anonymous access.

To allow anonymous access to the object, you must set the access control list (ACL) of the object that is specified as the default homepage or default 404 page to `public-read`. For more information about how to set object ACLs, see Configure ACL for objects.

When you access a static website hosted on a bucket by using the default endpoint of the bucket, the website is downloaded as a file to your computer. To preview a static website hosted on a bucket, you must map a custom domain name to the bucket and access the website by using the custom domain name. For more information about how to map a custom domain name to a bucket, see Map custom domain names.

Usage notes

For security reasons, starting from August 13, 2018 for regions inside mainland China, and September 25, 2019 for regions outside China, when you access web page objects whose MIME type is text/html and whose name extension is HTM, HTML, JSP, PLG, HTX, or STM by using browsers:

- If you use the default endpoint of the bucket to access the objects, the following header is automatically contained in the response: `Content-Disposition:'attachment=filename;'`. In this case, the web page objects are downloaded as attachments. The content of the object cannot be previewed.
- If you use a custom domain name mapped to the bucket to access the objects, the `Content-Disposition:'attachment=filename;'` header is not contained in the response. In this case, you can preview the object content if your browser supports preview of web page objects. For more information about how to map a custom domain name to a bucket, see Map custom domain names.

For more information, see Overview.

Configuration examples

After you host a static website on a bucket, you must upload an object whose name is the same as that of the default homepage to the bucket. Example: *index.html*. If the bucket contains a directory named *subdir/*, you must upload the *index.html* object to the directory. In addition, you must upload an object whose name is the same as that of the default 404 page to the bucket. Example: *error.html*. The following structure shows the objects and directories in the sample bucket:

```
Bucket
 ├── index.html
 ├── error.html
 ├── example.txt
 └── subdir/
      └── index.html
```

In this example, the custom domain name `example.com` is mapped to the bucket, the default homepage of the static website hosted on the bucket is *index.html*, and the default 404 page of the website is *error.html*. When you access the static website by using the custom domain name, OSS returns different responses based on your configurations of Static Pages for the bucket that hosts the website.

- If Subfolder Homepage is disabled:
  - When you access *https://example.com/* and *https://example.com/subdir/*, OSS returns *https://example.com/index.html*.
  - When you access *https://example.com/example.txt*, the *example.txt* object is obtained.
  - When you access *https://example.com/object*, OSS returns *https://example.com/error.html* if the *object* object does not exist.
- If Subfolder Homepage is enabled:
  - When you access *https://example.com/*, OSS returns *https://example.com/index.html*.
  - When you access *https://example.com/subdir/*, OSS returns *https://example.com/subdir/index.html*.
  - When you access *https://example.com/example.txt*, the *example.txt* object is obtained.
  - When you access *https://example.com/object*, OSS returns one of the following responses based on your configurations for Subfolder 404 Rule because the *object* object does not exist:
    - If Subfolder 404 Rule is set to the default value Redirect, OSS continues to check whether *object/index.html* exists. If object/index.html exists, OSS returns a 302 status code and redirects the request to *https://example.com/object/index.html*. If object/index.html does not exist, OSS returns a 404 status code and *https://example.com/error.html*.
    - If Subfolder 404 Rule is set to NoSuchKey, OSS returns a 404 status code and *https://example.com/error.html*.
    - If Subfolder 404 Rule is set to Index, OSS continues to check whether the *object/index.html* object exists. If this object exists, OSS returns 200 and the content of this object. If the object does not exist, OSS returns *https://example.com/error.html*.

Use the OSS console

1. Configure static website hosting.
   - If Subfolder Homepage is disabled:

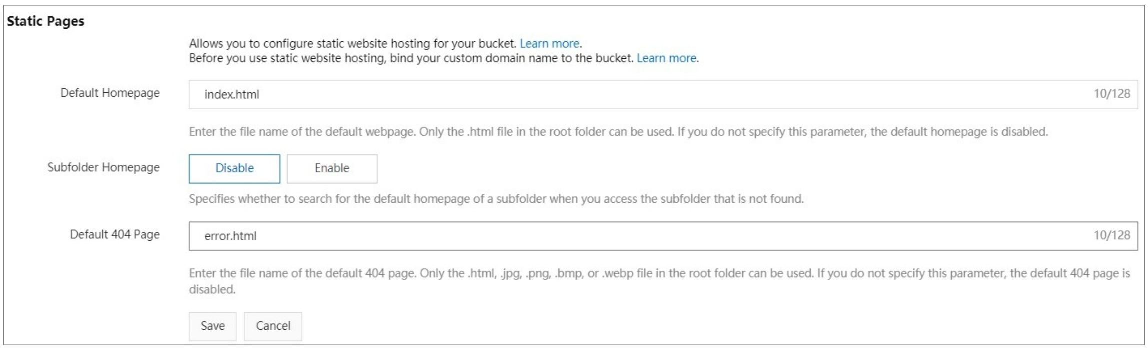
     In the preceding example, when you access the subdir/ subdirectory of the bucket, the default homepage object named index.html in the root directory of the bucket is returned instead of the object named index.html in the subdir/ subdirectory. In addition, if you access an object that does not exist in the bucket, the specified default 404 page is returned. Perform the following steps to configure static website hosting:

     a. Log on to the OSS console.

b. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket on which you want to host a static website.

c. In the left-side navigation pane, choose **Basic Settings > Static Pages**.

d. In the **Static Pages** section, click **Configure** and set the parameters described in the following table.



Static Pages

Allows you to configure static website hosting for your bucket. Learn more.
Before you use static website hosting, bind your custom domain name to the bucket. Learn more.

Default Homepage: index.html 10/128

Enter the file name of the default webpage. Only the .html file in the root folder can be used. If you do not specify this parameter, the default homepage is disabled.

Subfolder Homepage: Disable | Enable

Specifies whether to search for the default homepage of a subfolder when you access the subfolder that is not found.

Default 404 Page: error.html 10/128

Enter the file name of the default 404 page. Only the .html, .jpg, .png, .bmp, or .webp file in the root folder can be used. If you do not specify this parameter, the default 404 page is disabled.

Parameter	Description
Default Homepage	The default homepage that appears when you use a browser to access the static website hosted on the OSS bucket. In this example, set this parameter to *index.html*.
Subfolder Homepage	Specifies whether to enable the subfolder homepage feature for the bucket. In this example, select **Disable**. In that case, when you access the root directory of the bucket or a subdirectory whose URL ends with a forward slash (/), the default homepage object in the root directory of the bucket is returned.
Default 404 Page	The error page that is returned when the object that you want to access does not exist in the bucket and a 404 error occurs. Only an object in the root directory of the bucket can be specified as the default 404 page. In this example, set this parameter to *error.html*.
Error Page Status Code	The HTTP status code that is returned with the error page. Valid values: **404** and **200**.

e. Click **Save**.

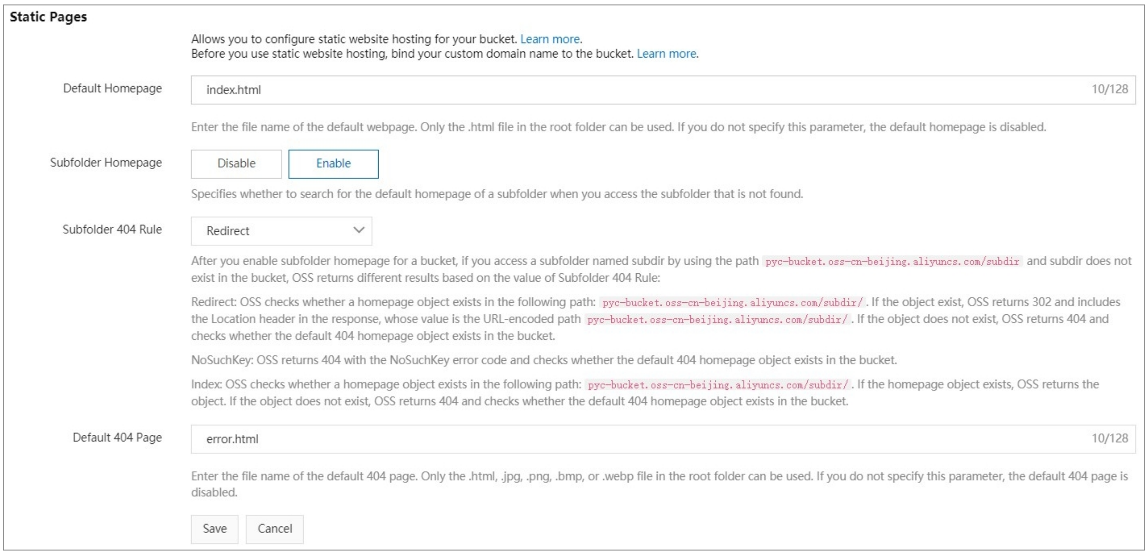
- If the subfolder homepage feature is enabled:

  In the preceding example, when you access the subdir/ subdirectory of the bucket, the default homepage object named index.html in the subdirectory of the bucket is returned. In addition, if you access an object that does not exist in the bucket, the specified default 404 page and a result based on the specified subdirectory 404 rule are returned. Perform the following steps to configure static website hosting:

  a. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket on which you want to host the static website.

  b. In the left-side navigation pane, choose **Basic Settings > Static Pages**.

c. In the **Static Pages** section, click **Configure** and set the parameters described in the following table.



Static Pages

Allows you to configure static website hosting for your bucket. Learn more.
Before you use static website hosting, bind your custom domain name to the bucket. Learn more.

Default Homepage: index.html 10/128

Enter the file name of the default webpage. Only the .html file in the root folder can be used. If you do not specify this parameter, the default homepage is disabled.

Subfolder Homepage: Disable [✓ Enable]

Specifies whether to search for the default homepage of a subfolder when you access the subfolder that is not found.

Subfolder 404 Rule: Redirect

After you enable subfolder homepage for a bucket, if you access a subfolder named subdir by using the path `pyc-bucket.oss-cn-beijing.aliyuncs.com/subdir` and subdir does not exist in the bucket, OSS returns different results based on the value of Subfolder 404 Rule:

Redirect: OSS checks whether a homepage object exists in the following path: `pyc-bucket.oss-cn-beijing.aliyuncs.com/subdir/`. If the object exist, OSS returns 302 and includes the Location header in the response, whose value is the URL-encoded path `pyc-bucket.oss-cn-beijing.aliyuncs.com/subdir/`. If the object does not exist, OSS returns 404 and checks whether the default 404 homepage object exists in the bucket.

NoSuchKey: OSS returns 404 with the NoSuchKey error code and checks whether the default 404 homepage object exists in the bucket.

Index: OSS checks whether a homepage object exists in the following path: `pyc-bucket.oss-cn-beijing.aliyuncs.com/subdir/`. If the homepage object exists, OSS returns the object. If the object does not exist, OSS returns 404 and checks whether the default 404 homepage object exists in the bucket.

Default 404 Page: error.html 10/128

Enter the file name of the default 404 page. Only the .html, .jpg, .png, .bmp, or .webp file in the root folder can be used. If you do not specify this parameter, the default 404 page is disabled.

Parameter	Description
Default Homepage	The default homepage that appears when you use a browser to access the static website hosted on the OSS bucket. In this example, set this parameter to *index.html*.
Subfolder Homepage	Specifies whether to enable the subfolder homepage for the bucket. In this example, select **Enable**. After you enable the subfolder homepage feature for a bucket, if you access the root directory of the bucket, the default homepage in the root directory is returned. If you access a subdirectory whose URL ends with a forward slash (/), the default homepage in the subdirectory is returned. For example, if you access `https://examplebucket.oss-cn-hangzhou.aliyuncs.com/subdir/`, the default homepage object `index.html` in the *subdir/* directory is returned.
Subfolder 404 Rule	The subdirectory 404 rule for the bucket. When you access an object that does not exist in the bucket, OSS returns different results based on the specified subdirectory 404 rule. For example, if you use the URL `https://examplebucket.oss-cn-hangzhou.aliyuncs.com/exampledir` to access an object named *exampledir* that does not exist in examplebucket, OSS returns different results based on the value that you set for this parameter. Default value: Redirect. Valid values: ■ **Redirect**: OSS checks whether the *exampledir/index.html* object exists. 　■ If this object exists, OSS returns 302 and redirects the request to `https://examplebucket.oss-cn-hangzhou.aliyuncs.com/exampledir/index.html`. 　■ If this object does not exist, OSS returns 404 and checks whether the `https://examplebucket.oss-cn-hangzhou.aliyuncs.com/error.html` object exists. If this object does not exist either, OSS returns 404 status code. ■ **NoSuckKey**: OSS returns 404 and checks whether the `https://examplebucket.oss-cn-hangzhou.aliyuncs.com/error.html` object exists. ■ **Index**: OSS checks whether the *exampledir/index.html* object exists. 　■ If this object exists, OSS returns 200 and the content of this object. 　■ If this object does not exist, OSS checks whether the `https://examplebucket.oss-cn-hangzhou.aliyuncs.com/error.html` object exists.
Default 404 Page	The error page that is returned when the object that you want to access does not exist in the bucket and a 404 error occurs. Only an object in the root directory of the bucket can be specified as the default 404 page. In this example, set this parameter to *error.html*.
Error Page Status Code	The HTTP status code that is returned with the error page. Valid values: **404** and **200**.

d. Click **Save**.

2. Create and upload a default homepage object.

If you set the default homepage to *index.html* when you configure static website hosting for the examplebucket bucket, you must upload an object named index.html to the root directory of the bucket. If you enable the subfolder homepage for the bucket, you must also upload *index.html* to the *subdir/* subdirectory of the bucket.

i. Create a file named *index.html*. The content of the *index.html* file is similar to the following example:

```
<html>
<head>
    <title>My Website Home Page</title>
    <meta charset="utf-8">
</head>
<body>
  <p>Now hosted on OSS.</p>
</body>
</html>
```

ii. Save *index.html* to a local path.

iii. Upload *index.html* to the root directory and subdir/ subdirectory of examplebucket. You must set the ACL of the index.html object to public read.

For more information about how to upload objects, see Simple upload.

3. Create and upload a default 404 page.

If you set the default 404 page to *error.html* when you configure static website hosting for examplebucket, you must upload an object named error.html to the root directory of examplebucket.

i. Create a file named *error.html*. The content of the *error.html* file is similar to the following example:

```
<html>
<head>
    <title>Hello OSS!</title>
    <meta charset="utf-8">
</head>
<body>
  <p>This is error 404 page.</p>
</body>
</html>
```

ii. Save *error.html* to a local path.

iii. Upload *error.html* to the root directory of examplebucket. You must set the ACL of the error.html object to public read.

Use OSS SDKs

The following code provides examples on how to configure static website hosting by using OSS SDKs for common programming languages. For more information about how to configure static website hosting by using OSS SDKs for other programming languages, see Overview.

Python

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify the endpoint based on your business requirements.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API operations. We recommend that you use a RAM user to call API operations or perform routine operations and maintenance. To create a RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Specify the bucket name.
SetBucketWebsiteRequest request = new SetBucketWebsiteRequest("<yourBucketName>");
// Specify the default homepage for the static website hosted on the bucket.
request.setIndexDocument("index.html");
// Specify the default 404 page for the static website hosted on the bucket.
request.setErrorDocument("error.html");
ossClient.setBucketWebsite(request);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Use the RESTful API

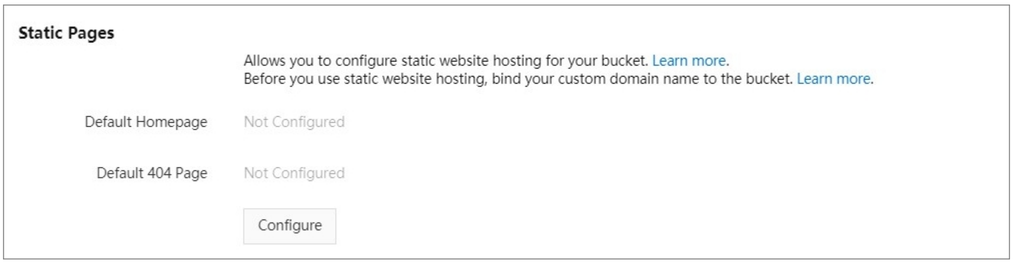
If your program requires more custom options to configure static website hosting, you can call RESTful API operations. In that case, you must manually write code to calculate the signature. For more information, see PutBucketWebsite.

FAQ

Can I disable the static website hosting feature after I enabled it for a bucket?

If you no longer need to use the configurations of static website hosting, perform the following steps to disable the static website hosting feature:

1.
2.
3. In the left-side navigation pane, choose **Basic Settings > Static Pages**. In the **Static Pages**, click **Configure**.
4. Remove the configurations of the Default Homepage and Default 404 Page parameters and click **Save**.
If a similar figure is returned, the static website hosting feature is disabled.



Static Pages

Allows you to configure static website hosting for your bucket. Learn more.
Before you use static website hosting, bind your custom domain name to the bucket. Learn more.

Default Homepage Not Configured

Default 404 Page Not Configured

References

- You can host a static website on an OSS bucket and allow users to access the website by using the custom domain name that is mapped to the bucket. Example: example.com. For more information about how to host a static website on a bucket, see Tutorial: Use a custom domain name to configure static website hosting.
- You can use React and the static website hosting feature to build a single-page application (SPA) at the frontend. For more information, see Tutorial: Use static website hosting to build a single-page application.

9.2. Tutorial: Use a custom domain name to configure static website hosting

You can host a static website on an Object Storage Service (OSS) bucket and allow users to access the website by using the custom domain name such as example.com that is mapped to the bucket. This tutorial applies when you want to host an existing static website or create a website on an OSS bucket from scratch.

Step 1: Register a domain name

Before you build a static website, you must register a domain name for your website. We recommend that you use Alibaba Cloud Domains to register a domain name for your website. For more information, see How to register an Alibaba Cloud domain name.

In this example, `example.com` is used as the domain name of the website.

> **Notice** If you want to map the registered domain name to a bucket that is located in a region in mainland China, you must apply for an ICP filing for the domain name at the Ministry of Industry and Information Technology (MIIT) of China. For more information, see Filing.

Step 2: Create a bucket

You must create a public read bucket to host the static website and store website data.

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click **Create Bucket**.
3. In the **Create Bucket** panel, configure parameters for the bucket. The following table describes the parameters that you must configure.

Parameter	Description
Bucket Name	Specify the bucket name. In this example, specify the bucket name as **examplebucket**.
Region	Select the region where the bucket is located. In this example, select **China (Hangzhou)**.
Storage Class	Select **Standard**.
Access Control List (ACL)	Select **Public Read**.

Retain the default values for other parameters. For more information, see Create buckets.

Step 3: Create and upload web page objects

You must create the default homepage and 404 error page objects for the website and upload the objects to the created bucket.

1. Create two local HTML files.
   - Default homepage file

     In this example, the following content is used to generate the homepage file *index.html* of the static website. Customize the content of the homepage file based on your actual requirements.

```
<html>
   <head>
       <title>Hello OSS!</title>
       <meta charset="utf-8">
   </head>
   <body>
       <P>Configure static website hosting for an OSS bucket</p>
       <P>This is the index page</p>
   </body>
 </html>
```

   - Default 404 error document

     In this example, the following content is used to generate the 404 error document *error.html* of the static website. Customize the content of the 404 error document based on your actual requirements.

```
<html>
<head>
   <title>Hello OSS!</title>
   <meta charset="utf-8">
</head>
<body>
   <P>This is the 404 page</p>
</body>
</html>
```

2. Upload the web page objects to the bucket.
   i. Log on to the OSS console.
   ii. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket to which you want to upload the web page objects.
   iii. In the left-side navigation pane of the page, click **Files**. On the page that appears, click **Upload**.
   iv. In the **Upload** section of the **Upload** panel, click **Upload** and select the two web page objects that you created. Retain the default values for other parameters.

Step 4: Configure static website hosting

1. In the left-side navigation pane, choose **Basic Settings > Static Pages**.
2. Click **Configure**. Set **Default Homepage** to *index.html* and **Default 404 Page** to *error.html*.

Note To redirect access to a subfolder to the *index.html* object in the subfolder, you can enable **Subfolder Homepage**. For more information, see Configure static website hosting.

3. Click **Save**.

Step 5: Map the custom domain name to the bucket

Map the registered custom domain name `example.com` to the examplebucket bucket that you create to use the domain name to access the bucket.

1. On the Overview page of the bucket to which you want to map the custom domain name, choose **Transmission > Domain Names** in the left-side navigation pane.
2. Click **Bind Custom Domain Name**.
3. Enter **example.com** in the **Custom Domain Name** field and turn on **Add CNAME Record Automatically**.
4. Click **OK**.

Step 5: (Optional) Accelerate access to your website by using Alibaba Cloud CDN

You can use Alibaba Cloud CDN to improve the performance of your website. Alibaba Cloud CDN caches the files of your websites, such as HTML files, images, and videos, to data centers around the world. When a visitor requests a file from your website, Alibaba Cloud CDN redirects the request to a copy of the file cached in the data center closest to the region in which the visitor is located. This way, the download is accelerated.

1. On the Overview page of the bucket, choose **Transmission > Domain Names** in the left-side navigation pane.
2. Click **Not Configured** on the right side of the domain name to go to the Alibaba Cloud CDN console.
3. On the **Add Domain Name** page, configure the parameters. The following table describes the parameters.

Parameter	Description
Domain Name to Accelerate	Use the default settings.
Resource Group	Select Default Resource Group.
Business type	Content delivery varies with business types. Select the appropriate business type based on your stored content and the content usage. Select **Image and Small File** in this example.
Origin Info	Use the default settings.
Port	Select the port used by CDN. Select **Port 80** in this example.
Region	Select the region for which you want to accelerate access. Select **Mainland China Only** in this example.

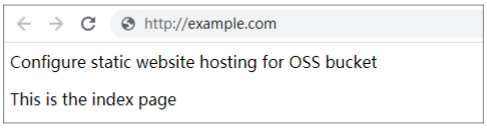
4. Click **Next**, and then click **Return to Domain Name List**.
5. Record the CNAME value of the domain name **example.com.w.kunlunsl.com**. Modify the CNAME record that is added when you map the custom domain name to the bucket.
   i. Log on to the Alibaba Cloud DNS console.
   ii. On the Manage DNS page, click **Configure** in the Actions column that corresponds to the domain name to which you want to add a CNAME record.
   iii. Find the CNAME record that is automatically added when you map the custom domain name. Click **Modify**.
   iv. Change **Value** to **example.com.w.kunlunsl.com**. Retain the default values for other parameters.
   v. Click **OK**.
6. (Optional) On the **Domain Names** tab, turn on **Auto CDN Cache Update**.

   If you want that auto CDN cache update can be triggered by specified operations, you can submit a ticket to apply for the feature. After you are authorized to configure auto CDN cache update for specified operations, click **Supported Operations** in the Auto CDN Cache Update column that corresponds to the domain name and select the operations that can trigger auto CDN cache update. For more information about the supported operations, see Map accelerated domain names.

Step 6: Test the website

Visit the following URLs in the browser to verify whether the website is running properly.

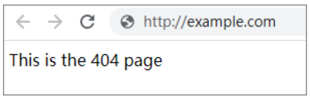
- Visit `http://example.com` to access the homepage of the static website. If static website hosting is correctly configured, a similar page is displayed.



Configure static website hosting for OSS bucket

This is the index page

- Visit `http://example.com/example.txt` to access an object that does not exist in the bucket. If static website hosting is configured, a similar page is displayed.



This is the 404 page

Step 7: Clean up resources

The resources created in this tutorial are used only to test the environment. We recommend that you clean up the created resources after the test is complete to avoid unnecessary fees.

- Delete the domain name that is mapped to the bucket. For more information, see Disable or remove an accelerated domain name.
- Delete the CNAME record that is added in Alibaba Cloud DNS. For more information, see Delete a DNS record.
- Delete created buckets and objects uploaded to the bucket. For more information, see Delete objects and Delete buckets.

9.3. Tutorial: Use static website hosting to build a single-page application

This topic describes how to use React and the static website hosting feature of Object Storage Service (OSS) to build a single-page application (SPA) at the front end.

Background information

A SPA is a web application or website that interacts with the user by dynamically rewriting the current web page with new data from the web server, instead of the default method of a web browser loading entire new pages. Page refresh does not occur in a SPA. This delivers smoother transition and better user experience that is similar to a native application. In a SPA, all necessary HTML, JavaScript, and CSS code is retrieved by the browser with a single page load, or appropriate resources are dynamically loaded and added to the page in response to user actions.

Prerequisites

- OSS SDK for Node.js is installed. For more information, see Installation.
- A bucket is created. In this tutorial, a bucket named examplebucket is used. For more information, see Create buckets.
- The custom domain name `example.com` is mapped to the examplebucket bucket. For more information, see Map custom domain names.

Step 1: Use React to build a SPA

1. Open `Command Prompt` or PowerShell. In this tutorial, `Command Prompt` is used.
2. Run the following command to create a project:

   ```
   npx create-react-app spa-demo
   ```

   The following output is returned:

   ```
   Need to install the following packages:
   create-react-app
   Ok to proceed? (y)
   ```

3. At the `Ok to proceed? (y)` prompt, enter *y* and press the Enter key.

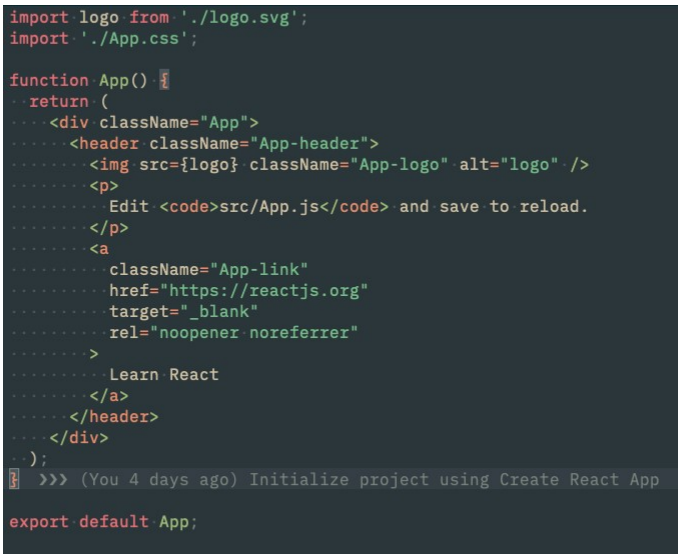
   After several minutes, the project is created. All project dependencies are installed at the same time.
4. Run the following command to go to the created project:

   ```
   cd spa-demo
   ```

5. Run the following command to preview the created project:

   ```
   npm run start
   ```

   The following figure shows the content of the App.js file.



```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}  ››› (You 4 days ago) Initialize project using Create React App

export default App;
```

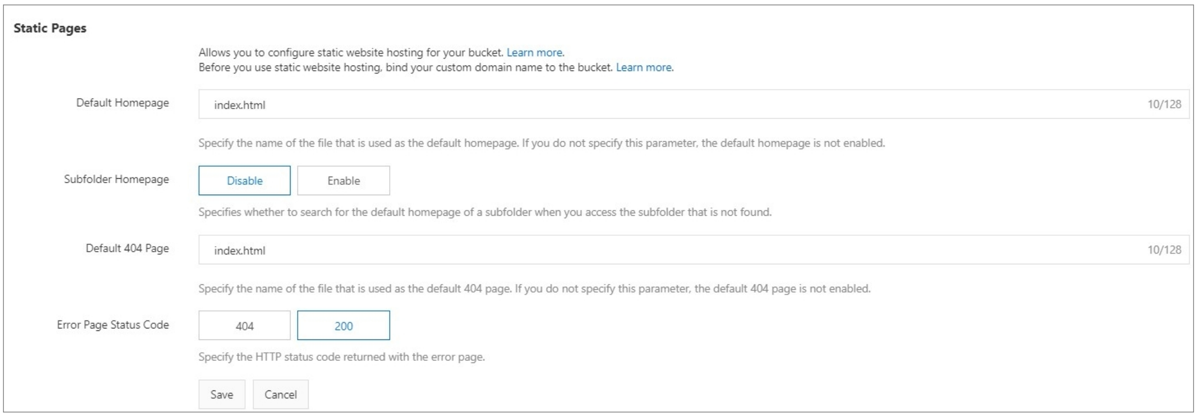
6. Debug the project and then preview the file to confirm that the SPA meets your requirements. Then, run the following command to build the project in the production environment:

   ```
   npm run build
   ```

   The *build* directory is generated in the root directory of the project.

Step 2: Configure static website hosting for the examplebucket bucket

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click **examplebucket**.
3. In the left-side navigation pane, choose **Basic Settings > Static Pages**. Click **Configure** in the **Static Pages** section. Configure the parameters as described in the following table.



Static Pages

Allows you to configure static website hosting for your bucket. Learn more.
Before you use static website hosting, bind your custom domain name to the bucket. Learn more.

Default Homepage: index.html 10/128

Specify the name of the file that is used as the default homepage. If you do not specify this parameter, the default homepage is not enabled.

Subfolder Homepage: Disable / Enable

Specifies whether to search for the default homepage of a subfolder when you access the subfolder that is not found.

Default 404 Page: index.html 10/128

Specify the name of the file that is used as the default 404 page. If you do not specify this parameter, the default 404 page is not enabled.

Error Page Status Code: 404 / 200

Specify the HTTP status code returned with the error page.

Parameter	Description
Default Homepage	Configure the default homepage that appears when you use a browser to access the static website hosted on the OSS bucket. In this example, set this parameter to *index.html*.
Subfolder Homepage	Specify whether to enable subdirectory homepage for the bucket. In this example, select **Disable**. In this case, when you access the root directory of the bucket or a subdirectory whose URL ends with a forward slash (/), the default homepage object in the root directory of the bucket is returned.
Default 404 Page	Specify the error page that is returned when the object that you want to access does not exist in the bucket and a 404 HTTP status code is returned. In this tutorial, the page file specified by this parameter is used as the default error page of the SPA. Set this parameter to *index.html*, which is the same as the value of Default Homepage.
Error Page Status Code	Specify that HTTP status code **200** is returned with the error page.

4. Click **Save**.

Step 3: Upload all files in the build directory

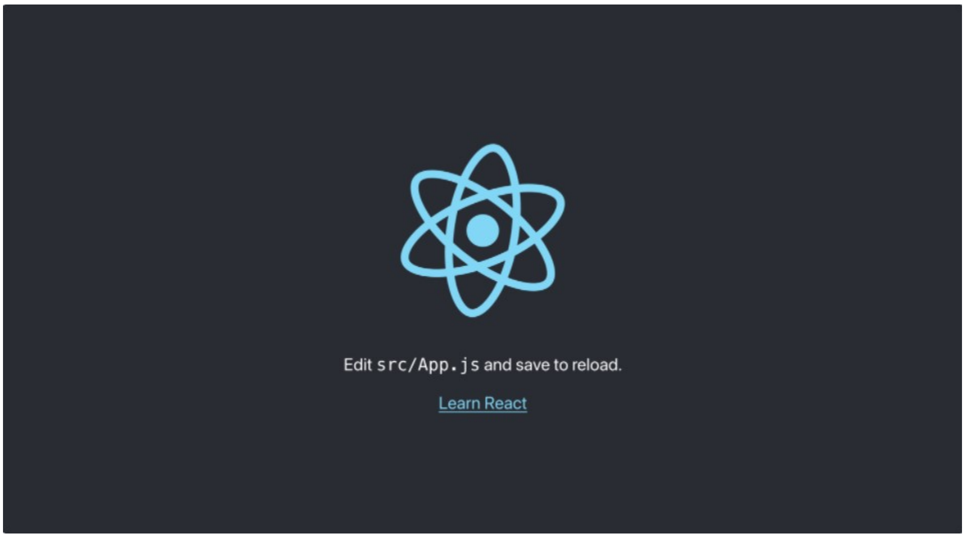
1. In the left-side navigation pane, click **Buckets**. On the Buckets page, click **examplebucket**.
2. In the left-side navigation pane, choose **Files > Files**.
3. On the **Files** page, click **Upload**.
4. On the **Upload** page, configure the parameters as described in the following table.

Parameter	Description
Upload To	Select **Current**.
File ACL	Set the access control list (ACL) of the objects to **Public Read**.
Files to Upload	Click **Select Folders**, and then select all files in the *build* directory of the *spa-demo* project.

5. Click **Upload**.
   You can view the upload progress of files in the **Task List** panel. After the files are uploaded, you can find an object named **index.html** in the examplebucket bucket.

Step 4: Use the custom domain name to access the SPA

1. Open your browser.
2. In this tutorial, the custom domain name *example.com* is used. Enter *https://example.com/index.html* in the browser to access the SPA.
   The following figure shows the returned page.



Edit src/App.js and save to reload.

FAQ

- The web page of the deployed SPA can be rendered after a route transition. However, HTTP status code 404 is returned when I refresh the page. What do I do?

  The default homepage or the default 404 page may be incorrectly configured when you configure static website hosting for the bucket. Make sure that both **Default Homepage** and **Default 404 Page** are set to *index.html*.

- The page is loaded after a route transition. However, HTTP status code 404 instead of 200 is returned. What do I do?

  The Error Page Status Code parameter may be left empty or incorrectly configured when you configure static website hosting for the bucket. Make sure that the **Error Page Status Code** parameter is set to **200**.

10.Data processing and indexing

10.1. Imag

10.1.1. Overview

You can add Image Processing (IMG) parameters to GetObject requests to process image objects stored in Object Storage Service (OSS). For example, you can add image watermarks to images or convert image formats.

Parameters

OSS allows you to directly use one or more parameters to process images. You can also encapsulate multiple IMG parameters in a style to batch process images. For more information about image styles, see 图片样式.

When multiple IMG parameters are specified, OSS processes the image in the order of the parameters. The following table describes the parameters that you can configure to process images.

IMG operation	Parameter	Description
Resize images	resize	Resizes images to a specified size.
Incircle	circle	Crops images based on the center point of images to ellipses of the specified size.
Custom crop	crop	Crops rectangular images of the specified size.
Indexed cut	indexcrop	Cuts images along the specified horizontal or vertical axis and selects one of the images.
Rounded rectangle	rounded-corners	Crops images to rounded rectangles based on the specified rounded corner size.
Auto-rotate	auto-orient	Auto-rotates images for which the auto-orient parameter is configured.
Rotate	rotate	Rotates images clockwise based on the specified angle.
Blur	blur	Blurs images.
Brightness	bright	Adjusts the brightness of images.
Sharpen	sharpen	Sharpens images.
Contrast	contrast	Adjusts the contrast of images.
Gradual display	interlace	Configures gradual display for the JPG images.
Adjust image quality	quality	Adjusts the quality of images in the JPG and WebP formats.
Format conversion	format	Converts image formats.
Add watermarks	watermark	Adds image or text watermarks to images.
Query the average tone	average-hue	Queries the average tone of images.
Query the EXIF data of an image	info	Queries image information, including basic information and EXIF information.

For example, after you configure the `resize` and `quality` parameters for the `example.jpg` source image, the URL for an image named exmaple.jpg is `https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300/quality,q_90` . You can configure Content Delivery Network (CDN) back-to-origin rules to filter or retain the IMG parameters contained in the URLs of images that you want to retrieve. This way, you can retrieve source images or images processed by specifying IMG parameters from OSS.

- Retrieve source images

  You can enable Parameter Filtering for CDN to remove all IMG parameters that follow the question mark (?) in the URL of the image you want to retrieve.In this case, the source image `example.jpg` is retrieved.

- Retrieve processed images

  You can enable Retain Specified Parameters for CDN to retain all IMG parameters that follows the question mark (?) in the URL of the image you want to retrieve.In this case, the processed image is retrieved.

For more information about how to configure CDN back-to-origin rules, see Enable Alibaba Cloud CDN to retain only specified URL parameters and ignore other URL parameters.

Implementation methods

You can use object URLs, API operations, and SDKs to process images. For more information, see IMG implementation modes.

Usage notes

When you use IMG, take note of the following items:

- Limits on source images
  - Only JPG, PNG, BMP, GIF, WebP, TIFF, and images are supported.
  - The size of the source image cannot exceed 20 MB.
  - For the rotate operation, the height or width of the source image cannot exceed 4,096 pixels. For other operations, the width or height of the source image cannot exceed 30,000 pixels, and the total pixel number of the source image cannot exceed 250 million.

    The pixel number of a dynamic image, such as a GIF image, is calculated by using the following formula: `Width × Height × Number of image frames` . The pixel number of a static image, such as a PNG image, is calculated by using the following formula: `Width × Height` .

- Limits on resized images

The width or height of the resized image cannot exceed 16,384 pixels. The total pixel number of the resized image cannot exceed 16,777,216 pixels.

- Limits on image styles

  You can create up to 50 image styles for each bucket. To create more than 50 styles for a bucket, contact the technical support .

Billing

When you use IMG, you are charged the following fees:

- Image processing fees

  You are charged for IMG based on the size of the source images only after the free quota is exceeded. For more information about image processing fees, see Data processing fees.

- Request fees

  A GetObject request is generated each time when you use IMG to process an image. You are charged based on the number of generated requests. For more information about request fees, see .

- Traffic fees

  You are charged for the outbound traffic over the Internet based on the size of the source images. For more information about traffic fees, see Traffic fees.

Release notes

IMG provides two API versions: the later API version and the earlier API version. This topic describes the APIs of the later version. APIs of the earlier version are not updated. For more information about the compatibility between the later and earlier versions of APIs, see FAQ on using old and new versions of APIs and domain names.

10.1.2. IMG implementation modes

You can use object URLs, API operations, or SDKs to process images in Object Storage Service (OSS). This topic describes how to use these methods to process images.

Add parameters to the URL of an image object

To process an image, you can add Image Processing (IMG) parameters or image style parameters to the end of the URL of the image object.

> **Notice** By default, when you use a URL to access an image object in a bucket, the image is downloaded. To ensure that an image object is previewed when you access the image object, you must map a custom domain name to your bucket and add a CNAME record. For more information, see Map custom domain names.

Image processing mode	Add IMG parameters	Add image style parameters
IMG URL	`https://bucketname.endpoint/objectname?x-oss-process=image/action,parame_value`	`https://bucketname.endpoint/objectname?x-oss-process=style/stylename`
Parameter description	• `https://bucketname.endpoint/objectname` : the URL of the image object. For more information about how to obtain the URL of an object, see How do I obtain the URL of an uploaded object?. • `x-oss-process=image/` : the fixed parameter, which indicates that the image object is processed by adding IMG parameters. • `action, param_value` : the action, parameter, and value of an IMG operation. These parameters determine the IMG operation that is used to process the image object. Separate multiple operations with forward slashes (/). OSS processes images in the order of IMG parameters. For example, `image/resize,w_200/rotate,90` indicates that OSS resizes the image to a width of 200 pixels, and rotates the image 90°. For more information about the supported IMG parameters, see Parameters.	• `https://bucketname.endpoint/objectname` : the URL of the image object. For more information about how to obtain the URL of an object, see How do I obtain the URL of an uploaded object?. • `x-oss-process=style/` : the fixed parameter, which indicates that the image object is processed by adding image style parameters. • `stylename` : the name of the style that you set in the OSS console. For more information, see 图片样式. If you specify a custom delimiter, you can use the delimiter to replace `?x-oss-process=style/` to simplify the IMG URL. For example, if you set the delimiter to an exclamation point (!), the URL of the processed image object is `<https://bucketname.endpoint/objectname!stylename` . For more information about how to configure custom delimiters, see Set delimiters.
Examples	https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300/quality,q_90	https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/panda_style

The preceding description applies only to objects that can be anonymously accessed. If your object does not allow anonymous access, you must add the IMG operation to the signature by using SDKs. For more information about OSS SDKs, see the following topics:

- Java SDK
- Python SDK
- PHP SDK
- Go SDK
- C SDK
- C++ SDK
- .NET SDK
- Android SDK
- iOS SDK
- Node.js SDK
- browser.js SDK

Use OSS SDKs to process images

You can configure IMG or image style parameters in SDKs to process images. The following code provides an example on how to use OSS SDK for Java to process images.

- Add IMG parameters

  When you use IMG parameters to process images, separate multiple IMG operations with forward slashes (/).

```
// Set yourEndpoint to the endpoint of the region in which the bucket is located. For example, if the bucket is located in the China (Hangzhou) reg
ion, set yourEndpoint to https://oss-cn-hangzhou.aliyuncs.com.
String endpoint = "yourEndpoint";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API
operations. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a RAM user,
log on to the RAM console.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Specify the name of the bucket in which the image you want to process is stored.
String bucketName = "examplebucket";
// Specify the full path of the image object. The full path of the object cannot contain the bucket name.
String objectName = "exampleobject.jpg";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Resize the image to the height and width of 100 pixels.
String style = "image/resize,m_fixed,w_100,h_100";
GetObjectRequest request = new GetObjectRequest(bucketName, objectName);
request.setProcess(style);
// Name the processed image example-resize.jpg and save the image to your local computer.
// Specify the local path to which you want to save the image. Example: D:\\localpath\\example-resize.jpg. If the specified local file exists, it i
s overwritten by the processed image. Otherwise, the local file is created.
// By default, if you set this parameter to the name of a local file such as example-resize.jpg without specifying the local path of the file, the
processed object is saved to the local path of the project to which the sample program belongs.
ossClient.getObject(request, new File("D:\\localpath\\example-resize.jpg"));
// Shut down the OSSClient instance.
ossClient.shutdown();
```

For more information about the naming conventions for objects and buckets, see Bucket naming conventions and Object naming conventions.

- Add image style parameters

```
// Set yourEndpoint to the endpoint of the region in which the bucket is located. For example, if the bucket is located in the China (Hangzhou) reg
ion, set yourEndpoint to https://oss-cn-hangzhou.aliyuncs.com.
String endpoint = "yourEndpoint";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API
operations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Specify the name of the bucket in which the image you want to process is stored.
String bucketName = "examplebucket";
// Specify the full path of the image object. The full path of the object cannot contain the bucket name.
String objectName = "exampleobject.jpg";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Use the custom style to process the image.
// Set yourCustomStyleName to the name of the image style you create in the OSS console.
String style = "style/yourCustomStyleName";
GetObjectRequest request = new GetObjectRequest(bucketName, objectName);
request.setProcess(style);
// Name the processed image example-new.jpg and save the image to your local computer.
// Specify the local path to which you want to save the image. Example: D:\\localpath\\example-new.jpg. If the specified local file exists, it is o
verwritten by the processed image. Otherwise, the local file is created.
// By default, if you set this parameter to the name of a local file such as example-new.jpg without specifying the local path of the file, the pro
cessed object is saved to the local path of the project to which the sample program belongs.
ossClient.getObject(request, new File("D:\\localpath\\example-new.jpg"));
// Shut down the OSSClient instance.
ossClient.shutdown();
```

For more information about SDK demos for other programming languages, see the following topics:

- Java SDK
- Python SDK
- PHP SDK
- Go SDK
- C SDK
- C++ SDK
- .NET SDK
- Android SDK
- iOS SDK
- Node.js SDK
- browser.js SDK

Call API operations to process images

You can add IMG or image style parameters to GetObject requests to process images.

- Add IMG parameters

  The following code provides a sample request:

```
GET /oss.jpg?x-oss-process=image/resize,w_100 HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 06:38:30 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:UNQDb7GapEgJkcde6OhZ9J*****
```

- Add image style parameters

The following code provides a sample request:

```
GET /oss.jpg?x-oss-process=style/styleexample HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 06:40:10 GMT
Authorization: OSS qn6qrrawuk53oqxo2otfjbyc:UNapEgQDb7GJkcde6OhZ9J*****
```

Save processed images

By default, IMG does not save processed images. However, OSS allows you to add the saveas parameter in an IMG request to save the processed image as an object to a specified bucket. For more information, see Save processed images.

10.1.3. IMG parameters

10.1.3.1. Resize images

You can use the resize parameters to adjust the size of images stored in Object Storage Service (OSS). This topic describes the parameters used to resize images and provides examples on how to resize images.

Parameters

Operation: `resize`

The following table describes the parameters that you can configure when you resize images.

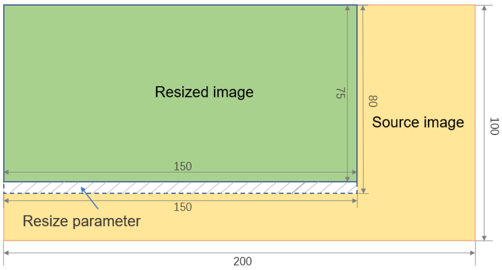
- Resize an image based on the specified height and width

Parameter	Required	Description	Valid value
m	Yes	Specifies the type of the resize operation. Default value: lfit.	○ lfit: OSS resizes the source image proportionally as large as possible within a rectangle based on the specified width and height. ○ mfit: OSS resizes the source image proportionally as small as possible beyond a rectangle based on the specified width and height. ○ fill: OSS resizes the source image proportionally as small as possible beyond a rectangle, and then crops the resized image from the center based on the specified width and height. ○ pad: OSS resizes the source image as large as possible within a rectangle based on the specified width and height, and fills the empty space by using the specified color. ○ fixed: OSS forcibly resizes the source image based on the specified width and height. For more information, see the examples below the table. Note When you set the parameter to lfit or mfit to proportionally resize an image, the ratio of w/h of the source image is rounded to numbers if the ratio is a decimal.
w	Yes	Specifies the width to which the image is to be resized.	[1,4096]
h	Yes	Specifies the height to which the image is to be resized.	[1,4096]
l	Yes	Specifies the length of the longer side to which the image is to be resized. Note Longer side refers to the side with the larger ratio of the source length to the requested length. Shorter side refers to the side with the smaller ratio of the source length to the requested length. For example, if a source image is resized from 400 × 200 pixels to 800 × 100 pixels, the source-to-requested ratios are 0.5 (400/800) and 2 (200/100). Therefore, 0.5 is smaller than 2, the 200-pixel side is the longer side, and the 400-pixel side the shorter side.	[1,4096]
s	Yes	Specifies the length of the shorter side to which the image is to be resized.	[1,4096]
limit	No	Specifies whether to perform the resize operation when the resized image is larger than the source image.	0 and 1. Default value: 1. ○ 1: OSS returns the source image without resizing the image. ○ 0: OSS resizes the source image based on the specified parameter.
color	Yes (only when the value of `m` is pad)	When you set the resize type to pad, you can select a color to fill the empty space.	RGB color values. For example, 000000 indicates black, and FFFFFF indicates white. Default value: FFFFFF (white).

Example: The size of the source image is 200 × 100 pixels. The w parameter is set to 150 pixels, and the h parameter is set to 80 pixels. The source image is resized to different sizes when you specify different resize types.

- lfit
  - Proportional resizing: The value of w/h of the source image must be equal to that of the resized image. Therefore, if w is 150 pixels, h is 75 pixels. If h is 80 pixels, w is 160 pixels.
  - Maximum image within a rectangle based on the specified width and height: The value of w*h of the resized image cannot exceed 150 × 80 pixels.

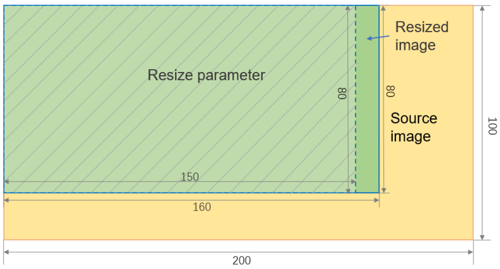
  The size of the resized image is 150 × 75 pixels based on the preceding conditions.



lfit

- mfit
  - Proportional resizing: The value of w/h of the source image must be equal to that of the resized image. Therefore, if w is 150 pixels, h is 75 pixels. If h is 80 pixels, w is 160 pixels.
  - Minimum image beyond a rectangle based on the specified width and height: The resized image must be a minimum rectangle whose size is greater than 150 × 80 pixels.

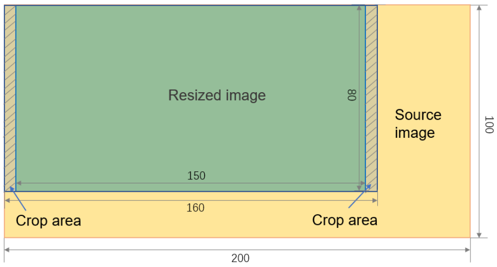
  The size of the resized image is 160 × 80 pixels based on the preceding conditions.



mfit

- fill

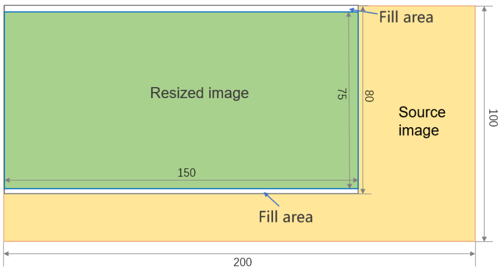
  The fill parameter is used to resize the source image proportionally as small as possible beyond a rectangle and crop the resized image based on the specified width and height. The source image is resized to 160 × 80 pixels, and w is cropped to 150 pixels from the center to obtain a resized image of 150 × 80 pixels.



fill

- pad

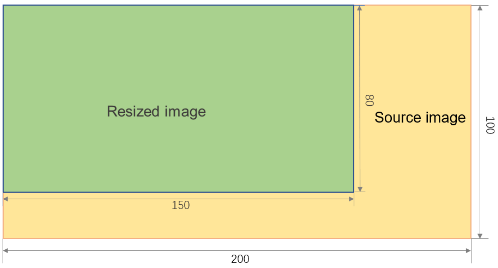
  The pad parameter is used to resize the source image as large as possible within a rectangle and fill the empty space based on the specified width and height. The source image is resized to 150 × 75 pixels, and h is centered and filled to 80 pixels to obtain a resized image of 150 × 80 pixels.



pad

- fixed

  The fixed parameter is used to resize the image based on the specified width and height. If the width and height ratio of the image is different from that of the source image, the image is deformed.



fixed

- Resize an image proportionally

Parameter	Required	Description	Valid value
p	Yes	OSS resizes an image by percentage.	[1,1000] A percentage smaller than 100 indicates that the image is scaled down. A percentage greater than 100 indicates that the image is scaled up.

Usage notes

- Limits on source images
  - Only JPG, PNG, BMP, GIF, WebP, and TIFF images are supported. GIF images can be resized based on the specified width and height but cannot be resized proportionally. GIF images become static images when you resize the images proportionally.
  - The size of the source image cannot exceed 20 MB.
  - The width or height of the source image cannot exceed 30,000 pixels. The total pixel number of the source image cannot exceed 250 million.

    The total pixel number of a dynamic image, such as a GIF image, is calculated by using the following formula: `Width × Height × Number of image frames`. The total pixel number of a static image, such as a PNG image, is calculated by using the following formula: `Width × Height`.
- Limits on resized images

  The width or height of a resized image cannot exceed 16,384 pixels. The total pixel number of the resized image cannot exceed 16,777,216.
- If the width or height of the resized image is specified:
  - The source image is resized proportionally when proportional resizing is performed. For example, if you resize the height of a source image from 200 × 100 pixels to 100 pixels, the width of the source image is resized to 50 pixels.
  - The source image is resized based on the specified width and height. For example, if you resize the height of a source image from 200 × 100 pixels to 100 pixels, the width of the source image is resized to 100 pixels.
- If you set the m parameter to mfit and specify a value for *l* or *s*, the longer side or the shorter side of the image is scaled based on the specified value of *l* or *s*.
- After you set the *m* parameter, the specified values of *l* and *s* do not take effect if you specify a value for *w* or *h*.
- By default, if the size of the resized image is larger than that of the source image, the source image is returned. You can add the `limit_0` parameter to enlarge the image. The URL used to enlarge the image is in the following format: `https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_500,limit_0`.

Examples

An image in the bucket named image-demo in the China (Hangzhou) region is used in the following examples. The following URL is used to access the image over the Internet:

https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg

- Resize the image proportionally

- Based on the width or height

  Configure parameters to resize the image:

  - Resize the source image to a height of 100 pixels: `resize,h_100`
  - Set the resize type to lfit: `m_lfit`

  The URL used to process the image is in the following format: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,h_100,m_lfit

- Based on the longer side

  Resize the source image based on the longer side of 100 pixels: `resize,l_100`

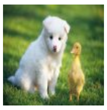
  The following URL is used to process the image: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,l_100

- Resize the image based on the specified width and height

  Configure parameters to resize the image:

  - Resize the source image to a width and a height of 100 pixels: `resize,h_100,w_100`
  - Set the resize type to fixed: `m_fixed`

  The URL used to process the image is in the following format: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,m_fixed,h_100,w_100

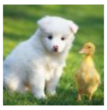
  

- Crop the image based on the specified width and height

  Configure parameters to resize the image:

  - Resize the source image to a width and a height of 100 pixels: `resize,h_100,w_100`
  - Set the resize type to fill: `m_fill`

  The URL used to process the image is in the following format: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,m_fill,h_100,w_100

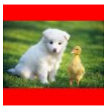
  

- Resize the source image based on the specified width and height and fill the empty space

  Configure parameters to resize the image:

  - Resize the source image to a width and a height of 100 pixels: `resize,h_100,w_100`
  - Set the resize type to pad: `m_pad`
  - Fill the empty space in red: `color_FF0000`

  The URL used to process the image is in the following format: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,m_pad,h_100,w_100,color_FF0000

- Resize an image proportionally

  Configure parameters to resize the image:

  Resize the source image by 50%: `resize,p_50`

  The following URL is used to process the image: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,p_50

FAQ

How do I access a resized image if the access control list (ACL) of the image is private?

To access the image, you must add signature information to the URL of the resized image. For more information about how to obtain the URL of an object, see How do I obtain the URL of an uploaded object?.

10.1.3.2. Add watermarks

You can add text or image watermarks to an image stored in Object Storage Service (OSS) by using Image Processing (IMG) parameters. This topic describes and provides examples on how to use parameters to add watermarks to an image.

Usage notes

- Only images stored in the current bucket can be used as watermarks. To use online or local images as watermarks, you must first upload the images to the current bucket.
- Only JPG, PNG, BMP, WebP, and TIFF images can be used as watermarks.
- You can add up to three different image watermarks to a single image, and the positions of each image watermark cannot be completely overlapped.
- Traditional Chinese characters cannot be used as text watermarks.

Parameters

Operation: watermark

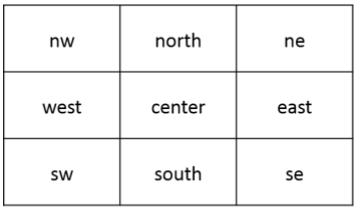
The following table lists the parameters that you can configure when you add watermarks to images.

- Basic parameters

Parameter	Required	Description	Valid value
t	No	The opacity of the watermark.	[0,100] Default value: 100. A value of 100 indicates that the watermark is opaque.
g	No	The position of the watermark on the image.	○ nw: upper left ○ north: upper middle ○ ne: upper right ○ west: middle left ○ center: center ○ east: middle right ○ sw: lower left ○ south: lower middle ○ se: lower right For the precise position that each value indicates, see the figure in the following section.
x	No	The horizontal margin is the horizontal distance between the watermark and the image edge. This parameter takes effect only when the watermark is on the upper left, middle left, lower left, upper right, middle right, or lower right of the image.	[0,4096] Default value: 10 Unit: px
y	No	The vertical margin is the vertical distance between the watermark and the image edge. This parameter takes effect only when the watermark is on the upper left, upper middle, upper right, lower left, lower middle, or lower right of the image.	[0,4096] Default value: 10 Unit: px
voffset	No	The vertical offset from the middle line. When the watermark is in the middle left, center, or middle right of the image, you can designate the vertical offset of the watermark along the middle line.	[-1000,1000] Default value: 0. Unit: px

You can use the horizontal margin, vertical margin, and vertical offset from the middle line to adjust the position of a watermark on an image. You can also use these parameters to adjust the watermark layout when the image has multiple watermarks.

The following figure shows the positions of watermarks based on coordinates.



nw	north	ne
west	center	east
sw	south	se

- Image watermark parameters

Parameter	Required	Description	Valid value

Parameter	Required	Description	Valid value
image	Yes	The complete name of the image object that you want to use as a watermark. The object name must be Base64-encoded. For more information, see Encode watermarks. For example, if you want to use an image object named *panda.png* in the *image* directory of the current bucket as a watermark, the object name to encode is *image/panda.png* and the encoded object name is `aW1hZ2UvcGFuZGEucG5n` . Note Only objects stored in the current bucket can be used as watermarks.	Base64-encoded strings.

- Parameters for watermark image preprocessing

You can preprocess a watermark image by calling the Resize images, Custom crop, Indexed cut, Rounded rectangle, and Rotate operations. In addition, the P parameter is supported when you resize a watermark image.

Parameter	Description	Valid value
P	The size of the watermark image relative to the source image. The value of this parameter specifies the size of the watermark as a percentage of the source image. For example, if you set this parameter to 10 for a source image of 100 × 100 pixels, the size of the watermark image is 10 × 10 pixels. If the source image is 200 × 200 pixels, the size of the watermark image is 20 × 20 pixels.	[1,100]

- Text watermark parameters

Parameter	Required	Description	Valid value
text	Yes	The content of the text watermark. The text content must be Base64-encoded. For more information, see Encode watermarks.	Base64-encoded strings. The string can be up to 64 characters in length.
type	No	The font of the text watermark. The font name must be Base64-encoded.	For more information about the supported fonts and the encoded strings for the fonts, see Font types and encoded strings. Default value: wqy-zenhei (encoded value: d3F5LXplbmhlaQ)
color	No	The color of the text watermark. The valid values for this parameter are RGB color values.	RGB color values. For example, 000000 indicates black, and FFFFFF indicates white. Default value: 000000. A value of 000000 indicates that the color of the text is black.
size	No	The size of the text watermark.	(0,1000] Default value: 40. Unit: px
shadow	No	The opacity of the shadow for the text watermark.	[0,100] Default value: 0. A value of 0 indicates no shadows are added to the text.
rotate	No	The degree by which the text is rotated clockwise.	[0,360] Default value: 0. A value of 0 indicates that the text is not rotated.
fill	No	Specifies whether to tile the source image with the text watermarks.	0 and 1. Default value: 0. ○ *1*: The source image is tiled with the text watermarks. ○ *0*: The source image is not tiled with the text watermarks.

The following table describes the valid values of the type parameter and the encoded strings of these values.

Parameter value	Font name	Encoded value
wqy-zenhei	WenQuanYi Zen Hei	d3F5LXplbmhlaQ
wqy-microhei	WenQuanYi Micro Hei	d3F5LW1pY3JvaGVp
fangzhengshusong	Fangzheng Shusong	ZmFuZ3poZW5nc2h1c29uZw
fangzhengkaiti	Fangzheng Kaiti	ZmFuZ3poZW5na2FpdGk
fangzhengheiti	Fangzheng Heiti	ZmFuZ3poZW5naGVpdGk

Parameter value	Font name	Encoded value
fangzhengfangsong	Fangzheng Fangsong	ZmFuZ3poZW5nZmFuZ3Nvbmc
droidsansfallback	DroidSansFallback	ZHJvaWRzYW5zZmFsbGJhY2s

- Text-and-image watermark parameters

Parameter	Required	Description	Valid value
order	No	The order of the text watermark and image watermark.	0 and 1. Default value: 0. ○ 0: The image watermark is on the top of the text watermark. ○ 1: The text watermark is on the top of the image watermark.
align	No	The alignment of the text watermark and image watermark.	0, 1, and 2. Default value: 2. ○ 0: The text watermark and the image watermark are aligned based on top alignment. ○ 1: The text watermark and the image watermark are aligned based on center alignment. ○ 2: The text watermark and the image watermark are aligned based on bottom alignment.
interval	No	The spacing between the text watermark and image watermark.	[0,1000] Default value: 0. Unit: px

Encode watermarks

The content, color, and font of a text watermark and the image name of an image watermark must be a URL-safe string that is Base64-encoded. Perform the following steps to encode watermarks:

1. Encode the content by using Base64.
2. Replace part of the encoded string.
   - Replace the plus signs (+) in the encoded string with hyphens (-).
   - Replace the forward slashes (/) in the encoded string with underscores (_).
   - Omit the equal signs (=) that are at the end of the encoded string.

We recommend that you use URL-safe Baes64 encoding tools to encode the content, color, and font of a text watermark and the image name of an image watermark.

Notice Encoded strings can be used only as parameters in specific watermark operations. Do not use encoded strings in signature strings.

Example 1: Add a text watermark to an image.

In the following examples, the source image named example.jpg is stored in a bucket named image-demo that is located in the China (Zhangjiakou) region. The URL used to access the image is https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg.

The following examples show how to add a text watermark to example.jpg:

- Add the string "Hello World" to the image as a text watermark

Base64-encode the string "Hello World" and convert the encoded result to a URL-safe string. For more information, see Encode watermarks. The encoded URL-safe string is `SGVsbG8gV29ybGQ`. Therefore, you can use the following URL to add the text watermark to example.jpg: https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg?x-oss-process=image/watermark,text_SGVsbG8gV29ybGQ.

- Configure multiple IMG parameters when you add a text watermark to the image

  In this example, IMG parameters are configured to perform the following operations on the source image and text watermark "Hello World" that you want to add to the source image:

  - Resize the source image *example.jpg* to 300 × 300 pixels. IMG parameter: `resize,w_300,h_300`
  - Set the text font of the text watermark to WenQuanYi Zen Hei. IMG parameter: `type_d3F5LXplbmhlaQ` (d3F5LXplbmhlaQ is the Base64-encoded value for WenQuanYi Zen Hei.)
  - Add the string "Hello World" to the source image as a text watermark. IMG parameter: `text_SGVsbG8gV29ybGQ`
  - Set the color of the text watermark to white and the size of the text to 30 pixels. IMG parameter: `color_FFFFFF,size_30`
  - Set the opacity of the shadow of the text watermark to 50%. IMG parameter: `shadow_50`
  - Set the position of the text watermark to lower right, the horizontal margin to 10 pixels, and the vertical offset from the middle line to 10 pixels. IMG parameter: `g_se,x_10,y_10`

  You can use the following URL to configure multiple IMG parameters when you add the text watermark to the image: https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300,h_300/watermark,type_d3F5LXplbmhlaQ,size_30,text_SGVsbG8gV29ybGQ,color_FFFFFF,shadow_50,t_100,g_se,x_10,y_10

Example 2: Add an image watermark to the source image.

The following examples show how to add image watermarks to the source image named example.jpg:

- Add an image named panda.png to the source image as an image watermark

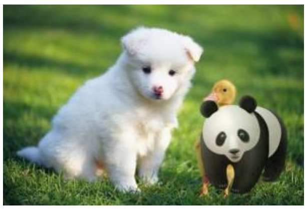
  Base64-encode the image name panda.png and convert the encoded result to a URL-safe string. The encoded URL-safe string is `cGFuZGEucG5n`. Therefore, you can use the following URL to add panda.png to example.jpg as an image watermark: https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg?x-oss-process=image/watermark,image_cGFuZGEucG5n.

- Configure multiple IMG parameters when you add an image watermark to the source image

  In this example, IMG parameters are configured to perform the following operations on the source image and the image named panda.png that you want to add to the source image as an image watermark:

  - Resize the source image example.jpg to 300 × 300 pixels. IMG parameter `resize,w_300,h_300`
  - Set the quality of the source image example.jpg to 90%. IMG parameter: `quality,q_90`
  - Add the image watermark *panda.png*. IMG parameter: `watermark,image_cGFuZGEucG5n` (cGFuZGEucG5n is the Base64-encoded value for panda.png.)
  - Set the opacity of the image watermark to 90%: `t_90`
  - Set the position of the image watermark to lower right, the horizontal margin to 10 pixels, and the vertical offset from the middle line to 10 pixels. IMG parameter: `g_se,x_10,y_10`

You can use the following URL to configure multiple IMG parameters when you add the image watermark to the source image: https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300,h_300/quality,q_90/watermark,image_cGFuZGEucG5n,t_90,g_se,x_10,y_10

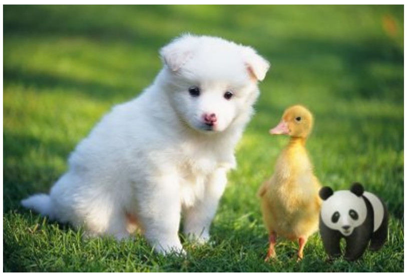


- Preprocess the image watermark and add it to the source image

  In this example, IMG parameters are configured to perform the following operations on the source image and the image named panda.png that you want to add to the source image as an image watermark:

  - Resize the width of the source image *example.jpg* to 300 pixels. IMG parameter: `resize,w_300`
  - Resize the image watermark *panda.png* to 30% of the original size. IMG parameter: `image_cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA` (`cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA` is the Base64-encoded value for `panda.png?x-oss-process=image/resize,P_30`.)
  - Set the opacity of the image watermark to 90%, the position of the image watermark to lower right, the horizontal margin to 10 pixels, and the vertical offset from the middle line to 10 pixels. IMG parameter: `t_90,g_se,x_10,y_10`

  You can use the following URL to configure multiple IMG parameters when you add the image watermark to the source image: https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300/watermark,image_cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA,t_90,g_se,x_10,y_10

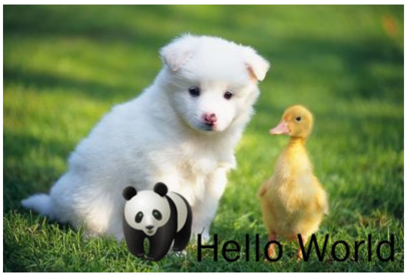


Example 3: Add text and image watermarks to the source image.

The following examples show how to add text and image watermarks to the source image named example.jpg:

- Add an image named panda.png to example.jpg as an image watermark and a string "Hello World" to example.jpg as a text watermark

  Based on the encoded results of the image name panda.png and the string "Hello World", you can use the following URL to add panda.png and "Hello World" to example.jpg as an image watermark and a text watermark: https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg?x-oss-process=image/watermark,image_cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA,text_SGVsbG8gV29ybGQ.



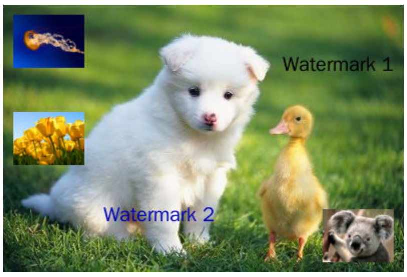
- Add multiple image and text watermarks to example.jpg

  In this example, two text watermarks (Watermark 1 and Watermark 2) and three image watermarks (Jellyfish.jpg, Koala.jpg, and Tulips.jpg) are added to the source image example.jpg. When you add multiple watermarks to the source image, use forward slashes (/) to separate the operations performed to configure each watermark.

  - Add the image Jellyfish.jpg to the source image as an image watermark. Resize the image watermark to 20% of the original size. Set the position of the image watermark to upper left, the horizontal margin to 10 pixels, and the vertical offset from the middle line to 10 pixels. You can configure the following IMG parameters to perform the preceding operations: `watermark,image_cGljcy9KZWxseWZpc2guanBnP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMjA,g_nw,x_10,y_10`. `cGljcy9KZWxseWZpc2guanBnP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMjA` is the Base64-encoded value of the image name Jellyfish.jpg.
  - Add the image Koala.jpg to the source image as an image watermark. Resize the image watermark to 20% of the original size. Set the position of the image watermark to lower right, the horizontal margin to 10 pixels, and the vertical offset from the middle line to 10 pixels. You can configure the following IMG parameters to perform the preceding operations: `watermark,image_cGljcy9Lb2FsYS5qcGc_eC1vc3MtcHJvY2Vzcz1pbWFnZS9yZXNpemUsUF8yMA,g_se,x_10,y_10`. `cGljcy9Lb2FsYS5qcGc_eC1vc3MtcHJvY2Vzcz1pbWFnZS9yZXNpemUsUF8yMA` is the Base64-encoded value of the image name Koala.jpg.
  - Add the image Tulips.jpg to the source image as an image watermark. Resize the image watermark to 20% of the original size. Set the position of the image watermark to middle left, the horizontal margin to 10 pixels, and the vertical offset from the middle line to 10 pixels. You can configure the following IMG parameters to perform the preceding operations: `watermark,image_cGljcy9UdWxpcHMuanBnP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMjA,g_west,x_10,y_10`. `cGljcy9UdWxpcHMuanBnP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMjA` is the Base64-encoded value of the image name Tulips.jpg.
  - Add the text watermark "Watermark 1" to the source image. Set the size of the text to 20 pixels. Set the position of the text watermark to upper right, the horizontal margin to 10 pixels, and the vertical offset from the middle line to 200 pixels. You can configure the following IMG parameters to perform the preceding operations: `watermark,text_V2F0ZXJtYXJrIDE,g_ne,size_20,x_10,y_200`. `V2F0ZXJtYXJrIDE` is the Base64-encoded value of the string "Watermark 1".

- Add the text watermark "Watermark 2" to the source image. Set the size of the text to 20 pixels and the color of the text to dark blue. Set the position of the text watermark to lower left, the horizontal margin to 100 pixels, and the vertical offset from the middle line to 50 pixels. You can configure the following IMG parameters to perform the preceding operations: `watermark,text_V2F0ZXJtYXJrIDI,color_0000b7,size_20,g_sw,x_100,y_50` . `V2F0ZXJtYXJrIDI` is the Base64-encoded value of the string "Watermark 2".

Based on the preceding IMG parameters, you can use the following URL to add two text watermarks (Watermark 1 and Watermark 2) and three image watermarks (Jellyfish.jpg, Koala.jpg, and Tulips.jpg) to the source image example.jpg: https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg?x-oss-process=image/watermark,image_cGljcy9KZWxseWZpc2guanBnP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMjA,g_nw,x_10,y_10/watermark,image_cGljcy9Lb2FsY



FAQ

How do I use online or local images as watermark images?

When IMG is used to add image watermarks to a source image, make sure that the watermark images and the source image are stored in the same bucket. To use online or local images to a source image as watermarks, you must first upload the images to the bucket in which the source image is stored.

10.1.3.3. Custom crop

You can use custom crop parameters to crop a rectangular image based on a specified size from a source image stored in OSS. This topic describes the parameters and examples to crop an image based on a specified dimension.

Parameters

Operation name: crop

The following table lists the parameters.

Parameter	Description	Valid value
w	The width of the cropped area.	[0, image width]. Default value: the maximum value.
h	The height of the cropped area.	[0, image height]. Default value: the maximum value.
x	The abscissa of the starting point. By default, the origin is located in the upper-left corner.	[0, image bound]
y	The ordinate of the starting point. By default, the origin is located in the upper-left corner.	[0, image bound]
g	The location of the origin for cropping. The origin is located in the upper-left corner of any of the nine-cell matrix.	• nw • north • ne • west • center • east • sw • south • se The following schematic view shows the available locations for the origin.

The following schematic view shows the available locations for the origin.

nw	north	ne
west	center	east
sw	south	se

Usage notes

Before you crop an image, take note of the following items:

- If the specified starting abscissa or ordinate values exceed those of the source image, the system returns `BadRequest` and the following error message: Advance cut's position is out of image.

- If the width and height specified from the starting point exceed those of the source image, the source image is cropped to its boundaries.

Examples

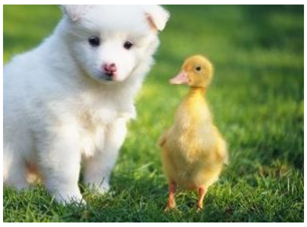
The image-demo bucket that is located in the China (Hangzhou) region is used as an example. Public endpoint of the image: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg

- Crop an image from the starting point (100, 50) to the bounds

  Requirements and parameters:

  - From the starting point (100, 50): `crop,x_100,y_50`
  - To the bounds: By default, the maximum values of w and h are used for cropping. Therefore, the w and h parameters can be omitted.

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/crop,x_100,y_50

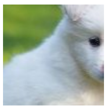
  

- Crop an area of 100 × 100 pixels from the starting point (100, 50)

  Requirements and parameters:

  - From the starting point (100, 50): `crop,x_100,y_50`
  - An area of 100 × 100 pixels: `w_100,h_100`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/crop,x_100,y_50,w_100,h_100

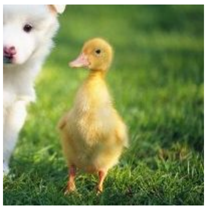
  

- Crop an area of 200 × 200 pixels in the lower-right corner of the source image

  Requirements and parameters:

  - From the starting point in the lower-right corner of the source image: `crop,g_se`
  - An area of 200 × 200 pixels: `w_200,h_200`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/crop,w_200,h_200,g_se

- Crop an area of 200 × 200 pixels in the lower-right corner of an image and stretch the cropped area downward by (10, 10)

  Requirements and parameters:

  - From the starting point in the lower-right corner of an image and stretch the cropped area downward by (10, 10): `crop,g_se,x_10,y_10`
  - An area of 200 × 200 pixels: `w_200,h_200`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/crop,x_10,y_10,w_200,h_200,g_se

10.1.3.4. Adjust image quality

The quality adjustment operation uses the format of a source image to compress the image. You can use the quality adjustment parameters to modify the quality of source images stored in Object Storage Service (OSS). This topic describes the parameters and examples for image quality adjustment.

Quality adjustment applies only to JPG and WebP images.

Parameters

Operation name: quality

The following table describes the parameters that you can configure when you adjust the quality of an image.

Parameter	Description	Valid value
q	Specifies the relative quality of the image and compresses the source image based on percentage. If the source image quality is 100%, you can obtain an image whose quality value is 90% after you add the `quality,q_90` parameter. If the quality value of the source image is 80%, you can obtain an image whose quality value is 72% after you add the `quality,q_90` parameter. **Note** The q parameter applies only to source images in the JPG format to specify the relative quality of the images. If a source image is in the WebP format, this parameter works the same as Q. The absolute quality is specified for the image.	[1,100]
Q	Specifies the absolute quality of the image and compresses the source image based on Q%. If the quality value of the source image is smaller than the specified Q value, the quality value of the compressed image is the quality value of the source image. For example, if the quality value of the source image is 95%, you can obtain an image that has a quality value of 90% after you add `quality,Q_90` . If the quality value of the source image is 80%, you can obtain an image that has a quality value of 80% after you add `quality,Q_90` . **Note** The Q parameter applies only to JPG and WebP images.	[1,100]

Examples

An image in the bucket named image-demo in the China (Hangzhou) region is used in the following examples. The following URL is used to access the image over the Internet:

https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg

- Adjust the relative quality of an image

  Configure the parameters based on the following requirements:

  - Resize the image to a width of 100 pixels: `resize,w_100`
  - Set the relative quality value of the image to 80%: `quality,q_80`

The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/quality,q_80

- Adjust the absolute quality of an image

  Configure the parameters based on the following requirements:

  - Resize the image to a width of 100 pixels: `resize,w_100`
  - Set the absolute quality value of the image to 80%: `quality,Q_80`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/quality,Q_80

10.1.3.5. Format conversion

You can use format conversion parameters to convert the format of a source image stored in OSS. This topic describes the parameters and examples to convert the format of an image.

Parameters

Operation name: format

The following table lists the parameters.

Valid value	Description
jpg	Saves the source image in the JPG format. By default, OSS fills the transparent area in white if the source image is in the PNG, WebP, or BMP format that supports alpha channels.
png	Saves the source image in the PNG format.
webp	Saves the source image in the WebP format.
bmp	Saves the source image in the BMP format.
gif	Saves the source image in the GIF format. If the source image is not in the GIF format, it is saved in the format of the source image.
tiff	Saves the source image in the TIFF format.

Usage notes

- When you perform a standard resize operation on an image, we recommend that you add the format parameter to the end of the last Image Processing (IMG) parameter.

  Example: `image/resize,w_100/format,jpg`

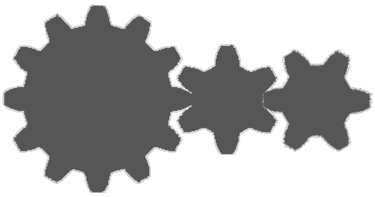
- When you resize and watermark an image, we recommend that you add the format parameter to the end of the resize parameter.

  Example: `image/reisze,w_100/format,jpg/watermark,...`

Examples

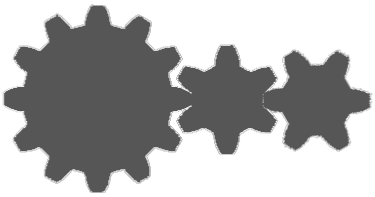
The image-demo bucket that is located in the China (Hangzhou) region is used as an example. Public endpoint of the image:

https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.gif



- Convert the format of the source image to PNG

The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.gif?x-oss-process=image/format,png

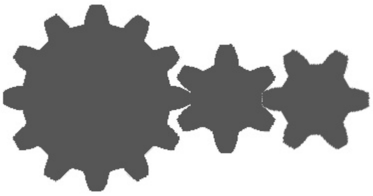


- Convert the format of the source image to JPG that supports gradual display

  Requirements and parameters:

  - Set the image to gradual display: `interlace,1`
  - Convert the format of the image to JPG: `format,jpg`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.gif?x-oss-process=image/interlace,1/format,jpg



- Resize the image to a height of 200 pixels and convert the format of the image to WebP

  Requirements and parameters:

  - Resize the image to a height of 200 pixels: `resize,w_200`
  - Convert the format of the image to WebP: `format,webp`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.gif?x-oss-process=image/resize,w_200/format,webp

10.1.3.6. Query the EXIF data of an image

Some images may contain Exchangeable Image File Format (EXIF) data that includes the attribute information and photographic information of the images. If you want to obtain the EXIF data of an image, add the info parameter to the URL of the image.

> **Note** The EXIF data of an image includes the image information such as the compression ratio, orientation, horizontal resolution, and vertical resolution. For more information about EXIF data, see EXIF2.31.

Parameters

Operation name: info

The image information is returned in the JSON format.

Examples

- Query an image that does not contain EXIF data

  http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/info

  If you add the info parameter to the URL of an image that does not contain EXIF data, only the basic information about the image is returned, such as the size, format, height, and width of the image.

```
{
  "FileSize": {"value": "21839"},
  "Format": {"value": "jpg"},
  "ImageHeight": {"value": "267"},
  "ImageWidth": {"value": "400"}
}
```

- Query an image that contains EXIF data

  http://image-demo.oss-cn-hangzhou.aliyuncs.com/f.jpg?x-oss-process=image/info

  If you add the info parameter to the URL of an image that contains EXIF data, the basic information about the image and the EXIF data of the image are returned.

```
{
  "Compression": {"value": "6"},
  "DateTime": {"value": "2015:02:11 15:38:27"},
  "ExifTag": {"value": "2212"},
  "FileSize": {"value": "23471"},
  "Format": {"value": "jpg"},
  "GPSLatitude": {"value": "0deg "},
  "GPSLatitudeRef": {"value": "North"},
  "GPSLongitude": {"value": "0deg "},
  "GPSLongitudeRef": {"value": "East"},
  "GPSMapDatum": {"value": "WGS-84"},
  "GPSTag": {"value": "4292"},
  "GPSVersionID": {"value": "2 2 0 0"},
  "ImageHeight": {"value": "333"},
  "ImageWidth": {"value": "424"},
  "JPEGInterchangeFormat": {"value": "4518"},
  "JPEGInterchangeFormatLength": {"value": "3232"},
  "Orientation": {"value": "7"},
  "ResolutionUnit": {"value": "2"},
  "Software": {"value": "Microsoft Windows Photo Viewer 6.1.7600.16385"},
  "XResolution": {"value": "96/1"},
  "YResolution": {"value": "96/1"}}
```

10.1.3.7. Auto-rotate

You can use the auto-orient parameter to specify whether source images stored in Object Storage Service (OSS) are rotated based on auto-rotate configurations. This topic describes the parameters and examples to rotate images when you configure auto-rotate.

Parameters

Operation name: auto-orient

The following table describes the parameters that you can configure when you configure auto-rotate.

Parameter	Description	Example
[value]	Specifies whether to perform auto-rotate.	0 and 1. Default value: 1. • 0: The orientation of the source image is retained. • 1: OSS performs auto-rotate on the image.

Usage notes

- If the source image does not have rotation parameters, the operation that you perform to set the auto-orient parameter does not affect the image.
- Most tools can be used to perform auto-rotate on images that have rotation parameters. Therefore, the images you view may be automatically rotated.
- Images that are processed by using auto-orient are re-compressed, which results in size differences between the processed images and the source images.

Examples

An image in the bucket named image-demo in the China (Hangzhou) region is used in this example. The following URL is used to access the image over the Internet:

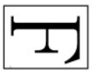
https://image-demo.oss-cn-hangzhou.aliyuncs.com/f.jpg

- Resize the image and retain the orientation

  Configure parameters to resize the image:

  - Resize the image to a width of 100 pixels: `resize,w_100`
  - Disable auto-rotate: `auto-orient,0`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/f.jpg?x-oss-process=image/resize,w_100/auto-orient,0

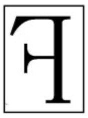
  

- Resize and automatically rotate the image

  Configure parameters to resize the image:

  - Resize the image to a width of 100 pixels: `resize,w_100`
  - Automatically rotate the image: `auto-orient,1`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/f.jpg?x-oss-process=image/resize,w_100/auto-orient,1

10.1.3.8. Incircle

You can use the incircle parameter to process an image stored in OSS as an incircle. This topic describes the parameters and examples to save an image in an ellipse.

Parameters

Operation name: circle

The following table lists the parameters.

Parameter	Description	Valid value
r	Specifies the radius of an incircle.	[1,4096]

Usage notes

- If the final format of the image is PNG, WebP, or BMP that supports alpha channels, the areas of the image outside the ellipse become transparent. If the final format of the image is JPG, the areas of the image outside the ellipse become white. We recommend that you save the processed image in PNG.
- If the value of r is greater than half of the shortest side of the source image, the incircle is returned based on the value of the half of the shortest side of the source image. Value of r = The shortest side of the source image/2.

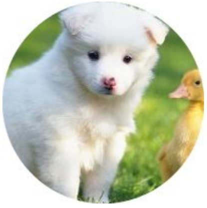
Examples

The image-demo bucket that is located in the China (Hangzhou) region is used as an example. Public endpoint of the image:

https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg

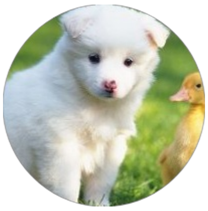
- Crop an image that has a crop radius of 100 pixels. If the image is saved in the JPG format, the areas of the image outside the ellipse become white.

  The URL used to process the image is in the following format: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/circle,r_100



- Crop an image that has a crop radius of 100 pixels. If the image is saved in the PNG format, the areas of the image outside the ellipse become transparent.

  The URL used to process the image is in the following format: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/circle,r_100/format,png



10.1.3.9. Indexed cut

You can use indexed cut parameters to cut a source image stored in OSS based on the specified size and retrieve the required image. This topic describes the parameters and examples to perform indexed cut.

Parameters

Operation name: indexcrop

The following table lists the parameters.

Parameter	Description	Valid value
x	The length of each image partition during horizontal cutting. One of the x and y parameters must be used.	[1, image width].
y	The length of each image partition during vertical cutting. One of the x and y parameters must be used.	[1, image height].
i	The image partition selected after cutting.	[0, maximum number of partitions). By default, the value is 0, which indicates the first partition.

Usage notes

- If the specified index exceeds that of the cut range, the system returns the source image.
- If both x and y are specified and their values are valid, the value of y takes effect.

Examples

The image-demo bucket that is located in the China (Hangzhou) region is used as an example. Public endpoint of the image:

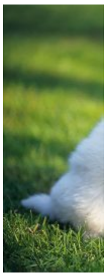
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg

- Cut an image along the horizontal axis

  Requirements and parameters:

  - Cut the image by 100 pixels along the horizontal axis: `indexcrop,x_100`
  - Retrieve the first partition: `i_0`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/indexcrop,x_100,i_0

- Cut an image along the vertical axis

  Requirements and parameters:

  - Cut the image by 100 pixels along the vertical axis: `indexcrop,y_100`
  - Retrieve the 11th partition: `i_10`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/indexcrop,y_100,i_10

  The source image is returned because the maximum number of partitions exceeds that of the cut range.

10.1.3.10. Rounded rectangle

You can round the corners of a rectangle image stored in OSS by adding rounded-corners parameters. This topic describes the parameters used to round the corners of a rectangle image and provides examples on how to round the corners of a rectangle image.

Parameters

Operation name: rounded-corners

The following table describes the parameters you can configure.

Parameter	Description	Valid value
r	The radius at which the corners are rounded.	[1,4096]

Usage notes

- If the final format of an image is PNG, WebP, or BMP that supports alpha channels, the areas of the image outside the rounded rectangle become transparent. If the final format of the image is JPG, the areas of the image outside the rounded rectangle become white. We recommend that you save the processed image in the PNG format.
- If the specified radius for the rounded corners is greater than the radius of the largest incircle of the source image, the radius of the largest incircle of the source image is used as the radius to round the corners. In this case, the radius at which the corners are rounded is equal to half of the smallest edge of the source image.

Examples

An image in the bucket named image-dem in the China (Hangzhou) region is used in this example. The following URL is used to access the image over the Internet:

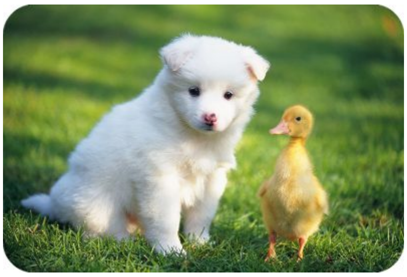
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg

- Round the corners of the source image.

  Take the following steps to configure parameters:

  - Set the radius at which the corners are rounded to 30 pixels to perform the rounded-corners operation: `rounded-corners,r_30` .
  - Save the processed image in the PNG format: `format,jpg` . If the format of the source image is JPG, you can leave this parameter unspecified.

  The following URL is used to process the image: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/rounded-corners,r_30



- Crop the source image before you round the image corners. Save the processed image in the PNG format.

  Take the following steps to configure parameters:

  - Crop the source image to 100 × 100 pixels from the default start position: `crop,w_100,h_100` .
  - Set the radius at which the corners are rounded to 10 pixels to perform the rounded-corners operation: `rounded-corners,r_10` .
  - Save the processed image in the PNG format: `format,png`

  The following URL is used to process the image: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/crop,w_100,h_100/rounded-corners,r_10/format,png

10.1.3.11. Blur

You can use blur parameters to blur a source image stored in Object Storage Service (OSS). This topic describes the parameters and examples to blur an image.

Parameters

Operation name: blur

The following table describes the parameters that you can configure to blur an image.

Parameter	Required	Description	Valid value
r	Yes	Specifies the blur radius.	[1,50] A greater value indicates a blurrier image.
s	Yes	Specifies the standard deviation of a normal distribution.	[1,50] A greater value indicates a blurrier image.

Examples

An image in the bucket named image-demo in the China (Hangzhou) region is used in the following examples. The following URL is used to access the image over the Internet:

https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg

If you set the r parameter to 3 and the s parameter to 2, the URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/blur,r_3,s_2

10.1.3.12. Rotate

You can rotate an image stored in OSS clockwise by adding rotate parameters. This topic describes the parameters used to rotate an image and provides examples on how to rotate an image.

Parameters

Operation name: rotate

The following table describes the parameters you can configure.

Parameter	Description	Valid value
[value]	The degree by which the image is rotated clockwise.	[0,360] Default value: 0. A value of 0 indicates that the image is not rotated.

Usage notes

- If an image is not rotated by 90°, 180°, 270°, or 360°, the size of the processed image increases.
- An image you want to rotate cannot exceeds 4096 × 4096 pixels.

Examples

An image in the bucket named image-dem in the China (Hangzhou) region is used in this example. The following URL is used to access the image over the Internet:

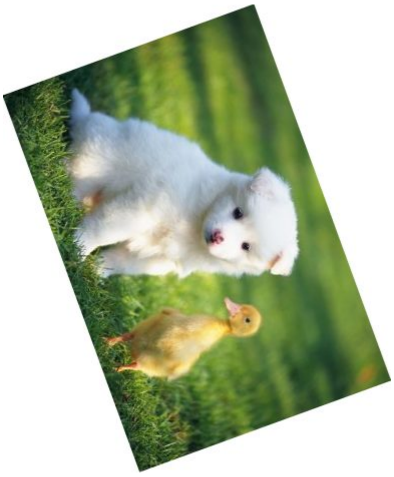
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg

- Rotate the source image by 90 degrees clockwise.

  The following URL is used to process the image: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/rotate,90

- Rotate the source image by 270 degrees clockwise

  The following URL is used to process the image: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/rotate,70



10.1.3.13. Gradual display

You can use gradual display parameters to configure gradual display for source images stored in OSS. This topic describes the parameters and examples to configure gradual display.

When the network environment is poor or the image size is large, the image can be displayed in two ways on the web page:

- Standard display: loads and displays images row by row from top to bottom.
- Gradual display: displays the fuzzy outline of the image, and then loads the image gradually until the complete image is displayed.

The gradual display operation applies only to the source images in the JPG format. If the source image is not in the JPG format, you must add the `format,jpg` parameter to convert the format of the image to JPG.

Parameters

Operation name: interlace

The following table lists the parameters.

Parameter	Description	Valid value
[value]	Specifies whether to set the image to gradual display.	0 and 1 • 1: indicates that the source image is set to gradual display. • 0: indicates that the source image is set to standard display.

Examples

The image-demo bucket that is located in the China (Hangzhou) region is used as an example. The public endpoint of the image is `https://image-demo.oss-cn-hangzhou.aliyuncs.com`. The images used are example.jpg and panda.png in the root folder.

- Resize the image to a width of 200 pixels and set the image to gradual display

  Requirements and parameters:

  - Resize the image to a width of 200 pixels: `resize,w_200`
  - Set the image to gradual display: `interlace,1`

  The URL used to process the image is in the following format: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_200/interlace,1

- Save the PNG image as a JPG image and set the image to gradual display

  Requirements and parameters:

  - Convert the format of the image to JPG: `format,jpg`
  - Set the image to gradual display: `interlace,1`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/panda.png?x-oss-process=image/format,jpg/interlace,1

10.1.3.14. Query the average tone

This topic describes the parameters and examples to query the average tone of an image.

Parameters

Operation name: average-hue

The returned average tone information is in the following format: 0xRRGGBB. RR, GG, and BB use two hexadecimal digits. RR indicates red. GG indicates green. BB indicates blue.

Examples

The image-demo bucket that is located in the China (Hangzhou) region is used as an example. Public endpoint of the image:

https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg

The URL used to query the average tone of the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/average-hue

The average color information returned in the browser is 0x5c783b. 0x5c783b indicates the RGB color value (92,120,59).

10.1.3.15. Brightness

You can adjust the brightness of an image stored in OSS by adding bright parameters. This topic describes the parameters used to adjust the brightness of an image and provides examples on how to adjust the brightness of an image.

Parameters

Operation name: bright

The following table describes the parameters you can configure.

Parameter	Description	Valid value
[value]	The percentage by which to adjust the image brightness.	[-100, 100] • A value smaller than 0 indicates that the brightness of the image is decreased. • A value of 0 indicates that the brightness of the image is not changed. • A value greater than 0 indicates that the brightness of the image is increased.

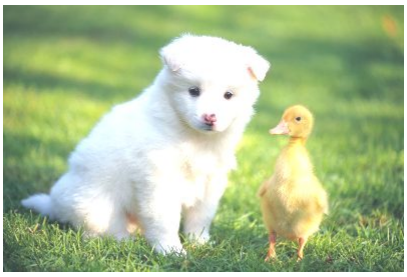
Examples

An image in the bucket named image-dem in the China (Hangzhou) region is used in this example. The following URL is used to access the image over the Internet:

https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg

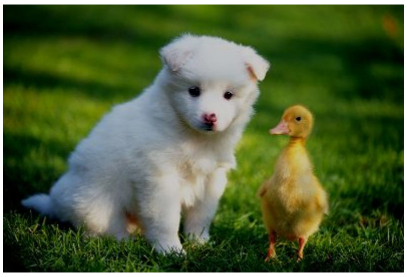
- Increase the brightness of the image by 50 percent

  The following URL is used to process the image: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/bright,50

- Decrease the brightness of the image by 50 percent

  The following URL is used to process the image: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/bright,-50

10.1.3.16. Sharpen

You can sharpen an image stored in OSS by adding sharpen parameters. This topic describes the parameters used to sharpen an image and provides examples on how to sharpen an image.

Parameters

Operation name: sharpen

The following table describes the parameters you can configure.

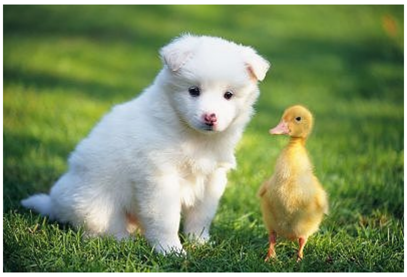
Parameter	Description	Valid value
[value]	The degree of sharpness.	[50,399] A greater value indicates a clearer image. However, an overlarge value may result in image artifacts. We recommend that you set this parameter to 100 for optimal effects.

Examples

An image in the bucket named image-dem in the China (Hangzhou) region is used in this example. The following URL is used to access the image over the Internet:

https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg

Sharpen the source image. The degree of sharpness is set to 100. The following URL is used to process the image: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/sharpen,100



10.1.3.17. Contrast

Contrast refers to the measurement of different brightness levels between the brightest white and the darkest black of an image, that is, the grayscale contrast of an image. You can use contrast parameter to adjust the contrast of the source images stored in OSS. This topic describes the parameters and examples to adjust the contrast for an image.

Parameters

Operation name: contrast

Parameter	Description	Valid value
[value]	The contrast of the image.	[-100,100] • A value smaller than 0: reduces the contrast. • A value of 0: maintains the contrast. • A value greater than 0: increases the contrast.

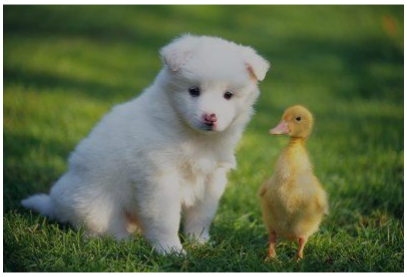
Examples

The image-demo bucket that is located in the China (Hangzhou) region is used as an example. Public endpoint of the image:

https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg

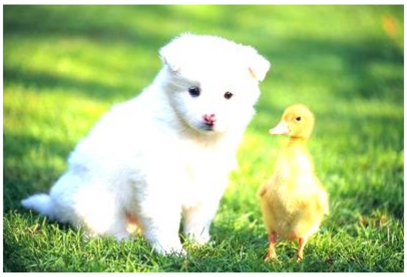
- Reduce the contrast by 50

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/contrast,-50

- Increase the contrast by 50

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/contrast,50

10.1.4. Save processed images

By default, Image Processing (IMG) does not save processed images. You must add the saveas parameter to an IMG request to save a processed image as an object to a specified bucket.

Usage notes

- Required permissions

  To save a processed image, you must have the `oss:PostProcessTask` permission on the source bucket in which the source image is stored, the `oss:PutBucket` permission on the destination bucket in which you want to store the processed image, and the `oss:PutObject` permission on the object as which you want to store the processed image.

- Region

  You can save the processed image to the same bucket in which the source image is stored or to a different bucket. However, the source bucket and the destination bucket must belong to the same Alibaba Cloud account and must be in the same region.

- Storage method

  Images that are processed by using object URLs cannot be directly saved to a specified bucket. You can save the processed images to your computer and then upload them to the specified bucket.

- ACL

  The access control list (ACL) of the processed image is the same as that of the bucket in which the image is saved and cannot be customized.

- Storage duration

  If you want to store the processed image for the specific duration, configure a lifecycle rule for the image object to specify the time when the object expires. For more information, see Lifecycle rules based on the last modified time.

Use OSS SDKs

The following code provides examples on how to save processed images by using OSS (Object Storage Service) SDKs for common programming languages. For more information about how to save processed images by using OSS SDKs for other programming languages, see Overview.

Java

```
// Set yourEndpoint to the endpoint of the region in which the bucket is located. For example, if your bucket is located in the China (Hangzhou) regi
on, set yourEndpoint to https://oss-cn-hangzhou.aliyuncs.com.
String endpoint = "yourEndpoint";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API op
erations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Specify the name of the bucket in which the image you want to process is located.
String bucketName = "examplebucket";
// Specify the full path of the image object. The full path of the object cannot contain bucket names.
String sourceImage = "exampleimage.png";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
try {
    // Resize the image to the height and width of 100 pixels.
    StringBuilder sbStyle = new StringBuilder();
    Formatter styleFormatter = new Formatter(sbStyle);
    String styleType = "image/resize,m_fixed,w_100,h_100";
    // Name the processed image example-resize.png and save the image to the current bucket.
    // Specify the full path of the image object. The full path of the object cannot contain bucket names.
    String targetImage = "example-resize.png";
    styleFormatter.format("%s|sys/saveas,o_%s,b_%s", styleType,
            BinaryUtil.toBase64String(targetImage.getBytes()),
            BinaryUtil.toBase64String(bucketName.getBytes()));
    System.out.println(sbStyle.toString());
    ProcessObjectRequest request = new ProcessObjectRequest(bucketName, sourceImage, sbStyle.toString());
    GenericResult processResult = ossClient.processObject(request);
    String json = IOUtils.readStreamAsString(processResult.getResponse().getContent(), "UTF-8");
    processResult.getResponse().getContent().close();
    System.out.println(json);
} catch (Exception e) {
    e.printStackTrace();
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

Use the RESTful API

If your program requires more custom options to save processed images, you can call RESTful API operations. In this case, you must manually write code to calculate the signature.

To process an image, you can call PostObject and pass x-oss-process in the message body of the PostObject request. Then, add the saveas parameter in the request to save the processed image to a specified bucket. For more information, see PostObject.

You must specify the parameters described in the following table when you add saveas to the request.

Parameter	Description
o	The name of the object as which the processed image is stored. This parameter must be URL-safe Base64-encoded. For more information, see Encode watermarks.

Parameter	Description
b	The name of the bucket in which the processed image is stored. This parameter must be URL-safe Base64-encoded. If this parameter is not specified, the processed image is saved to the current bucket.

You can use the following methods to process an image and save the processed image to a specified bucket:

- The following code provides an example on how to configure IMG parameters to process an image and save the processed image to a specified bucket:

```
POST /ObjectName?x-oss-process HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Content-Length: 247
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:KU5h8YMUC78M30dXqf3JxrT****=
// Proportionally resize the source image named test.jpg to a width of 100 pixels and save the processed image to a bucket named test.
x-oss-process=image/resize,w_100|sys/saveas,o_dGVzdC5qcGc,b_dGVzdA
```

- The following code provides an example on how to use an image style to process an image and save the processed image to a specified bucket:

```
POST /ObjectName?x-oss-process HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Content-Length: 247
Date: Fri, 04 May 2012 03:22:13 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:KU5h8YMUC78M30dXqf3JxrT****=
// Use an image style named examplestyle to process the source image named test.jpg and save the processed image to a bucket named test.
x-oss-process=style/examplestyle|sys/saveas,o_dGVzdC5qcGc,b_dGVzdA
```

10.1.5. Common errors

If an error occurs when you access Image Processing (IMG), IMG returns an error code and an error message. This helps you locate and fix the error.

Error response format

The following example shows an error response:

```
<Error>
  <Code>BadRequest</Code>
  <Message>Input is not base64 decoding.</Message>
  <RequestId>52B155D2D8BD99A15D0005FF</RequestId>
  <HostId>userdomain</HostId>
</Error>
```

The error response contains the following elements:

- `Code` : the error code that IMG returns to the user.
- `Message` : the detailed error information provided by IMG.
- `RequestId` : the UUID used to identify an error request. If the problem persists, you can send this UUID to technical support to help locate the cause of the error.
- `HostId` : the ID used to identify the accessed IMG cluster.

Error codes

The following table describes the error codes contained in responses to IMG requests.

Error code	Description	Solution
InvalidArgument	The error message returned because the parameter is invalid.	HTTP 400 status code
BadRequest	The error message returned because a request error occurs.	
MissingArgument	The error message returned because a required parameter is not specified.	
ImageTooLarge	The error message returned because the image size exceeds the limit.	
WatermarkError	The error message returned because a watermark error occurs.	
NotImplemented	The error message returned because the access is denied.	
AccessDenied	The error message returned because access is denied.	HTTP 403 status code
SignatureDoesNotMatch	The error message returned because the signature calculated by OSS does not match the signature provided in the request.	
NoSuchKey	The error message returned because the specified image does not exist.	HTTP 404 status code
NoSuchStyle	The error message returned because the specified style does not exist.	

Error code	Description	Solution
InternalError	The error message returned because an internal error occurs.	HTTP 500 status code

SDK demos

- OSS SDK for Java
- OSS SDK for Python
- OSS SDK for PHP
- OSS SDK for Go
- OSS SDK for C++
- OSS SDK for C
- OSS SDK for .NET
- OSS SDK for Node.js
- OSS SDK for Browser.js
- OSS SDK for Android

10.1.6. FAQ

This topic provides answers to frequently asked questions about OSS Image Processing (IMG).

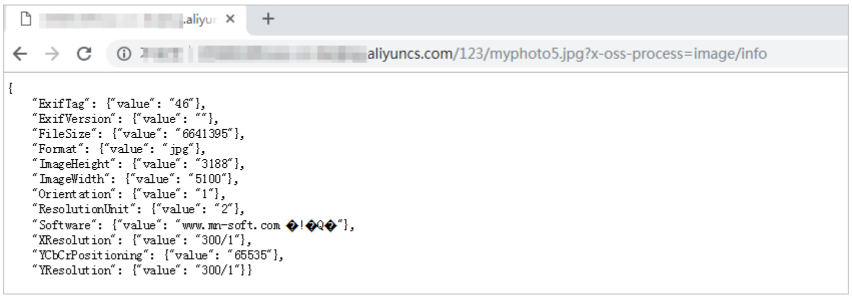
If you encounter problems such as invalid parameter values, you can use the `? x-oss-process=image/info` parameter to check whether parameter values of the source image exceed thresholds. Each side of the image cannot exceed 4,096 pixels. The product of dimensions cannot exceed 4,096 pixels × 4,096 pixels.

What do I do if "Picture exceed the maximum allowable rotation range" is reported when I rotate an image?

Cause: The length of a single side of the source image exceeds 4,096 pixels, or the product of dimensions of four sides exceed 4,096 pixels × 4,096 pixels.

Troubleshooting:

1. Use the `? x-oss-process=image/info` parameter to obtain information about the image and determine whether the length of a single side or product of dimensions exceed limits.



[redacted].aliyu

[redacted]aliyuncs.com/123/myphoto5.jpg?x-oss-process=image/info

```
{
    "ExifTag": {"value": "46"},
    "ExifVersion": {"value": ""},
    "FileSize": {"value": "6641395"},
    "Format": {"value": "jpg"},
    "ImageHeight": {"value": "3188"},
    "ImageWidth": {"value": "5100"},
    "Orientation": {"value": "1"},
    "ResolutionUnit": {"value": "2"},
    "Software": {"value": "www.mn-soft.com �!�Q�"},
    "XResolution": {"value": "300/1"},
    "YCbCrPositioning": {"value": "65535"},
    "YResolution": {"value": "300/1"}}
```

2. The `ImageWidth` value is 5100, which exceeds 4096.

Solution: Use the `auto-orient,0` parameter to disable automatic rotation, use the `resize` parameter to adjust the image size, and rotate the image. Example: *http://test.oss-cn-beijing.aliyuncs.com/123/myphoto5.jpg?x-oss-process=image/auto-orient,0/resize,m_lfit,h_2000,w_2000,limit_1/rotate,90.*

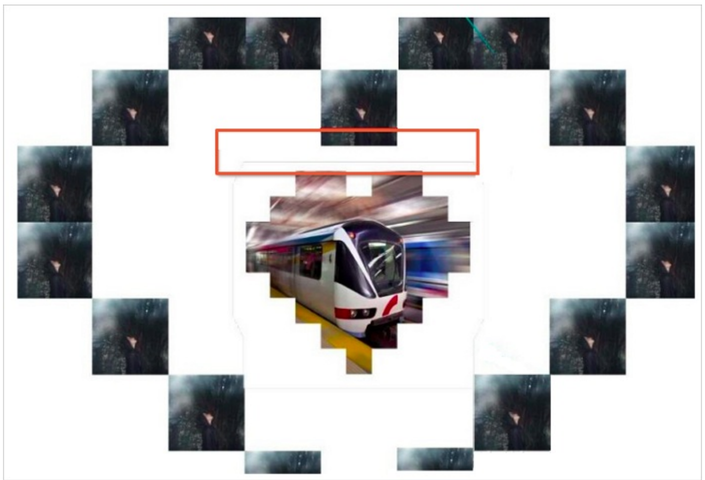
What do I do if the average tone queried by OSS does not match that of the image?

Cause: The average tone is not obtained by calculating the ratios of colors, but by the prominent colors of the image. Logic of calculating the average tone:

1. Calculate the average colors of the image.
2. Traverse each pixel. Calculate the difference between the pixel color and each average color. If the difference is greater than a threshold, the pixel is added to the list of prominent pixels.
3. Calculate the average value of the prominent pixels. The average value is the average tone.

Solution: Use the `? x-oss-process=image/average-hue` parameter to query average tone-related parameters for the OSS image.

What do I do if black outlines appear when I add a watermark to an image?



Cause: The black outlines are not caused by IMG. When you add a watermark to a source image, the source image is overlaid with the watermark. If the watermark and source image differ in RGB values, black outlines may appear after the source image is overlaid. This phenomenon is common when you use image processing tools.

Use the `? x-oss-process=image/average-hue` parameter to query the RGB parameters of the two images. You can add this parameter to the end of the source image URL to determine whether the RGB parameter values of the two images are the same.

Solution: The RGB parameter value of the source image is 0x0e0e0e. The RGB parameter value of the watermark is 0xffffff. When you add the watermark to the image, the black outline appears. You can use the `t` parameter to adjust the transparency and make the outline invisible. Valid values of t: 1 to 100. Example: *http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300/watermark,image_cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA,t_90,g_se,x_10,y_10,t_50*.

What do I do if my image style fails to be accessed over CDN back-to-origin?

Use OSS endpoints to test whether the image style can be accessed. Use the following URLs to analyze the cause:

http://test.oss-cn-beijing.aliyuncs.com/MomClass/ChuXin/3_2_336_462.jpg@30-30bl

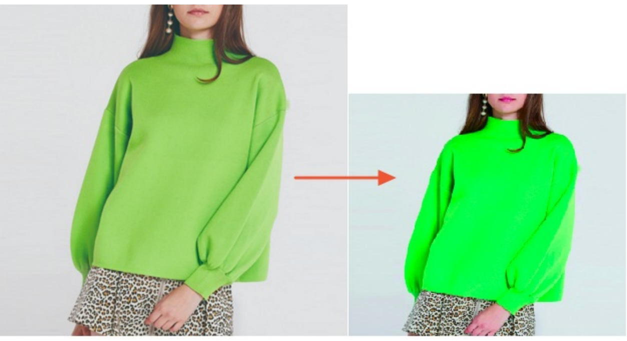
http://test.img-cn-beijing.aliyuncs.com/MomClass/ChuXin/3_2_336_462.jpg@30-30bl

- *img-cn-region.aliyuncs.com* indicates the OSS endpoint of the earlier version. OSS endpoints of the earlier version and the current version differ in the delimiter and syntax used for IMG.
- *oss-cn-region.aliyuncs.com* indicates the endpoint used after 2017. This endpoint does not support the syntax of IMG or the at sign (@) as those used in the earlier endpoint. To access an image whose endpoint contains img, synchronize the endpoint with the latest OSS endpoint.

The preceding endpoint of the earlier version uses Gaussian Blur-related parameters. After the earlier endpoint is synchronized with the latest OSS endpoint, you can access the image through the latest OSS endpoint.

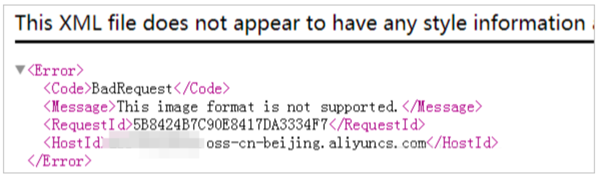
http://test.oss-cn-beijing.aliyuncs.com/MomClass/ChuXin/3_2_336_462.jpg?x-oss-process=image/blur,r_3,s_30

What do I do if my image color is brightened after the image is resized?



Solution: Use tools such as Photoshop to query the color model of the source image. If the source image uses the RGB color model, the colors remain unchanged after the image is resized. If the source image uses the CMYK color model, the colors slightly change after the image is resized. The support for CMYK is in progress. Colors change after the image is resized.

What do I do if my image can be opened from my local computer, but the system shows that the image has been corrupted when I process the image?



This XML file does not appear to have any style information

```
▼<Error>
   <Code>BadRequest</Code>
   <Message>This image format is not supported.</Message>
   <RequestId>5B8424B7C90E8417DA3334F7</RequestId>
   <HostId>[redacted]oss-cn-beijing.aliyuncs.com</HostId>
 </Error>
```

Problem description: My image can be opened from my local computer, but the system shows that the image has been corrupted when I upload the image to OSS and process the image.

Troubleshooting:

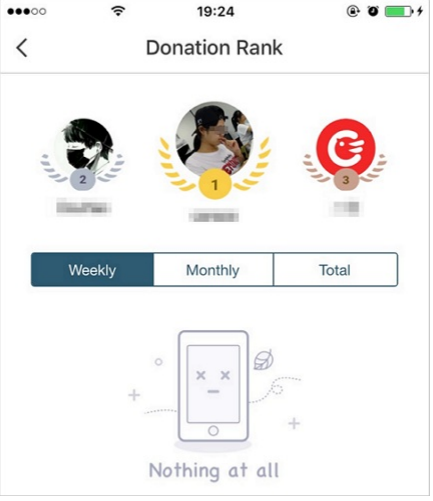
1. Obtain the URL of the source image. Use `? x-oss-process=image/info` to view information of the source image. If the source image is unavailable, and an error is reported, the source image is corrupted.
2. To troubleshoot this problem, use the ImageMagick tool to adjust the image. If an error is reported, the image is corrupted. An example of resizing the image:

   ```
   convert -resize 1024x768 1123331261_15353307414801n.jpg
   ```

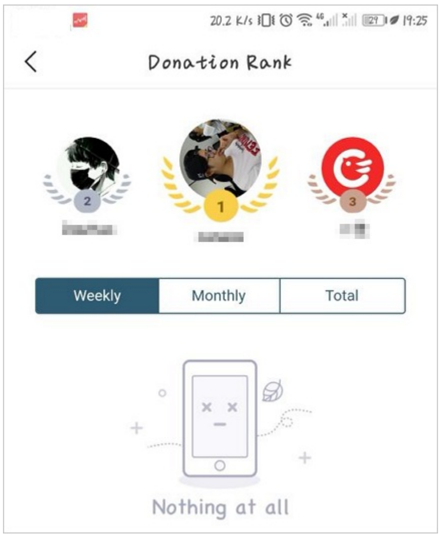
The corrupted image can be opened from the local computer because the local image viewer has repaired the image. However, OSS does not repair the corrupted image. Therefore, the image cannot be displayed by the browser.

What do I do if an image stored in OSS is rotated 90 degrees?

Problem description: This image can be accessed through an OSS endpoint.



When the image is accessed over CDN, the image is rotated 90 degrees.



Cause: OSS works properly because the image can be accessed through an OSS endpoint. The possible cause lies in how the browser processes the image because the image is rotated when the image is accessed over CDN. Use the `? x-oss-process=image/info` parameter to query information about the source image. The `rotation 90` parameter is included.

Solution: Delete the rotation-related parameters.

What do I do if IMG does not take effect after CDN is used to access an image?

Troubleshooting: Check whether the **parameter filtering** feature is enabled for CDN. If this feature is enabled, parameters that follow the question mark (?) are removed from the URL when the image is accessed over CDN. For more information, see Configure parameter filtering.

What do I do if "Picture exceed the maximum allowable rotation range" is reported when I use an image?

Troubleshooting:

- Use the ImageMagick tool to check whether the automatic rotation feature is enabled for the source image.
- Use the `auto-orient,0` parameter to process the image. If the image can be processed, the source image has automatic rotation enabled.

If the source image has automatic rotation enabled, the image width and height cannot exceed 4,096 pixels.

What do I do if a blank image appears when I access an image configured with CDN from an iOS device through an URL that includes IMG parameters, but the image is accessible after I refresh the web page or when I access the image from a computer?

Cause: OSS works properly because the image can be accessed from the computer and mobile phone. Otherwise, the image cannot be accessed from a computer.

Troubleshooting:

1. Check whether access from a mobile phone to the image in OSS is normal.
   - If the image in OSS is accessible, whereas the access over CDN is abnormal, CDN node errors may cause loading failures or incorrect content is cached on CDN nodes.
   - If access to OSS and over CDN are both abnormal, but access over CDN becomes normal after the website is refreshed. One possible cause is that incorrect content is cached on CDN nodes.
2. Check whether access from a computer to the image in OSS is normal.
   - If the access is normal, the cause lies in the mobile phone.
   - If the access is abnormal, the cause lies in the image.

In this case, the image format is WebP. iOS does not support this format.

What do I do if I cannot open the source image and processed image that are stored in OSS?

Troubleshooting:

1. Download one of the images.

```
[root@edas02 aliyun-oss-php-sdk]# wget https://zh.mobi/test/123.jpg
--2018-11-2210:55:16--  https://zh.mobi/test/123.jpg
The IP address of the zh.mobi host is being resolved.
The HTTP request is sent. Waiting for the response. 200 OK
Image size: 4141232 (3.9M) [image/jpeg].
The image is saved as 123.jpg.
100%[=======================================================================================>] 4,141,232 12.5MB/s Time: 0.3s.
2018-11-2210:55:16 (12.5 MB/s) - 123.jpg [4141232/4141232]) saved.
```

2. Use ImageMagick to check the encoding methods used for the image.

```
[root@edas02 aliyun-oss-php-sdk]# identify 123.jpg
identify: Not a JPEG file: starts with 0x000x00 `123.jpg' @ error/jpeg.c/JPEGErrorHandler/316.
[root@edas02 aliyun-oss-php-sdk]#
```

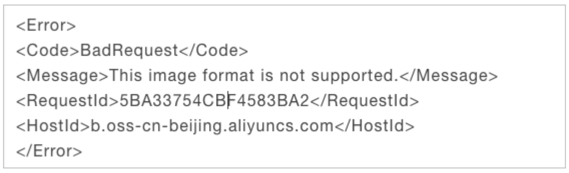
   If an encoding method error occurs, the cause does not lie in OSS. You can use this tool to analyze similar errors.

What do I do if a split line appears after I process my image?

Cause: No split lines exist, but a problem caused by pixel changes after the image is processed. The source image is an RGB (true color) image whose height is 2,560 pixels and width is 1,440 pixels. After the image is processed, the image height is cropped to 1,092 pixels, and the image width is cropped to 1,080 pixels. The image size in pixels is reduced, which causes abnormal display of the image.

Solution: Use the `quality,Q_100` parameter to increase the absolute quality value of the image to 100.

What do I do if "BadRequest" is reported when I use IMG?



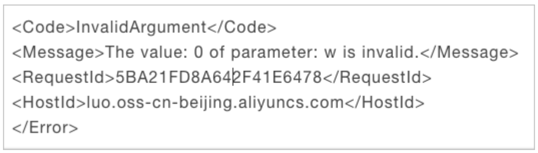
```
<Error>
<Code>BadRequest</Code>
<Message>This image format is not supported.</Message>
<RequestId>5BA33754CBF4583BA2</RequestId>
<HostId>b.oss-cn-beijing.aliyuncs.com</HostId>
</Error>
```

Troubleshooting:

1. Use the convert command supported by ImageMagick to view the format of the source image.
2. Check whether the format of the source image is supported by IMG. For more information about image formats supported by OSS, see Usage notes.

Solution: Convert the format of the image to a format supported by IMG.

What do I do if "InvalidArgument" is reported when I use IMG?



```
<Code>InvalidArgument</Code>
<Message>The value: 0 of parameter: w is invalid.</Message>
<RequestId>5BA21FD8A642F41E6478</RequestId>
<HostId>luo.oss-cn-beijing.aliyuncs.com</HostId>
</Error>
```

Cause: Check the request parameters of the source image. Parameters similar to `20180899269957.jpg@0w_2e_11_1an.src` are supported by the endpoints in the `img-cn-xx` format. After img-cn-xx is changed to `oss-cn-xxx` , endpoints in the oss-cn-xxx format does not support access that uses the request method of img-cn-xx-based endpoints. img-cn-xx-based endpoints do not support access over HTTPS.

Can OSS identify custom query parameters used for dynamically resizing an image in a URL?

No, this function is unavailable in OSS.

How do I display a text watermark in two lines and can I add multiple text watermarks to an image?

A text watermark cannot be displayed in multiple lines in OSS. However, you can add multiple text watermarks to an image. For more information, see Add watermarks.

10.1.7. FAQ on using old and new versions of APIs and domain names

There are major differences between new and old versions of APIs:

- New version API: `http://bucket.<endpoint>/object?x-oss-process=image/action,parame_value`

  All image manipulation operations are passed by `x-oss-process` . Each action is executed sequentially without any need for channel management.

- Old versionAPI: `http://channel.<endpoint>/object@action.format`

  It can be processed as a separator by `@` .

What are the advantages of OSS domain names when used with the Image Service?

Item	Use IMG domain access	Direct use of OSS domain name access
Use	Store and process two Domain Name Systems	One-stop processing for upload, management, process, distribution.
Is new version of API supported?	Supported	Supported
Is old version of API supported?	Supported	Not supported by default
Is https supported?	Not supported.	Supported
Is VPC Network supported?	Not supported	Supported
Is multi-domain binding supported?	Not supported	Supported
Is source station update automatically refresh Alibaba CDN supported?	Not supported	Supported

Note

- When OSS domain names are being used, only APIs for the new version of the ING service can be used. When IMG domain names are being used, APIs for the old and new versions of the IMG service can be used.
- If the IMG domain name is expected to be capable of multi-CDN acceleration, the IMG domain name can be directly accessed by configuring the CDN to go back to the source host, and domain name binding is not required to complete the CDN acceleration.

What is the logic here for the two API methods and the two domain name access methods on the console?

Bucket processed before enabling the old version of image

- To keep the logic consistent with the original, the user sees the Domain Name of the old version of IMG, and custom domain names that have previously been bound.
- The user's original graph protection configuration on the IMG domain name has no effect on the OSS domain name. When you start the same step in cross-region replication, the original graph protection and style separator are synchronized to the OSS domain name.
- When the user closes the image processing service for the current bucket, the style configuration and domain name binding are cleared, and automatically jump to the new page.

Newly created bucket or a bucket that has not previously opened the IMG service:

- The default is to be able to use the image processing service, which does not need to be up or turned off.
- No need to bind domain names, the domain name binding operation is directly consistent with the domain name management of the bucket itself.

If I'm currently using APIs for the old version of the IMG service, how do I switch to OSS domain names?

Currently, APIs for the old version of the IMG service cannot be used with OSS domain names without a request being sent to Alibaba Cloud. To request use of APIs for the old version, submit a ticket to Alibaba Cloud asking for this service. For style-based access, both OSS and IMG domain names can be used. If all your images are accessed by style, follow these steps to switch to the use of OSS domain names:

1. Enable configuration synchronization in the current Image Service configurations, so that style separators and the source image protection feature can be synchronized to OSS domain names.
2. If you use a custom domain name, direct its CNAME to the OSS domain name.

Are style configurations the same for IMG and OSS domain names?

All style configurations are shared by IMG and OSS domain names. Style configurations for IMG domain names can be applied to OSS domains.

10.2. Video snapshots

This topic describes the parameters that you can configure to capture video snapshots and provides examples.

Usage notes

- When you capture video snapshots, you are charged for the number of captured images. For more information about billing, see Data processing fees.
- Object Storage Service (OSS) can capture images from video objects only in the H.264 and H.265 formats.
- By default, OSS does not automatically store captured images. You must manually download the captured images to your local storage devices.

Parameters

Operation type: `video`

Operation name: snapshot

Parameter	Description	Value value
t	The time when the image is to be captured.	[0, video duration] Unit: milliseconds
w	The width based on which to capture the image. If this parameter is set to 0, the width based on which to capture the image is automatically calculated.	[0, video width] Unit: pixels
h	The height based on which to capture the image. If this parameter is set to 0, the height based on which to capture the image is automatically calculated. If w and h are set to 0, the width and height of the source image are used.	[0, video height] Unit: pixels
m	The mode used to capture the image. If this parameter is not specified, the image is captured in the default mode. In other words, the image at the specified point in time in the video is captured. If this parameter is set to fast, the most recent key image before the specified time is captured.	*fast*
f	The format of the captured image.	*jpg* and *png*
ar	Specifies whether to automatically rotate the image based on the video information.	*auto*, *h*, and *w* - *auto* indicates that the captured image is automatically rotated based on the video information. - *h* indicates that the captured image is automatically rotated based on the mode in which the height is greater than the width. - *w* indicates that the captured image is automatically rotated based on the mode in which the width is greater than the height.

Examples

- Use the fast mode to capture the image at the seventh second of the video. Export the captured image as a JPG image whose width is 800 pixels and height is 600 pixels.

  The URL of the processed image is in the following format: `<Video URL>?x-oss-process=video/snapshot,t_7000,f_jpg,w_800,h_600,m_fast` .

- Capture the image at the fiftieth second of the video accurately. Export the captured image as a JPG image whose width is 800 pixels and height is 600 pixels.

  The URL of the processed image is in the following format: `<Video URL>? x-oss-process=video/snapshot,t_50000,f_jpg,w_800,h_600` .

Generate a signed URL to capture a video snapshot

You can use OSS SDKs to generate a signed URL to capture a video snapshot. The following code provides an example on how to use OSS SDK for Java to generate a signed URL to capture a video snapshot:

```
// In this example, the endpoint of the China (Hangzhou) region is used. Specify the endpoint based on your business requirements.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API operations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Specify the name of the bucket. Example: examplebucket.
String bucketName = "examplebucket";
// Specify the full path of the video object. If the video object is not stored in the root directory of the bucket, you must include the directory of the object in the path. Example: examplefolder/videotest.mp4.
String objectName = "examplefolder/videotest.mp4";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
Capture the image at the fiftieth second of the video accurately. Export the captured image as a JPG image whose width is 800 pixels and height is 600 pixels.
String style = "video/snapshot,t_50000,f_jpg,w_800,h_600";
// Set the validity period of the URL to 10 minutes.
Date expiration = new Date(new Date().getTime() + 1000 * 60 * 10 );
GeneratePresignedUrlRequest req = new GeneratePresignedUrlRequest(bucketName, objectName, HttpMethod.GET);
req.setExpiration(expiration);
req.setProcess(style);
URL signedUrl = ossClient.generatePresignedUrl(req);
System.out.println(signedUrl);
// Shut down the OSSClient instance.
ossClient.shutdown();
```

The method used to generate signed URLs to capture snapshots is similar to that used to generated signed URLs to process images by using Image Processing (IMG). To use the following OSS SDKs to generate a signed URL to capture a snapshot, replace the IMG operations in the code with the snapshot operation.

- Python SDK
- PHP SDK
- Go SDK
- C SDK
- C++ SDK
- .NET SDK
- Android SDK
- iOS SDK
- Node.js SDK
- browser.js SDK

11.Monitoring service

11.1. Overview

The monitoring service of OSS provides metrics to measure the running status and performance of the system. The monitoring service also provides a custom alert service to help you track requests, analyze usage, collect statistics on business trends, and discover and diagnose system problems in a timely manner.

Metrics of the OSS monitoring service are classified into basic service metrics, performance metrics, and billable usage metrics. For more information, see Metrics.

High real-time performance

Real-time monitoring can reveal potential traffic fluctuations during various periods. This facilitates the analysis and evaluation of business scenarios. Real-time monitoring metrics (except for billable usage metrics) allow for minute-level metric data to be collected and aggregated with an output delay of less than a minute. User information that is collected every minute is aggregated into a single value and displayed within one minute to represent the overall monitoring for the minute.

Billable usage metrics

In line with the billing policies, billable usage metrics are collected and presented based on the following criteria:

- Billable usage metric data is collected each hour. Resource metering data for each hour is aggregated into a single value that represents the overall monitoring for that hour.
- Billable usage metric data is displayed with a delay of nearly 30 minutes.
- The statistics collection time is the start time of the relevant statistical period.
- The statistics collection deadline is the end time of the last statistical period of the current month. If no metric data is produced in the current month, the statistics collection deadline is 00:00:00 on the first day of the current month.
- A maximum amount of metric data is pushed for presentation. For precise metric data, go to the Billing Management console and click Usage Records.

For example, you only use PutObject requests to upload data and perform the operation an average of 10 times per minute. Then in the hour from 08:00:00 to 09:00:00 on May 10, 2018, the metric value of the number of PUT requests is 600 (10 × 60 minutes), the statistics collection time is 08:00:00 on May 10, 2018, and the data is displayed at around 09:30:00 on May 10, 2018. If the data is the last metric data from 00:00 on May 01, 2018, the statistics collection deadline for the current month is 09:00:00 on May 10, 2018. If you have not produced any metric data in May 2018, the statistics collection deadline will be 00:00:00 on May 1, 2018.

OSS alert service

You can use an account to set up to 1,000 alert rules. You can configure alert rules for other monitoring metrics in addition to billable usage metrics and statistical metrics. You can also configure multiple alert rules for a single metric.

- For more information about the alert service, see Alert service overview
- For more information about how to use the OSS alert service, see Alert service.
- For more information about the metrics of the OSS monitoring service, see Metrics.

Metric data retention policy

Metric data is retained for 31 days before being automatically cleared upon expiration. To store or perform offline analysis on metric data that is over 31 days old, you can use tools or write code to read the CloudMonitor data storage. For more information, see Access metric data by using the CloudMonitor API.

The OSS console displays metric data from the last seven days. To query metric data that is older than seven days, we recommend that you use CloudMonitor SDKs. For more information, see Access metric data by using the CloudMonitor OpenAPI.

Access metric data by using the CloudMonitor OpenAPI

The CloudMonitor API allows you to access OSS metric data. For more information, see the following topics:

- CloudMonitor SDK reference
- Metric item reference

Monitoring, diagnosis, and troubleshooting

The Service monitoring, diagnosis, and troubleshooting topic describes how to monitor the running status of OSS and perform diagnosis and troubleshooting. This topic provides an overview of the following aspects:

- OSS monitoring

  Describes how to use the OSS monitoring service to monitor the running status and performance of OSS.
- Tracking and diagnosis

  Describes how to use the OSS monitoring service and logging function to diagnose problems as well as how to associate the relevant information in log files for tracking and diagnosis.
- Troubleshooting

  Describes typical problems and methods to troubleshoot the problems.

Precautions

Bucket names must be globally unique in OSS. If you delete a bucket and then create a bucket with the same name as the deleted bucket, the monitoring and alert rules set for the deleted bucket will be applied to the new bucket.

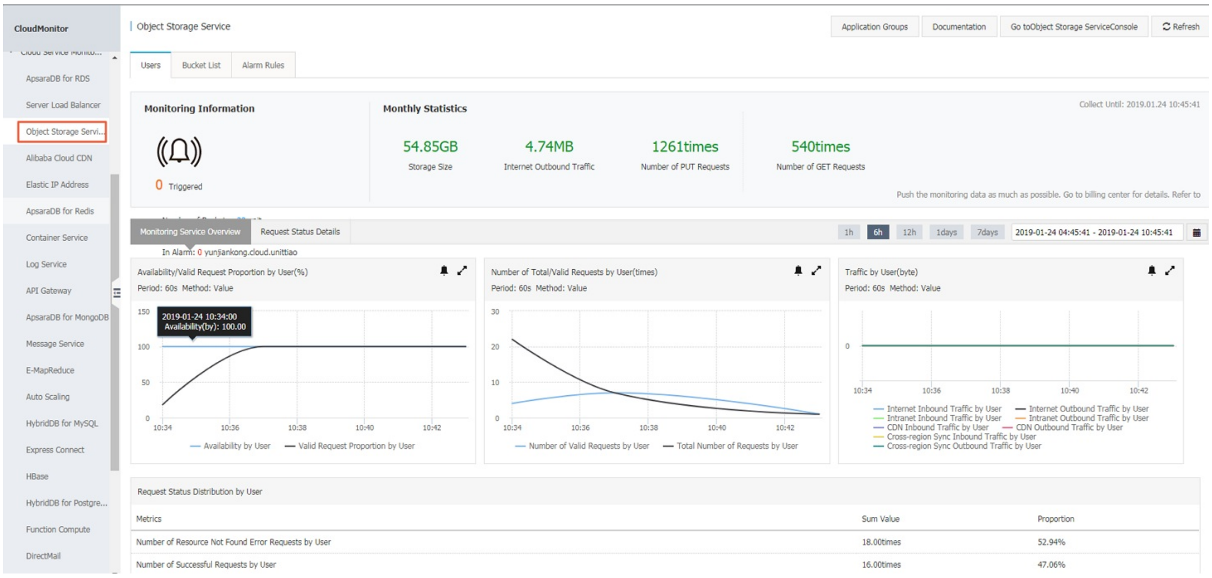
11.2. Monitoring service console user guide

This topic describes how to use OSS monitoring service.

Access monitoring service

OSS monitoring service is available in the CloudMonitor console. You can use either of the following methods to access monitoring service:

- Log on to the OSS console. In the right section of the Overview page, find the Alert Rules section. Click **Manage** to access monitoring service.
- Log on to the CloudMonitor console to view monitoring service.



0 Triggered

Collect Until: 2019.01.24 10:45:41

54.85GB Storage Size

4.74MB Internet Outbound Traffic

1261times Number of PUT Requests

540times Number of GET Requests

Push the monitoring data as much as possible. Go to billing center for details. Refer to

2019-01-24 04:45:41 - 2019-01-24 10:45:41

In Alarm: 0 yunjiankong.cloud.unittiao

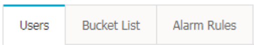
Request Status Distribution by User

Metrics	Sum Value	Proportion
Number of Resource Not Found Error Requests by User	18.00times	52.94%
Number of Successful Requests by User	16.00times	47.06%

OSS monitoring service page

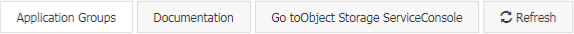
The OSS monitoring service page consists of the following tabs:

- Users
- Bucket List
- Alarm Rules



The page does not provide the auto-refresh function. You can click the **Refresh** button in the upper-right corner to automatically update monitoring data.

In the upper-right corner, you can click **Object Storage ServiceConsole** to log on to the OSS console.

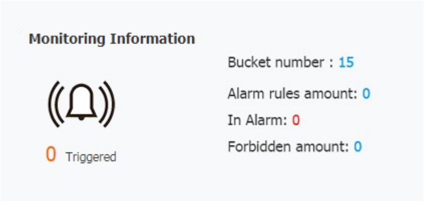


Users

The Users tab displays monitoring data by user. The Users tab consists of the Monitoring Information, Monthly Statistics, and user-level metrics sections.

- Monitoring Information

  This section displays all buckets in your account and the alert rules of your OSS instance.

  0 Triggered

  Bucket number : 15

  Alarm rules amount: 0

  In Alarm: 0

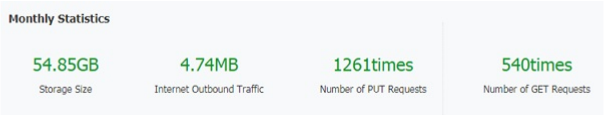
  Forbidden amount: 0

  - Click the number next to **Number of Buckets**. The Bucket List tab appears.
    - Click the number next to **Number of Alarm Rules**. The Alarm Rules tab appears.
    - Click the number next to **In Alarm**. The Alarm Rules tab appears. The Alarm Rules tab displays enabled alert rules.
    - Click the number next to **Number of Rules Disabled**. The Alarm Rules tab appears. The Alarm Rules tab displays disabled alert rules.
    - Click the number under the alert bell icon. The Alarm Rules tab appears. The Alarm Rules tab displays enabled alert rules.

- Monthly Statistics

  This section displays information about metered OSS resources that you have consumed since 00:00:00 on the first day of the current month. The following metrics are displayed:

  - Storage Size
  - Internet Outbound Traffic
  - Number of PUT Requests
  - Number of GET Requests

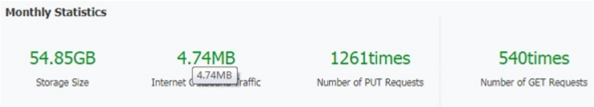
  54.85GB Storage Size

  4.74MB Internet Outbound Traffic

  1261times Number of PUT Requests

  540times Number of GET Requests

  The unit of each metric is automatically adjusted based on their quantities. The exact value is displayed when you move the pointer over a value.

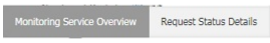


Monthly Statistics

54.85GB	4.74MB	1261times	540times
Storage Size	Internet Outbound Traffic	Number of PUT Requests	Number of GET Requests

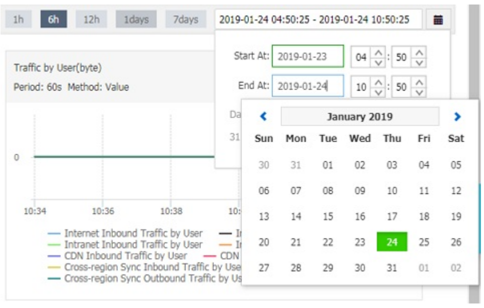
- User-level metrics

  This section displays charts and tables for user-level metrics. This section consists of Monitoring Service Overview and Request Status Details.

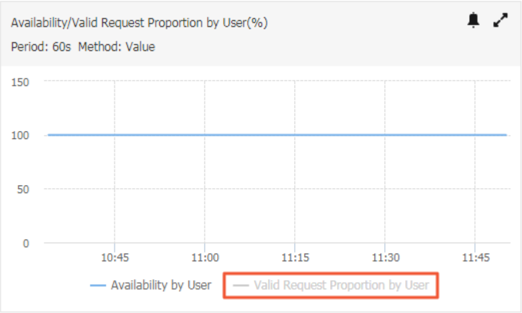
  You can select a predetermined time range or click the date picker to select a custom time range.

  - The following predetermined time ranges are available: 1 h, 6 h, 12 h, 1days, and 7days. The default time range is 1 h.
  - You can use the date picker to select custom time ranges that are accurate to the minute. You can only view statistics that have been generated in the last seven days.

  
  1h
  6h
  12h
  1days
  7days
  2019-01-24 04:50:25 - 2019-01-24 10:50:25
  Traffic by User(byte)
  Period: 60s Method: Value
  Start At: 2019-01-23 04 : 50
  End At: 2019-01-24 10 : 50
  January 2019
  Sun Mon Tue Wed Thu Fri Sat
  30 31 01 02 03 04 05
  06 07 08 09 10 11 12
  13 14 15 16 17 18 19
  20 21 22 23 24 25 26
  27 28 29 30 31 01 02
  0
  10:34
  10:36
  10:38
  Internet Inbound Traffic by User
  Intranet Inbound Traffic by User
  CDN Inbound Traffic by User
  Cross-region Sync Inbound Traffic by User
  Cross-region Sync Outbound Traffic by User
  CDN

  The charts also support the following display modes:

  - Click a legend to hide or display the curve the legend corresponds to, as shown in the following figure.

    
    Availability/Valid Request Proportion by User(%)
    Period: 60s Method: Value
    150
    100
    50
    0
    10:45
    11:00
    11:15
    11:30
    11:45
    Availability by User
    Valid Request Proportion by User

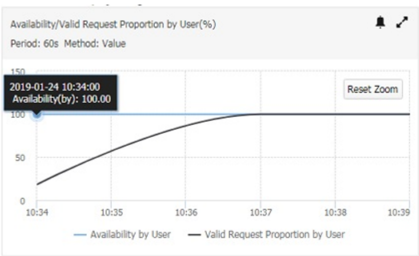
  - In the upper right corner of a chart, you can click the

    ⤢

    icon to enlarge the chart. Note that tables cannot be enlarged.
  - In the upper right corner of a chart, you can click the

    🔔

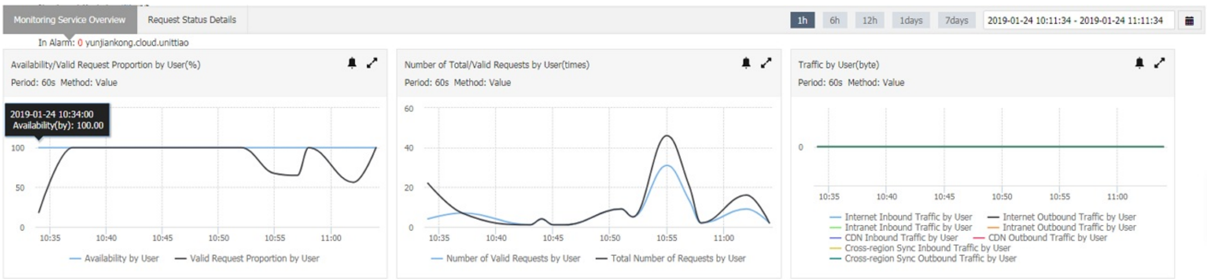
    icon to configure related alert rules. For more information, see Alert service. You cannot configure alert rules for metered metrics or tables.
  - Move the pointer over a curve. Hold down and drag the pointer to increase the time range. You can also click **Reset Zoom** to display the statistics generated within the original time range.

- Monitoring Service Overview

  This section displays the following table and charts for metrics:

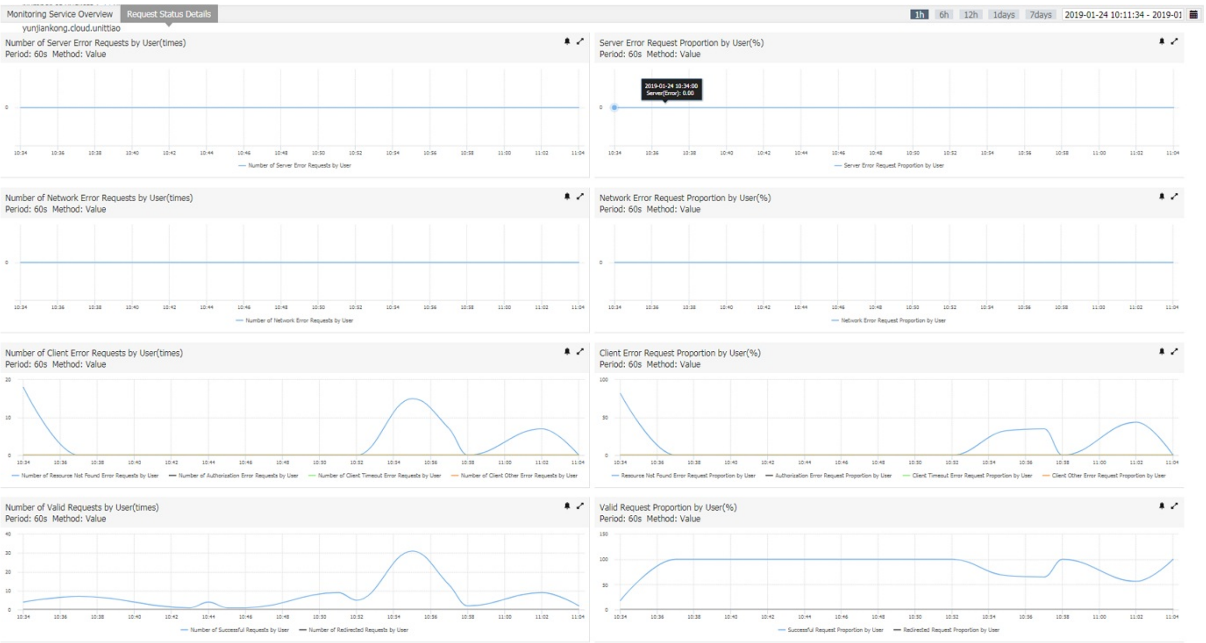
  - Availability/Valid Request Proportion by User(%), which includes the following metrics: the availability and percentage of valid requests.
  - Number of Total/Valid Requests by User(times), which includes the following metrics: the total number of requests and number of valid requests.
  - Traffic by User(byte), which includes the following metrics: the public outbound traffic, public inbound traffic, internal outbound traffic, internal inbound traffic, CDN outbound traffic, CDN inbound traffic, cross-region replication outbound traffic, and cross-region replication inbound traffic.
  - Request Status Distribution by User, which includes the following metrics: the number and percentage of each type of requests within the specified time range.


Monitoring Service Overview
Request Status Details
1h
6h
12h
1days
7days
2019-01-24 10:11:34 - 2019-01-24 11:11:34
In Alarm: 0 yunjiankong.cloud.unittiao
Availability/Valid Request Proportion by User(%)
Period: 60s Method: Value
2019-01-24 10:34:00
Availability(by): 100.00
100
50
0
10:35
10:40
10:45
10:50
10:55
11:00
Availability by User
Valid Request Proportion by User
Number of Total/Valid Requests by User(times)
Period: 60s Method: Value
60
40
20
0
Number of Valid Requests by User
Total Number of Requests by User
Traffic by User(byte)
Period: 60s Method: Value
0
Internet Inbound Traffic by User
Internet Outbound Traffic by User
Intranet Inbound Traffic by User
Intranet Outbound Traffic by User
CDN Inbound Traffic by User
CDN Outbound Traffic by User
Cross-region Sync Inbound Traffic by User
Cross-region Sync Outbound Traffic by User

- Request Status Details

  This section displays the monitoring of request state distribution by using the following charts:

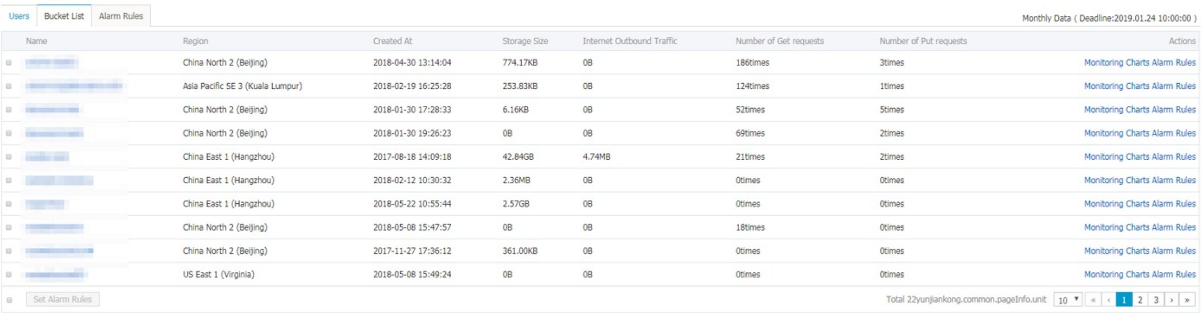
  - Number of Server Error Requests by User(times)
  - Server Error Request Proportion by User(%)
  - Number of Network Error Requests by User(times)
  - Network Error Request Proportion by User(%)
  - Number of Client Error Requests by User(times), which includes the following metrics: the number of error requests that indicate resources not found, number of authorization error requests, number of client-side timeout error requests, and number of other client-side error requests.
  - Client Error Request Proportion by User(%), which includes the following metrics: the percentage of error requests that indicate resources not found, percentage of authorization error requests, percentage of client-side timeout error requests, and percentage of other client-side error requests.
  - Number of Valid Requests by User(times), which includes the following metrics: the number of successful requests and number of redirect requests.
  - Valid Request Proportion by User(%), which includes the following metrics: the percentage of successful requests and percentage of redirect requests.


Monitoring Service Overview
Request Status Details
1h
6h
12h
1days
7days
2019-01-24 10:11:34 - 2019-01
yunjiankong.cloud.unittiao
Number of Server Error Requests by User(times)
Period: 60s Method: Value
Number of Server Error Requests by User
Server Error Request Proportion by User(%)
Period: 60s Method: Value
2019-01-24 10:34:00
Server(Error): 0.00
Server Error Request Proportion by User
Number of Network Error Requests by User(times)
Period: 60s Method: Value
Number of Network Error Requests by User
Network Error Request Proportion by User(%)
Period: 60s Method: Value
Network Error Request Proportion by User
Number of Client Error Requests by User(times)
Period: 60s Method: Value
Number of Resource Not Found Error Requests by User
Number of Authorization Error Requests by User
Number of Client Timeout Error Requests by User
Number of Client Other Error Requests by User
Client Error Request Proportion by User(%)
Period: 60s Method: Value
Resource Not Found Error Request Proportion by User
Authorization Error Request Proportion by User
Client Timeout Error Request Proportion by User
Client Other Error Request Proportion by User
Number of Valid Requests by User(times)
Period: 60s Method: Value
Number of Successful Requests by User
Number of Redirected Requests by User
Valid Request Proportion by User(%)
Period: 60s Method: Value
Successful Request Proportion by User
Redirected Request Proportion by User
10:34
10:36
10:38
10:40
10:42
10:44
10:46
10:48
10:50
10:52
10:54
10:56
10:58
11:00
11:02
11:04

Bucket List

- Bucket information list

  The Bucket List tab displays the information about the bucket, including the bucket name, region, creation time, statistics of the current month, and related operations.



Users Bucket List Alarm Rules

Monthly Data (Deadline:2019.01.24 10:00:00)

Name	Region	Created At	Storage Size	Internet Outbound Traffic	Number of Get requests	Number of Put requests	Actions
[redacted]	China North 2 (Beijing)	2018-04-30 13:14:04	774.17KB	0B	186times	3times	Monitoring Charts Alarm Rules
[redacted]	Asia Pacific SE 3 (Kuala Lumpur)	2018-02-19 16:25:28	253.83KB	0B	124times	1times	Monitoring Charts Alarm Rules
[redacted]	China North 2 (Beijing)	2018-01-30 17:28:33	6.16KB	0B	52times	5times	Monitoring Charts Alarm Rules
[redacted]	China North 2 (Beijing)	2018-01-30 19:26:23	0B	0B	69times	2times	Monitoring Charts Alarm Rules
[redacted]	China East 1 (Hangzhou)	2017-08-18 14:09:18	42.84GB	4.74MB	21times	2times	Monitoring Charts Alarm Rules
[redacted]	China East 1 (Hangzhou)	2018-02-12 10:30:32	2.36MB	0B	0times	0times	Monitoring Charts Alarm Rules
[redacted]	China East 1 (Hangzhou)	2018-05-22 10:55:44	2.57GB	0B	0times	0times	Monitoring Charts Alarm Rules
[redacted]	China North 2 (Beijing)	2018-05-08 15:47:57	0B	0B	18times	0times	Monitoring Charts Alarm Rules
[redacted]	China North 2 (Beijing)	2017-11-27 17:36:12	361.00KB	0B	0times	0times	Monitoring Charts Alarm Rules
[redacted]	US East 1 (Virginia)	2018-05-08 15:49:24	0B	0B	0times	0times	Monitoring Charts Alarm Rules

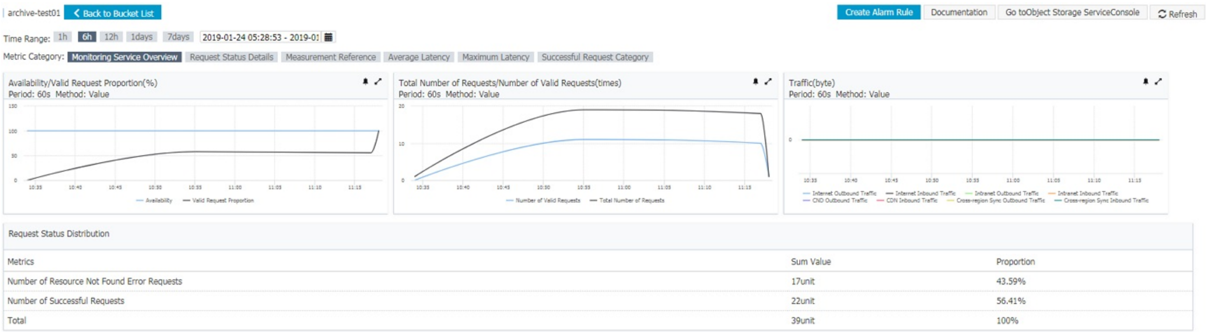
Set Alarm Rules   Total 22yunjiankong.common.pageInfo.unit 10 ▾ « ‹ 1 2 3 › »

  - Statistics of the current month include the storage size, public outbound traffic, the number of PUT requests, and the number of GET requests for each bucket.

- ○ Click the name of a bucket to go to the bucket monitoring overview page. You can also click Monitoring Charts in the Actions column corresponding to the bucket to go to the bucket monitoring overview page.
- ○ Click Alarm Rules in the Actions column corresponding to the bucket to go to the Alarm Rules tab. This tab displays all alert rules for the bucket.
- ○ In the upper-left corner, enter a keyword in the search box for fuzzy match.
- ○ Select the check boxes in front of bucket names and click Set Alarm Rules to configure alert rules at a time. For more information, see Alert service.

- Bucket-level metrics

  In the list of buckets, click **Monitoring Charts** in the Actions column corresponding to a bucket. The bucket monitoring overview page appears.



Request Status Distribution

Metrics	Sum Value	Proportion
Number of Resource Not Found Error Requests	17unit	43.59%
Number of Successful Requests	22unit	56.41%
Total	39unit	100%

The bucket monitoring overview page displays metric charts based on the following metric groups:

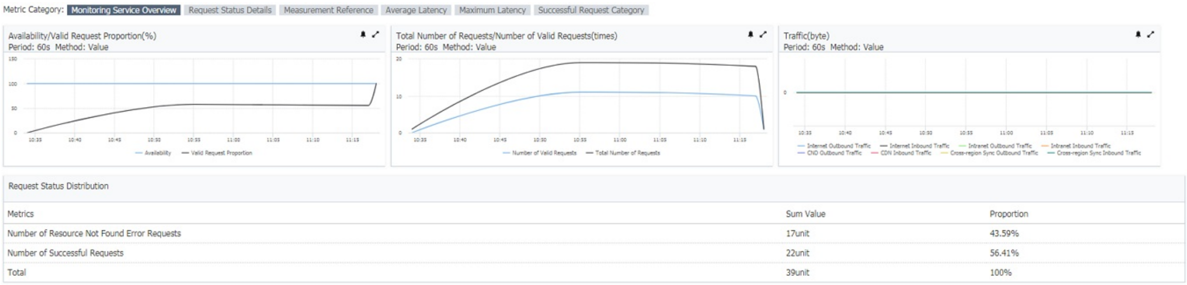
- ○ Monitoring Service Overview
- ○ Request Status Details
- ○ Measurement Reference
- ○ Average Latency
- ○ Maximum Latency
- ○ Successful Request Category

All metrics, except for metered metrics, are displayed at an aggregation granularity of 60s. By default, bucket-level charts display data of the last six hours, and user-level chart display data of the last hour. Click **Back to Bucket List**. The Bucket List tab appears.

- ○ Monitoring Service Overview

  This metric group is similar to the service monitoring overview at the user level, but displays metric data at the bucket level. The following metric charts are included:

  - ■ Availability/Valid Request Proportion(%), which includes the following metrics: availability and percentage of valid requests.
  - ■ Total Number of Requests/Number of Valid Requests, which includes the following metrics: the total number of requests and number of valid requests.
  - ■ Traffic(byte), which includes the following metrics: the public outbound traffic, public inbound traffic, internal outbound traffic, internal inbound traffic, CDN outbound traffic, CDN inbound traffic, cross-region replication outbound traffic, and cross-region replication inbound traffic.
  - ■ Request Status Distribution, which includes the following metrics: the number and percentage of each type of requests within the specified time range.



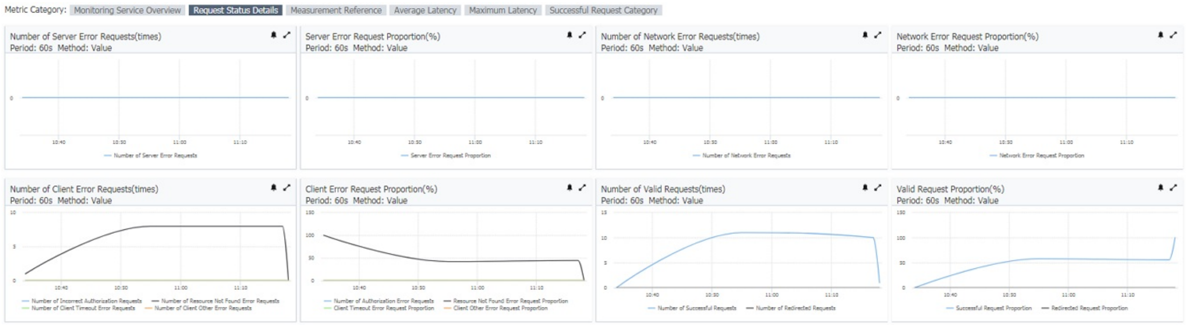
Request Status Distribution

Metrics	Sum Value	Proportion
Number of Resource Not Found Error Requests	17unit	43.59%
Number of Successful Requests	22unit	56.41%
Total	39unit	100%

- Request Status Details

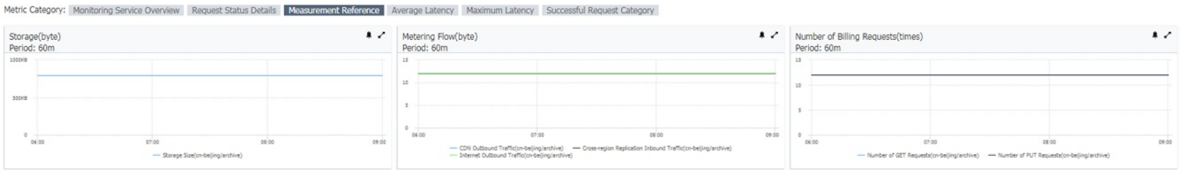
  This metric group displays the monitoring of request state distribution by using the following charts:

  - Number of Server Error Requests(times)
  - Server Error Request Proportion(%)
  - Number of Network Error Requests(times)
  - Network Error Request Proportion(%)
  - Number of Client Error Requests(times), which includes the following metrics: the number of error requests that indicate resources not found, number of authorization error requests, number of client-side timeout error requests, and number of other client-side error requests.
  - Client Error Request Proportion(%), which includes the following metrics: the percentage of error requests that indicate resources not found, percentage of authorization error requests, percentage of client-side timeout error requests, and percentage of other client-side error requests.
  - Number of Valid Requests(times), which includes the following metrics: the number of successful requests and number of redirect requests.
  - Valid Request Proportion(%), which includes the following metrics: the percentage of successful requests and percentage of redirect requests.


Metric Category:
Monitoring Service Overview
Request Status Details
Measurement Reference
Average Latency
Maximum Latency
Successful Request Category
Number of Server Error Requests(times)
Period: 60s Method: Value
0
10:40
10:50
11:00
11:10
Number of Server Error Requests
Server Error Request Proportion(%)
Period: 60s Method: Value
Server Error Request Proportion
Number of Network Error Requests(times)
Period: 60s Method: Value
Number of Network Error Requests
Network Error Request Proportion(%)
Period: 60s Method: Value
Network Error Request Proportion
Number of Client Error Requests(times)
Period: 60s Method: Value
10
5
Number of Incorrect Authorization Requests
Number of Resource Not Found Error Requests
Number of Client Timeout Error Requests
Number of Client Other Error Requests
Client Error Request Proportion(%)
Period: 60s Method: Value
150
100
50
Number of Authorization Error Requests
Resource Not Found Error Request Proportion
Client Timeout Error Request Proportion
Client Other Error Request Proportion
Number of Valid Requests(times)
Period: 60s Method: Value
15
Number of Successful Requests
Number of Redirected Requests
Valid Request Proportion(%)
Period: 60s Method: Value
Successful Request Proportion
Redirected Request Proportion

- Measurement Reference

  This metric group displays metric information on an hourly basis.


Metric Category:
Monitoring Service Overview
Request Status Details
Measurement Reference
Average Latency
Maximum Latency
Successful Request Category
Storage(byte)
Period: 60m
1000KB
500KB
0
06:00
07:00
08:00
09:00
Storage Size(cn-beijing/archive)
Metering Flow(byte)
Period: 60m
15
10
5
CDN Outbound Traffic(cn-beijing/archive)
Cross-region Replication Inbound Traffic(cn-beijing/archive)
Internet Outbound Traffic(cn-beijing/archive)
Number of Billing Requests(times)
Period: 60m
Number of GET Requests(cn-beijing/archive)
Number of PUT Requests(cn-beijing/archive)

  The following metered metric charts are included:

  - Storage(byte)
  - Metering Flow(byte)
  - Number of Billing Requests(times), which includes the following metrics: the number of PUT requests and the number of GET requests.

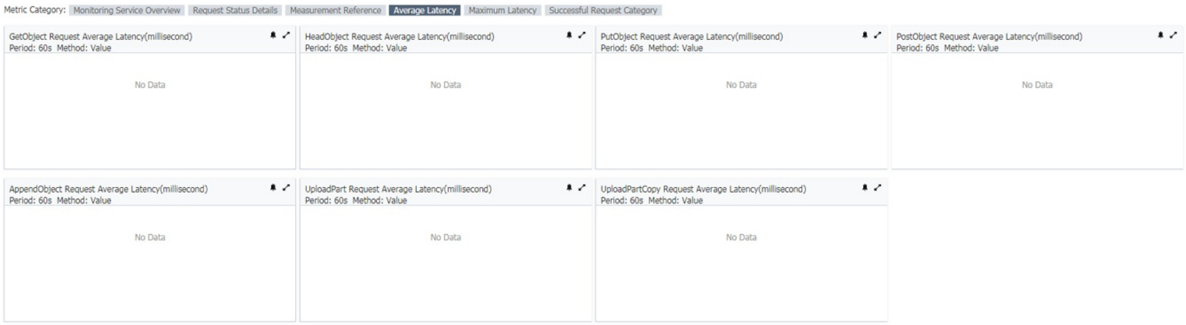
  After a bucket is created, statistics are collected beginning on the hour of the hour after the current time. The collected statistics are displayed within 30 minutes.

- Average Latency

  This metric group displays the charts for the average latency of API operations. The metric charts include:

  - GetObject Request Average Latency(millisecond)
  - HeadObject Request Average Latency(millisecond)
  - PutObject Request Average Latency(millisecond)
  - PostObject Request Average Latency(millisecond)
  - AppendObject Request Average Latency(millisecond)
  - UploadPart Request Average Latency(millisecond)
  - UploadPartCopy Request Average Latency(millisecond)

  Each chart shows the corresponding average end-to-end latency and average server latency.

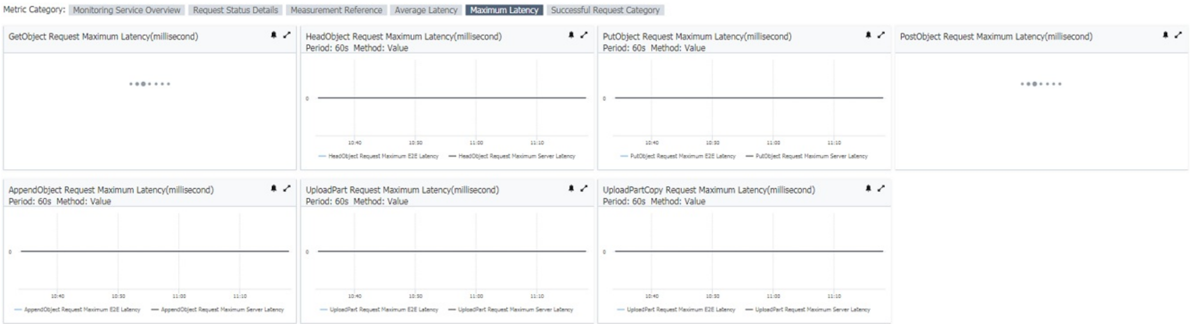

Metric Category:
Monitoring Service Overview
Request Status Details
Measurement Reference
Average Latency
Maximum Latency
Successful Request Category
GetObject Request Average Latency(millisecond)
Period: 60s Method: Value
No Data
HeadObject Request Average Latency(millisecond)
Period: 60s Method: Value
No Data
PutObject Request Average Latency(millisecond)
Period: 60s Method: Value
No Data
PostObject Request Average Latency(millisecond)
Period: 60s Method: Value
No Data
AppendObject Request Average Latency(millisecond)
Period: 60s Method: Value
No Data
UploadPart Request Average Latency(millisecond)
Period: 60s Method: Value
No Data
UploadPartCopy Request Average Latency(millisecond)
Period: 60s Method: Value
No Data

- Maximum Latency

  This metric group displays charts for the maximum latency metrics of API operations. The metric charts include:

  - GetObject Request Maximum Latency(millisecond)
  - HeadObject Request Maximum Latency(millisecond)
  - PutObject Request Maximum Latency(millisecond)
  - PostObject Request Maximum Latency(millisecond)
  - AppendObject Request Maximum Latency(millisecond)
  - UploadPart Request Maximum Latency(millisecond)
  - UploadPartCopy Request Maximum Latency(millisecond)

  Each chart shows the corresponding maximum end-to-end latency and maximum server latency.

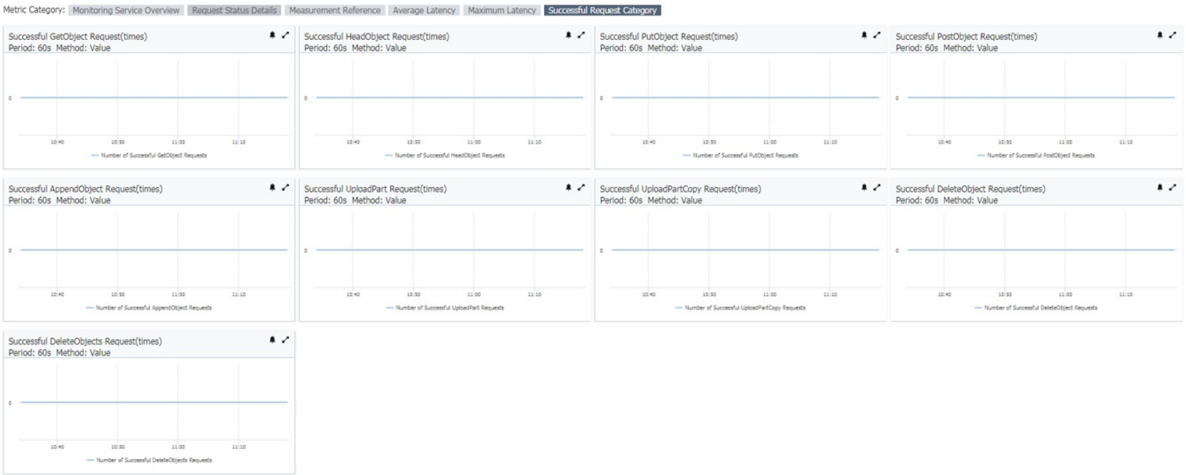

Metric Category:
Monitoring Service Overview
Request Status Details
Measurement Reference
Average Latency
Maximum Latency
Successful Request Category
GetObject Request Maximum Latency(millisecond)
HeadObject Request Maximum Latency(millisecond)
Period: 60s Method: Value
PutObject Request Maximum Latency(millisecond)
Period: 60s Method: Value
PostObject Request Maximum Latency(millisecond)
0
10:40
10:50
11:00
11:10
HeadObject Request Maximum E2E Latency
HeadObject Request Maximum Server Latency
PutObject Request Maximum E2E Latency
PutObject Request Maximum Server Latency
AppendObject Request Maximum Latency(millisecond)
Period: 60s Method: Value
UploadPart Request Maximum Latency(millisecond)
Period: 60s Method: Value
UploadPartCopy Request Maximum Latency(millisecond)
Period: 60s Method: Value
AppendObject Request Maximum E2E Latency
AppendObject Request Maximum Server Latency
UploadPart Request Maximum E2E Latency
UploadPart Request Maximum Server Latency
UploadPart Request Maximum E2E Latency
UploadPart Request Maximum Server Latency

- Successful Request Category

  This metric group displays charts for the successful requests of API operations. The charts include:

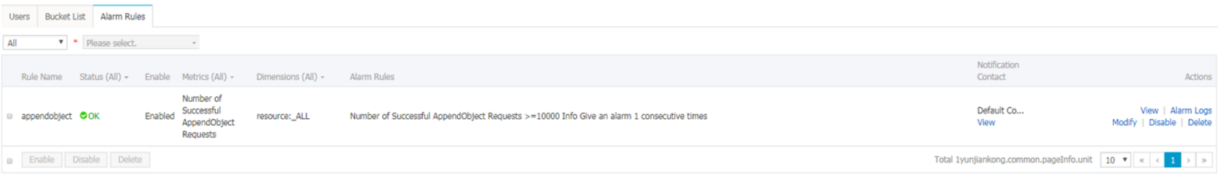
  - Successful GetObject Requests(times)
  - Successful HeadObject Requests(times)
  - Successful PutObject Requests(times)
  - Successful PostObject Requests(times)
  - Successful AppendObject Requests(times)
  - Successful UploadPart Requests(times)
  - Successful UploadPartCopy Requests(times)
  - Successful DeleteObject Requests(times)
  - Successful DeleteObjects Requests(times)

  The following figure shows an example.


Metric Category:
Monitoring Service Overview
Request Status Details
Measurement Reference
Average Latency
Maximum Latency
Successful Request Category
Successful GetObject Request(times)
Successful HeadObject Request(times)
Successful PutObject Request(times)
Successful PostObject Request(times)
Successful AppendObject Request(times)
Successful UploadPart Request(times)
Successful UploadPartCopy Request(times)
Successful DeleteObject Request(times)
Successful DeleteObjects Request(times)
Period: 60s Method: Value
0
10:40
10:50
11:00
11:10
Number of Successful GetObject Requests
Number of Successful HeadObject Requests
Number of Successful PutObject Requests
Number of Successful PostObject Requests
Number of Successful AppendObject Requests
Number of Successful UploadPart Requests
Number of Successful UploadPartCopy Requests
Number of Successful DeleteObject Requests
Number of Successful DeleteObjects Requests

Alarm Rules

The Alarm Rules tab allows you to view and manage your alert rules.



	Rule Name	Status (All)	Enable	Metrics (All)	Dimensions (All)	Alarm Rules	Notification Contact	Actions
□	appendobject	OK	Enabled	Number of Successful AppendObject Requests	resource:_ALL	Number of Successful AppendObject Requests >=10000 Info Give an alarm 1 consecutive times	Default Co... View	View | Alarm Logs Modify | Disable | Delete

For more information about the Alarm Rules tab, see Alert service.

References

For more information about the precautions and user guide for monitoring service, see Service monitoring, diagnosis, and troubleshooting.

11.3. Alert service

This topic describes alert rules for monitoring OSS in the CloudMonitor console and how to set such alert rules.

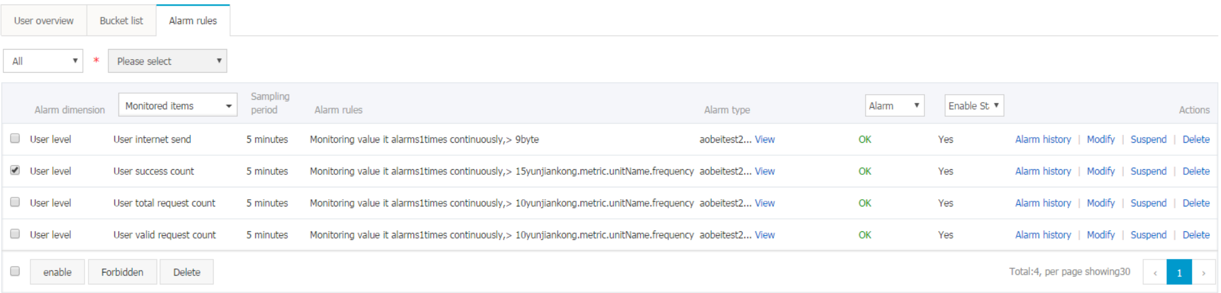
Before getting started with OSS alert rules, you can read the following topics to get familiar with the basic concepts of the alert service and the configurations of alert contacts and alert contact groups.

- Alert service overview
- Manage alert contacts and alert contact groups

OSS alert rules are developed based on OSS metrics. Therefore, OSS alert rules are categorized by dimensions similar to those of OSS metrics. Two alert dimensions are available: user-level and bucket-level.

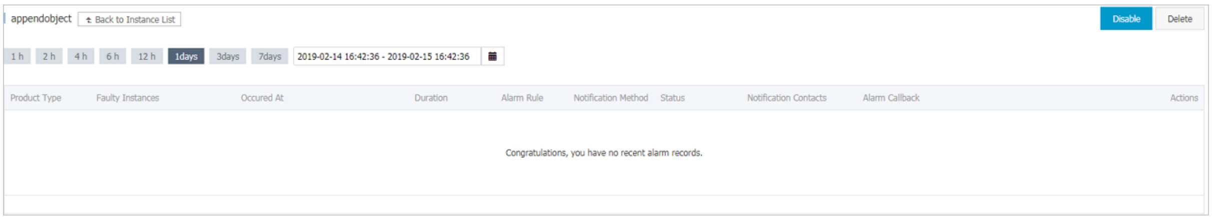
Alarm Rules tab

The CloudMonitor console provides an Alarm Rules tab for OSS monitoring and alerting. On this tab, you can view, modify, enable, disable, and delete alert rules. You can also view the historical alert information of a specific alert rule.



	Alarm dimension	Monitored items	Sampling period	Alarm rules	Alarm type	Alarm	Enable St	Actions
□	User level	User internet send	5 minutes	Monitoring value it alarms1times continuously,> 9byte	aobeitest2... View	OK	Yes	Alarm history | Modify | Suspend | Delete
☑	User level	User success count	5 minutes	Monitoring value it alarms1times continuously,> 15yunjiankong.metric.unitName.frequency	aobeitest2... View	OK	Yes	Alarm history | Modify | Suspend | Delete
□	User level	User total request count	5 minutes	Monitoring value it alarms1times continuously,> 10yunjiankong.metric.unitName.frequency	aobeitest2... View	OK	Yes	Alarm history | Modify | Suspend | Delete
□	User level	User valid request count	5 minutes	Monitoring value it alarms1times continuously,> 10yunjiankong.metric.unitName.frequency	aobeitest2... View	OK	Yes	Alarm history | Modify | Suspend | Delete

- Click **View** in the Actions column corresponding to an alert rule to view its content.
- Click **Modify** in the Actions column corresponding to an alert rule to modify it.
- Click **Delete** in the Actions column corresponding to an alert rule to delete it. You can also select multiple alert rules and click **Delete** below the alert rule list to delete multiple alert rules at a time.
- If an alert rule is in the *Enabled* state, click **Disable** in the Actions column corresponding to the alert rule to disable it. After the alert rule is disabled, you no longer receive alert information for this rule. You can also select multiple alert rules and click **Disable** below the alert rule list to disable multiple alert rules at a time.
- If an alert rule is in the *Disabled* state, click **Enable** in the Actions column corresponding to the alert rule to enable it. The alert rule takes effect again to detect exceptions and send alert information. You can also select multiple alert rules and click **Enable** below the alert rule list to enable multiple alert rules at a time.
- Click **Alarm Logs** in the Actions column corresponding to an alert rule. You can view the information about historical alerts corresponding to this rule.



2019-02-14 16:42:36 - 2019-02-15 16:42:36

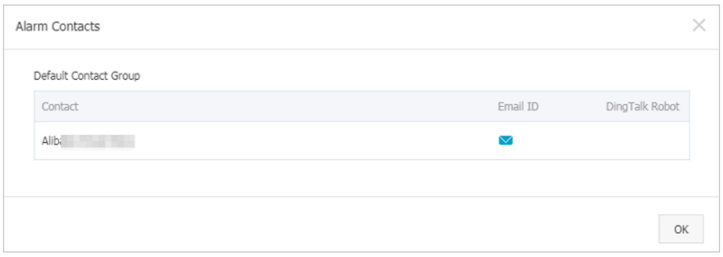
Product Type	Faulty Instances	Occured At	Duration	Alarm Rule	Notification Method	Status	Notification Contacts	Alarm Callback	Actions

Congratulations, you have no recent alarm records.

Relevant concepts:

- Historical alert information records the past status changes of a selected alert rule, such as a status change from OK to Alarm, a status change from Alarm to OK, and a special status change of Muted.
- When an alert rule is in the **Muted** state, the alert triggered by this alert rule remains active within a specified period and is not cleared. During the muted period, the system does not send alert information to notification contacts until the muted period expires.
- Historical alert information is kept for one month. Alerts generated earlier than one month are automatically deleted. You can query data of three days at most, and cannot query data generated 31 days ago.

You can click **View** in the Notification Contact column corresponding to an alert rule to view the members of notification contacts (in alert contact groups) and the methods that they use to receive alert information (such as SMS message, email, or TradeManager), as shown in the following figure.



Default Contact Group

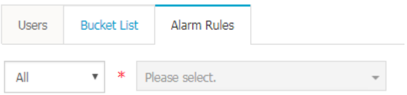
Contact	Email ID	DingTalk Robot
Alib[redacted]	✉	

View alert rules

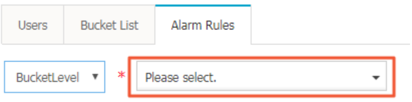
- Locate alert rules

  You can use the GUI elements on the Alarm Rules tab to locate the alert rules that you search for.

- Alert dimension drop-down list: You can select All or BucketLevel. If you select *All*, all user-level and bucket-level alert rules appear in the alert rule list.



- Bucket drop-down list: If you select *BucketLevel* from the alert dimension drop-down list, the bucket drop-down list displays all buckets of the current Alibaba Cloud account. You can select a bucket to view all alert rules for this bucket.



- Metrics drop-down list: It displays all OSS metrics, including user-level and bucket-level metrics. If you select *All*, user-level or bucket-level alert rules for all metrics appear in the alert rule list.
- Status drop-down list: You can select a status to view all alert rules in this state, such as OK, Alarm, Insufficient Data, Enable, or Disable. If you select All, alert rules in all statuses appear in the alert rule list.

- View all alert rules

  If you click the **Alarm Rules** tab, all alert rules automatically appear in the alert rule list. You can also select *All* from the alert dimension drop-down list to view all alert rules. Then, you can use the Metrics and Status drop-down lists to set filtering conditions and accurately locate alert rules in this alert dimension.

- View alert rules for a specific bucket

  To view the alert rules for a specific bucket, you need to select BucketLevel from the alert dimension drop-down list and select the name of the destination bucket from the bucket drop-down list. You can also click the **Bucket List** tab. On the tab that appears, you can click **Alarm Rules** in the Actions column corresponding to the relevant bucket to go to the Alarm Rules tab, where you can view all alert rules for this bucket. Then, you can use the Metrics and Status drop-down lists to set filtering conditions and accurately locate alert rules in this alert dimension.

- View alert rules related to a specific metric

  You can select a metric from the Metrics drop-down list to view all alert rules for this metric.

- View alert rules in a certain alert state

  You can select an alert status from the Status drop-down list, such as Alarm, to view all alert rules that are in this status.

Add an alert rule

1. Use any of the following methods to go to the Create Alarm Rule page:
   - On the Users tab, click **Monitoring Service Overview** and click

     🔔

     in any chart.
   - On the Bucket List tab, click the relevant bucket name to go to the bucket details page. Click **Create Alarm Rule**.
   - On the Bucket List tab, click the relevant bucket name. On the page that appears, click **Monitoring Service Overview** and click

     🔔

     in any chart.
2. Set the alert rule as needed.
   - **Related Resource**
     - *Products*: Select *Object Storage Service*.
     - *Resource Range*: Select *All Resources* or *bucketDimensions* as needed.
     - *Bucket* (if you set Resource Range to bucketDimensions): Select one or more buckets as needed.
   - **Set Alarm Rules**
     - **Alarm Rule**: Enter the alert rule name.
     - **Rule Describe**: Select the content, time, and threshold of the metric as needed.
     - **+Add Alarm Rule**: Click it to add more alert rules.
     - **Mute for**: Specify the interval for sending an alert notification if the exception persists after the alert is triggered.
     - **Triggered when threshold is exceeded for**: Specify the times that the rule is matched consecutively to send alert notifications. For example, if you select Internet Outbound Traffic, 1mins, Value, >, and 100 for Rule Describe and set Triggered when threshold is exceeded for to 3, an alert is triggered only when the Internet outbound traffic exceeds 100 MB three consecutive times within 1 minute.
     - *Effective Period*: Select the time the alert rule takes effect.
   - **Notification Method**
     - **Notification Contact**: If you have set an alert contact group by following the procedure in Manage alert contacts and alert contact groups, select the group. If you have not set any alert contact groups, click **Quickly create a contact group** to create a group by following the instructions.
     - *Notification Methods*: Select notification methods for the alert rule.
     - **Email Subject**: Enter the subject of the notification email.
     - **Email Remark**: optional. Enter the remarks of the email.
     - **HTTP CallBack**: Enter a URL that can be accessed from the Internet. CloudMonitor sends a POST request to push the alert notification to this URL. Currently, only HTTP is supported.
3. Click **Confirm** to complete the alert rule setting.

Notes

Currently, alert rules for a bucket are not associated with the existence of the bucket. If you delete a bucket, alert rules for this bucket still exist. Therefore, we recommend that you delete alert rules for a bucket before deleting this bucket.

11.4. Access monitoring data

The monitoring service of Object Storage Service (OSS) provides metrics such as the running status, performance, and metering of the system. These metrics help you track requests, analyze usage, collect statistics for business trends, and discover and diagnose system issues at your earliest opportunity. This topic describes the parameters used to query OSS monitoring data by using the API or SDKs provided by Cloud Monitor.

Note For more information about Cloud Monitor SDK examples, see CloudMonitor SDK for Java.

Space

Space is used to specify the cloud services to be monitored. The namespace used by the monitoring service of OSS is `acs_oss_dashboard`.

The following SDK for Java provides an example on how to specify the monitoring service of OSS:

```
DescribeMetricListRequest request = new DescribeMetricListRequest();
request.setNamespace("acs_oss_dashboard");
```

StartTime and EndTime

StartTime and EndTime specify the time range to query monitoring data. The value range of time parameters in Cloud Monitor uses a left-open and right-closed interval in the `(StartTime, EndTime]` format. You can query data based on the range from StartTime to EndTime (including the EndTime value). The interval between start time and end time cannot exceed 31 days. Only data generated within the last 31 days can be queried.

The following SDK for Java provides an example on how to query monitoring data based the specified time range:

```
// Specify the end time based on which to query monitoring data.
request.setEndTime("2019-05-13 11:06:27");
// Specify the start time based on which to query monitoring data.
request.setStartTime("2019-05-13 10:20:27");
```

Dimensions

Dimensions specifies the name of the bucket you want to query. If you do not specify Dimensions, data at the user level is queried. For more information about levels, see Metric.

The following SDK for Java provides an example on how to query bucket-level data:

```
// Specify the name of the bucket you want to query.
request.setDimensions("{\"BucketName\":\"<yourBucketName>\"}");
```

Period

Period specifies the period during which to query metrics. The period for metering metrics of the OSS monitoring service is 3,600s. The period for other metrics is 60s. For more information about the metrics, see Metric.

The following SDK for Java provides an example on how to monitor a specified non-metering metric:

```
request.setPeriod("60");
```

Metric

Metric is used to specify the metric you want to query. The following Java SDK code provides an example on how to specify the metric you want to query:

```
// Set the name of the metric.
request.setMetric("<MetricName>");
```

Metrics include non-metering metrics and metering metrics.

- Non-metering metrics

Level	Metric	Metric name	Unit
	UserAvailability	Availability	%
	UserRequestValidRate	Valid request rate	%
	UserTotalRequestCount	Total number of requests	Count
	UserValidRequestCount	Number of valid requests	Count
	UserInternetSend	Internet outbound traffic	Byte
	UserInternetRecv	Internet inbound traffic	Byte
	UserIntranetSend	Internal network outbound traffic	Byte
	UserIntranetRecv	Internal network inbound traffic	Byte
	UserCdnSend	CDN outbound traffic	Byte
	UserCdnRecv	CDN inbound traffic	Byte

Level	Metric	Metric name	Unit
User level	UserSyncSend	Cross-region replication (CRR) outbound traffic	Byte
	UserSyncRecv	CRR inbound traffic	Byte
	UserServerErrorCount	Number of server-side error requests	Count
	UserServerErrorRate	Server-side error request rate	%
	UserNetworkErrorCount	Number of network-side error requests	Count
	UserNetworkErrorRate	Network-side error request rate	%
	UserAuthorizationErrorCount	Number of client-side authorization error requests	Count
	UserAuthorizationErrorRate	Client-side authorization error request rate	%
	UserResourceNotFoundErrorCount	Number of client-side error requests that indicate resources not found	Count
	UserResourceNotFoundErrorRate	Rate of client-side error requests that indicate resources not found	%
	UserClientTimeoutErrorCount	Number of client-side timeout error requests	Count
	UserClientOtherErrorRate	Client-side timeout error request rate	%
	UserClientOtherErrorCount	Number of other client-side error requests	Count
	UserClientOtherErrorRate	Rate of other client-side error requests	%
	UserSuccessCount	Number of successful requests	Count
	UserSuccessRate	Successful request rate	%
	UserRedirectCount	Number of redirect requests	Count
	UserRedirectRate	Redirect request rate	%
	Availability	Availability	%
	RequestValidRate	Valid request rate	%
	TotalRequestCount	Total number of requests	Count
	ValidRequestCount	Number of valid requests	Count
	InternetSend	Internet outbound traffic	Byte
	InternetRecv	Internet inbound traffic	Byte
	IntranetSend	Internal network outbound traffic	Byte
	IntranetRecv	Internal network inbound traffic	Byte
	InternetSendBandwidth	Outbound bandwidth over Internet	bps
	InternetRecvBandwidth	Inbound bandwidth over Internet	bps
	IntranetSendBandwidth	Outbound bandwidth over the internal network	bps
	IntranetRecvBandwidth	Inbound bandwidth over the internal network	bps
	CdnSend	CDN outbound traffic	Byte
	CdnRecv	CDN inbound traffic	Byte
	SyncSend	Cross-region replication (CRR) outbound traffic	Byte
	SyncRecv	CRR inbound traffic	Byte
	ServerErrorCount	Number of server-side error requests	Count
	ServerErrorRate	Server-side error request rate	%
	NetworkErrorCount	Number of network-side error requests	Count
	NetworkErrorRate	Network-side error request rate	%

Level	Metric	Metric name	Unit
Bucket level	AuthorizationErrorCount	Number of client-side authorization error requests	Count
	AuthorizationErrorRate	Client-side authorization error request rate	%
	ResourceNotFoundErrorCount	Number of client-side error requests that indicate resources not found	Count
	ResourceNotFoundErrorRate	Rate of client-side error requests that indicate resources not found	%
	ClientTimeoutErrorCount	Number of client-side timeout error requests	Count
	ClientTimeoutErrorRate	Client-side timeout error request rate	%
	ClientOtherErrorCount	Number of other client-side error requests	Count
	ClientOtherErrorRate	Rate of other client-side error requests	%
	SuccessCount	Number of successful requests	Count
	SuccessRate	Successful request rate	%
	RedirectCount	Number of redirect requests	Count
	RedirectRate	Redirect request rate	%
	GetObjectE2eLatency	Average end-to-end latency of GetObject requests	Millisecond
	GetObjectServerLatency	Average server latency of GetObject requests	Millisecond
	MaxGetObjectE2eLatency	Maximum end-to-end latency of GetObject requests	Millisecond
	MaxGetObjectServerLatency	Maximum server latency of GetObject requests	Millisecond
	HeadObjectE2eLatency	Average end-to-end latency of HeadObject requests	Millisecond
	HeadObjectServerLatency	Average server latency of HeadObject requests	Millisecond
	MaxHeadObjectE2eLatency	Maximum end-to-end latency of HeadObject requests	Millisecond
	MaxHeadObjectServerLatency	Maximum server latency of HeadObject requests	Millisecond
	PutObjectE2eLatency	Average end-to-end latency of PutObject requests	Millisecond
	PutObjectServerLatency	Average server latency of PutObject requests	Millisecond
	MaxPutObjectE2eLatency	Maximum end-to-end latency of PutObject requests	Millisecond
	MaxPutObjectServerLatency	Maximum server latency of PutObject requests	Millisecond
	PostObjectE2eLatency	Average end-to-end latency of PostObject requests	Millisecond
	PostObjectServerLatency	Average server latency of PostObject requests	Millisecond
	MaxPostObjectE2eLatency	Maximum end-to-end latency of PostObject requests	Millisecond
	MaxPostObjectServerLatency	Maximum server latency of PostObject requests	Millisecond
	AppendObjectE2eLatency	Average end-to-end latency of AppendObject requests	Millisecond
	AppendObjectServerLatency	Average server latency of AppendObject requests	Millisecond
	MaxAppendObjectE2eLatency	Maximum end-to-end latency of AppendObject requests	Millisecond

Level	Metric	Metric name	Unit
	MaxAppendObjectServerLatency	Maximum server latency of AppendObject requests	Millisecond
	UploadPartE2eLatency	Average end-to-end latency of UploadPart requests	Millisecond
	UploadPartServerLatency	Average server latency of UploadPart requests	Millisecond
	MaxUploadPartE2eLatency	Maximum end-to-end latency of UploadPart requests	Millisecond
	MaxUploadPartServerLatency	Maximum server latency of UploadPart requests	Millisecond
	UploadPartCopyE2eLatency	Average end-to-end latency of UploadPartCopy requests	Millisecond
	UploadPartCopyServerLatency	Average server latency of UploadPartCopy requests	Millisecond
	MaxUploadPartCopyE2eLatency	Maximum end-to-end latency of UploadPartCopy requests	Millisecond
	MaxUploadPartCopyServerLatency	Maximum server latency of UploadPartCopy requests	Millisecond
	GetObjectCount	Number of successful GetObject requests	Count
	HeadObjectCount	Number of successful HeadObject requests	Count
	PutObjectCount	Number of successful PutObject Requests	Count
	PostObjectCount	Number of successful PostObject requests	Count
	AppendObjectCount	Number of successful AppendObject requests	Count
	UploadPartCount	Number of successful UploadPart requests	Count
	UploadPartCopyCount	Number of successful UploadPartCopy requests	Count
	DeleteObjectCount	Number of successful DeleteObject requests	Count
	DeleteObjectsCount	Number of successful DeleteObjects requests	Count

- Metering metrics

  If you specify Dimensions when you query the following metrics, data at the bucket level is queried. If you do not specify Dimensions, data at the user level is queried.

Metric	Metric name	Unit
MeteringStorageUtilization	Storage size	Byte
MeteringGetRequest	Number of GET requests	Count
MeteringPutRequest	Number of PUT requests	Count
MeteringInternetTX	Metered Internet outbound traffic	Byte
MeteringCdnTX	Metered CDN outbound traffic	Byte
MeteringSyncRX	Metered CRR inbound traffic	Byte

11.5. Metrics

The Object Storage Service (OSS) monitoring service provides user- and bucket-level metrics for different scenarios. OSS analyzes existing metrics and collects statistics generated during a period of time, such as request status distribution by user and monthly statistics. Then, OSS generates a statistical analysis that can facilitate metric data analysis and the application of billing policies.

Note

- Metrics such as sums, maximum values, and average values are reported at one-minute intervals. Billable usage metrics are reported at one-hour intervals.
- CloudMonitor provides a default dashboard that displays monitoring data for Elastic Compute Service (ECS) instances. If you want to view OSS user- and bucket-level metrics, click Add View in the CloudMonitor console to add the metrics to the dashboard. For more information, see Manage the monitoring charts of a custom dashboard.

User-level metrics

User-level metrics describe the overall usage and information about all OSS buckets owned by an Alibaba Cloud account. User-level metrics include monthly statistics, monitoring service overview, and request status details.

- The metrics for monthly statistics display the storage usage of OSS resources that you have used between 00:00:00 on the first day of the current month and the current time you view the collection. The following table describes the metrics.

Metric name	Unit	Description
Storage size	Byte	The total storage size used by all the buckets owned by the Alibaba Cloud account between 00:00:00 on the first day of the current month and the current time when you view the collection.
Internet outbound traffic	Byte	The total amount of Internet outbound traffic sent by the user between 00:00:00 on the first day of the current month and the current time when you view the collection.
Number of PUT requests	Count	The total number of PUT requests sent by the user between 00:00:00 on the first day of the current month and the current time you view the collection.
Number of GET requests	Count	The total number of GET requests sent by the user between 00:00:00 on the first day of the current month and the current time when you view the collection.

Monthly statistics

- Monitoring service overview shows basic service metrics. The following table describes the metrics.

Metric name	Unit	Description
Availability by User	%	The metric that describes the system availability of OSS. You can obtain the metric value based on the following formula: Metric value = `1 - Server error requests with the returned HTTP status code 5xx/All requests` .
Valid Request Proportion by User	%	The percentage of valid requests out of all requests.
Number of Total Requests by User	Count	The total number of requests that are received and processed by the OSS server.
Number of Valid Requests by User	Count	The total number of requests with the returned HTTP status codes 2xx and 3xx.
Internet Outbound Traffic by User	Byte	The amount of outbound traffic sent over the Internet.
Internet Inbound Traffic by User	Byte	The amount of inbound traffic received over the Internet.
Intranet Outbound Traffic by User	Byte	The amount of outbound traffic sent over the internal network.
Intranet Inbound Traffic by User	Byte	The amount of inbound traffic received over the internal network.
CDN Outbound Traffic by User	Byte	The amount of outbound traffic sent over Content Delivery Network (CDN) after CDN is activated. Such outbound traffic over CDN is back-to-origin traffic.
CDN Inbound Traffic by User	Byte	The amount of inbound traffic received over CDN after CDN is activated.
Cross-region Sync Outbound Traffic by User	Byte	The amount of outbound traffic generated during data replication after cross-region replication (CRR) is enabled.
Cross-region Sync Inbound Traffic by User	Byte	The amount of inbound traffic generated during data replication after CRR is enabled.

Monitoring service overview

- The metrics for request status details describe requests based on returned status codes and OSS error codes. The metrics are basic service metrics. The following table describes the metrics.

Metric name	Unit	Description
Number of Server Error Requests by User	Count	The total number of system-level error requests with the returned HTTP status code 5xx.
Server Error Request Proportion by User	%	The percentage of requests with server-side errors out of all requests.
Number of Network Error Requests by User	Count	The total number of requests with the returned HTTP status code 499.
Network Error Request Proportion by User	%	The percentage of requests with network errors out of all requests.
Number of Authorization Error Requests by User	Count	The total number of requests with the returned HTTP status code 403.
Authorization Error Request Proportion by User	%	The percentage of requests with authorization errors out of all requests.
Number of Resource Not Found Error Requests by User	Count	The total number of requests with the returned HTTP status code 404.

Metric name	Unit	Description
Resource Not Found Error Request Proportion by User	%	The percentage of requests with errors indicating resources not found out of all requests.
Number of Client Timeout Error Requests by User	Count	The total number of requests with the returned HTTP status code 408 or OSS error code RequestTimeout.
Client Timeout Error Request Proportion by User	%	The percentage of requests with client-side timeout errors out of all requests.
Number of Client Other Error Requests by User	Count	The total number of requests other than the preceding client-side error requests with the returned HTTP status code 4xx.
Client Other Error Request Proportion by User	%	The percentage of requests with other client-side errors out of all requests.
Number of Successful Requests by User	Count	The total number of requests with the returned HTTP status code 2xx.
Successful Request Proportion by User	%	The percentage of successful requests out of all requests.
Number of Redirected Requests by User	Count	The total number of requests with the returned HTTP status code 3xx.
Redirected Request Proportion by User	%	The percentage of redirected requests out of all requests.

Request status details

Bucket-level metrics

Bucket-level metrics include all the preceding user-level metrics and measurement and performance metrics, such as measurement reference, average latency, and successful request category.

Notice Bucket-level metrics indicate the performance of each bucket, which is different from user-level metrics. For example, the storage size metric of the user-level indicates the storage size used by all buckets owned by the Alibaba Cloud account between 00:00:00 on the first day of the current month and the current time you view the collection. The storage size metric of the bucket-level indicates the storage size used by a bucket during a specific period of time.

- The following table describes the metrics.

Metric name	Unit	Description
Storage Size	Byte	The amount of storage occupied by the bucket each hour.
Internet Outbound Traffic	Byte	The total amount of outbound traffic sent from the bucket over the Internet each hour.
Number of PUT Requests	Count	The total number of PUT requests made to the bucket each hour.
Number of GET Requests	Count	The total number of GET requests made to the bucket each hour.

Measurement reference

- Request latency indicates the system performance. The metric collects only the statistics of successful requests with the returned status code 2xx. The OSS monitoring service provides minute-level average latency and maximum latency metrics to indicate the system average response capability and thrashing.

  In addition, the latency metrics are displayed based on the end-to-end (E2E) latency and server latency, which help you analyze key performance and environmental issues.

  - E2E latency refers to the end-to-end latency of successful requests made to OSS. E2E latency includes the time required for OSS to read a request, send the response, and receive acknowledgment of the response.
  - Server latency refers to the time required for OSS to process a successful request. Server latency does not include the network latency specified in E2E latency.

  The following table describes the metrics.

Metric name	Unit	Description
GetObject Request Average E2E Latency	Millisecond	The average E2E latency of successful GetObject requests.
GetObject Request Average Server Latency	Millisecond	The average server latency of successful GetObject requests.
GetObject Request Maximum E2E Latency	Millisecond	The maximum end-to-end latency of successful GetObject requests.
GetObject Request Maximum Server Latency	Millisecond	The maximum server latency of successful GetObject requests.
HeadObject Request Average E2E Latency	Millisecond	The average E2E latency of successful HeadObject requests.
HeadObject Request Average Server Latency	Millisecond	The average server latency of successful HeadObject requests.
HeadObject Request Maximum E2E Latency	Millisecond	The maximum end-to-end latency of successful HeadObject requests.
HeadObject Request Maximum Server Latency	Millisecond	The maximum server latency of successful HeadObject requests.
PutObject Request Average E2E Latency	Millisecond	The average E2E latency of successful PutObject requests.
PutObject Request Average Server Latency	Millisecond	The average server latency of successful PutObject requests.
PutObject Request Maximum E2E Latency	Millisecond	The maximum end-to-end latency of successful PutObject requests.
PutObject Request Maximum Server Latency	Millisecond	The maximum server latency of successful PutObject requests.

Metric name	Unit	Description
PostObject Request Average E2E Latency	Millisecond	The average end-to-end latency of successful PostObject requests.
PostObject Request Average Server Latency	Millisecond	The average server latency of successful PostObject requests.
PostObject Request Maximum E2E Latency	Millisecond	The maximum end-to-end latency of successful PostObject requests.
PostObject Request Maximum Server Latency	Millisecond	The maximum server latency of successful PostObject requests.
AppendObject Request Average E2E Latency	Millisecond	The average end-to-end latency of successful AppendObject requests.
AppendObject Request Average Server Latency	Millisecond	The average server latency of successful AppendObject requests.
AppendObject Request Maximum E2E Latency	Millisecond	The maximum end-to-end latency of successful AppendObject requests.
AppendObject Request Maximum Server Latency	Millisecond	The maximum server latency of successful AppendObject requests.
UploadPart Request Average E2E Latency	Millisecond	The average end-to-end latency of successful UploadPart requests.
UploadPart Request Average Server Latency	Millisecond	The average server latency of successful UploadPart requests.
UploadPart Request Maximum E2E Latency	Millisecond	The maximum end-to-end latency of successful UploadPart requests.
UploadPart Request Maximum Server Latency	Millisecond	The maximum server latency of successful UploadPart requests.
UploadPartCopy Request Average E2E Latency	Millisecond	The average end-to-end latency of successful UploadPartCopy requests.
UploadPartCopy Request Average Server Latency	Millisecond	The average server latency of successful UploadPartCopy requests.
UploadPartCopy Request Maximum E2E Latency	Millisecond	The maximum end-to-end latency of successful UploadPartCopy requests.
UploadPartCopy Request Maximum Server Latency	Millisecond	The maximum server latency of successful UploadPartCopy requests.

Latency

- The metrics for successful request category indicate the capability of the system to process requests. The following table describes the metrics.

Metric name	Unit	Description
Successful GetObject Request	Count	The number of successful GetObject requests.
Successful HeadObject Request	Count	The number of successful HeadObject requests.
Successful PutObject Request	Count	The number of successful PutObject requests.
Successful PostObject Request	Count	The number of successful PostObject requests.
Successful AppendObject Request	Count	The number of successful AppendObject requests.
Successful UploadPart Request	Count	The number of successful UploadPart requests.
Successful UploadPartCopy Request	Count	The number of successful UploadPartCopy requests.
Successful DeleteObject Request	Count	The number of successful DeleteObject requests.
Successful DeleteObjects Request	Count	The number of successful DeleteObjects requests.

Successful request category

- The following table describes the metrics.

Metric name	Unit	Description
Mirror Inbound Traffic	Byte	The total amount of inbound traffic generated by requests with the returned value 200 and 206 for a specific origin.
Mirror Inbound Traffic by Status	Byte	The amount of inbound traffic generated by requests with a specific value for a specific origin.
Mirror Request Average Transfer Speed	Byte/s	The average transfer speed of requests with the returned value 200 and 206 for a specific origin.
Mirror Request Average Transfer Speed by Status	Byte/s	The average transfer speed of requests with a specific value for a specific origin.
Mirror Request Count	Count	The total number of requests with the returned value 200 and 206 for a specific origin.
Mirror Request Count by Status	Count	The total number of requests with a specific returned value for a specific origin.
Mirror Request Average Latency	Millisecond	The average latency of requests with the returned value 200 and 206 for a specific origin.
Mirror Request Average Latency by Status	Millisecond	The average latency of requests with a specific returned value for a specific origin.
Mirror Request Proportion by Status	%	The proportion of requests with a specific status code, such as 2xx, 3xx, 4xx, and 5xx, for a specific origin out of all requests.
Mirror Request with a Specific Status Code Count	Count	The number of requests with a specific status code, such as 2xx, 3xx, 4xx, and 5xx, for a specific origin.

Mirroring-based back-to-origin

11.6. Monitoring, diagnosis, and troubleshooting

Compared with traditional applications, cloud applications cost less in infrastructure investment and O&M. However, it is more difficult to monitor the running status and performance of cloud applications, locate faults, and troubleshoot faults. To solve this issue, Object Storage Service (OSS) provides the monitoring service and logging feature for you to monitor the performance of your application and locate faults.

This topic describes how to use the OSS monitoring service, the logging feature, and third-party tools to monitor, diagnose, and troubleshoot issues when you use OSS to store the data of your business. The OSS monitoring service serves the following purposes:

- Monitor the running status and performance of OSS in real time and send alert notifications.
- Provide effective methods and tools to locate issues.
- Solve issues based on relevant guides.

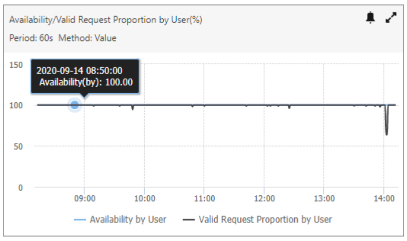
This topic includes the following sections:

- Monitoring service: describes how to use the OSS monitoring service to monitor the running status and performance of OSS.
- Tracking and diagnosis: describe how to use the OSS monitoring service and logging feature to track and diagnose faults.
- Troubleshooting: describes common issues and solutions to these issues.

Monitoring service

- Monitor the overall status
  - Availability/Valid Request Proportion by User (%)

    Availability/Valid Request Proportion by User is the most important metric used to indicate OSS stability and whether OSS is correctly used. A percentage less than 100% indicates that some requests fail.

    A percentage less than 100% may be caused by OSS optimization such as partition migration for load balancing. In this case, OSS SDKs provide relevant retry mechanisms to handle error requests due to this temporary optimization. This way, your business is not affected.

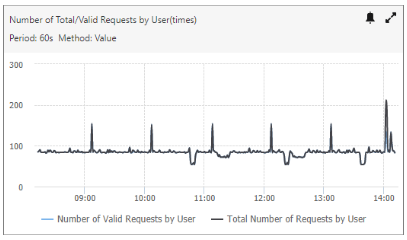
    To determine the types and causes of error requests and troubleshoot the errors, analyze the requests based on request status details and request status distribution in the Cloud Monitor console for OSS.

    In addition, it is expected that valid request proportion may be less than 100% in some business scenarios. For example, you need to send a request to check whether an object exists before you manage the object in some scenarios. If the object does not exist, a 404 error is returned for the request, which results in a valid request proportion lower than 100%. In this case, the error is returned as expected and does not indicate an actual issue.

    For businesses that require high availability of OSS, you can configure an alert rule that is triggered when the metric falls below an expected threshold value.

  - Number of Total/Valid Requests by User

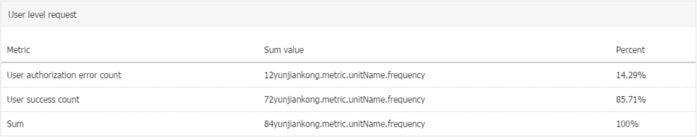
    The Number of Total/Valid Requests by User metric shows the running status of OSS in the aspect of request times. If the number of valid requests is smaller than that of total requests, it indicates that some requests fail.

    To follow the fluctuations of the number of valid requests, especially spikes and dips, and analyze the causes, you can configure alert rules to receive timely notifications. For more information, see Alert service.

  - Request Status Distribution by User

    When the availability/valid request proportion is less than 100%, or the number of valid requests is smaller than the number of total requests, you can view Request Status Distribution by User to determine the type of error requests. For more information about Request Status Distribution by User, see Metrics.

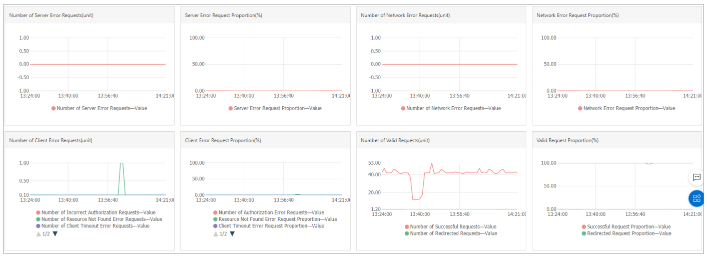
    

    User level request

Metric	Sum value	Percent
User authorization error count	12yunjiankong.metric.unitName.frequency	14.29%
User success count	72yunjiankong.metric.unitName.frequency	85.71%
Sum	84yunjiankong.metric.unitName.frequency	100%

- Monitor request status

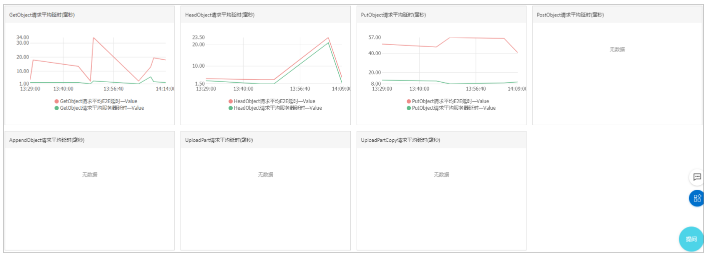
Request Status Details monitors different types of requests and provides more details about the monitoring based on the request status distribution.


Number of Server Error Requests(unit)
Number of Server Error Requests—Value
Server Error Request Proportion(%)
Server Error Request Proportion—Value
Number of Network Error Requests(unit)
Number of Network Error Requests—Value
Network Error Request Proportion(%)
Network Error Request Proportion—Value
Number of Client Error Requests(unit)
Number of Incorrect Authorization Requests—Value
Number of Resource Not Found Error Requests—Value
Number of Client Timeout Error Requests—Value
Client Error Request Proportion(%)
Number of Authorization Error Requests—Value
Resource Not Found Error Request Proportion—Value
Client Timeout Error Request Proportion—Value
Number of Valid Requests(unit)
Number of Successful Requests—Value
Number of Redirected Requests—Value
Valid Request Proportion(%)
Successful Request Proportion—Value
Redirected Request Proportion—Value
13:24:00
13:40:00
13:56:40
14:21:00

- Monitor performance

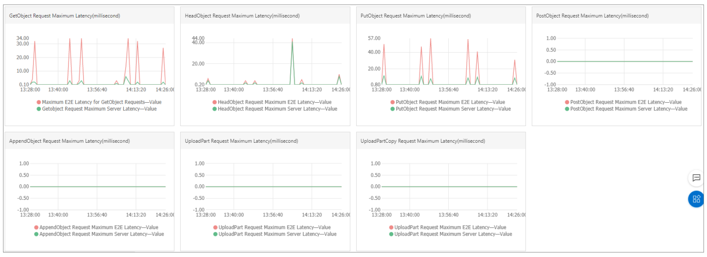
  The monitoring service provides the following metrics to help you monitor the performance of OSS:

  - Average Latency, which includes average E2E latency and average server latency

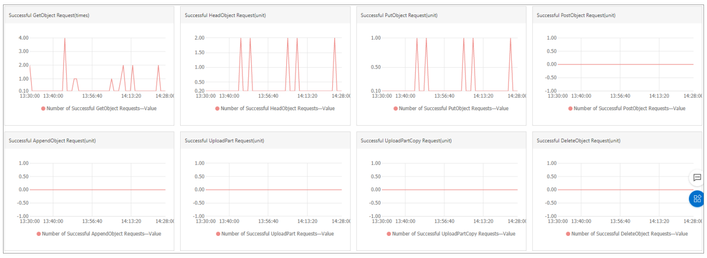
    
    GetObject请求平均延时(毫秒)
GetObject请求平均E2E延时—Value
GetObject请求平均服务器延时—Value
HeadObject请求平均延时(毫秒)
HeadObject请求平均E2E延时—Value
HeadObject请求平均服务器延时—Value
PutObject请求平均延时(毫秒)
PutObject请求平均E2E延时—Value
PutObject请求平均服务器延时—Value
PostObject请求平均延时(毫秒)
无数据
AppendObject请求平均延时(毫秒)
无数据
UploadPart请求平均延时(毫秒)
无数据
UploadPartCopy请求平均延时(毫秒)
无数据
提问

    The latency metrics indicate the average and maximum time that are consumed to process a type of request generated by calling API operations. The E2E latency metric indicates the time used to transmit a request between the client and the server, which includes the time used to process, read, and respond to the request, and network latency in this process. The server latency is the time used to process the request on the server side, which does not include the network latency during the communications between the server and the client. Therefore, when the E2E latency jumps and the server latency keeps stable, it is reasonable to determine that the E2E latency increase due to poor network conditions and exclude the possibility of OSS failure.

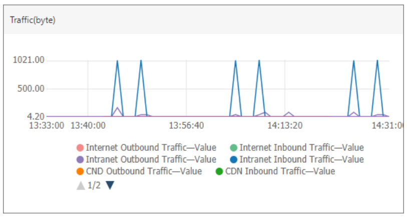
  - Maximum Latency, which includes maximum E2E latency and maximum server latency

    
    GetObject Request Maximum Latency(millisecond)
Maximum E2E Latency for GetObject Requests—Value
Getobject Request Maximum Server Latency—Value
HeadObject Request Maximum Latency(millisecond)
HeadObject Request Maximum E2E Latency—Value
HeadObject Request Maximum Server Latency—Value
PutObject Request Maximum Latency(millisecond)
PutObject Request Maximum E2E Latency—Value
PutObject Request Maximum Server Latency—Value
PostObject Request Maximum Latency(millisecond)
PostObject Request Maximum E2E Latency—Value
PostObject Request Maximum Server Latency—Value
AppendObject Request Maximum Latency(millisecond)
AppendObject Request Maximum E2E Latency—Value
AppendObject Request Maximum Server Latency—Value
UploadPart Request Maximum Latency(millisecond)
UploadPart Request Maximum E2E Latency—Value
UploadPart Request Maximum Server Latency—Value
UploadPartCopy Request Maximum Latency(millisecond)
UploadPart Request Maximum E2E Latency—Value
UploadPart Request Maximum Server Latency—Value

  - Successful Request Category

    
    Successful GetObject Request(times)
Number of Successful GetObject Requests—Value
Successful HeadObject Request(unit)
Number of Successful HeadObject Requests—Value
Successful PutObject Request(unit)
Number of Successful PutObject Requests—Value
Successful PostObject Request(unit)
Number of Successful PostObject Requests—Value
Successful AppendObject Request(unit)
Number of Successful AppendObject Requests—Value
Successful UploadPart Request(unit)
Number of Successful UploadPart Requests—Value
Successful UploadPartCopy Request(unit)
Number of Successful UploadPartCopy Requests—Value
Successful DeleteObject Request(unit)
Number of Successful DeleteObject Requests—Value

  - Traffic by User

    The Traffic by User metric indicates the traffic used by requests over the Internet, requests over the internal network, Content Delivery Network (CDN) back-to-origin, and cross-region replication (CRR). The monitoring service provides a metric of traffic by user and a metric of traffic by bucket.

  OSS monitors the preceding metrics except for traffic, by type of requests that are sent by calling the following API operations:

  - GetObject
  - HeadObject
  - PutObject
  - PostObject
  - AppendObject

- UploadPart
- UploadPartCopy

In addition, successful request category also includes the requests sent by calling the following API operations:

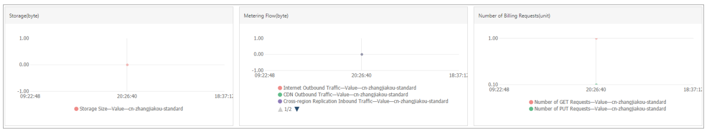
- DeleteObject
- DeleteObjects

For metrics that indicate the performance of OSS, you must focus on their abnormal fluctuations such as a spike in average latency and extended high latency for requests. You can configure alert rules for performance metrics so that a notification is sent when an alert rule is triggered.

- Monitor billable items

  The OSS monitoring service allows you to monitor the following billable items: the storage usage, the number of the PUT requests and the GET requests, outbound traffic over the Internet, which does not include CRR outbound traffic or CDN outbound traffic. The OSS monitoring service does not support alert settings for billable items and OpenAPI reading.

  The OSS monitoring service collects monitoring data by bucket on an hourly basis. You can view the chart of a specific bucket to obtain the continuous data trend of the bucket. You can predict the trend of storage usage of your business and your future storage costs based on the monitoring graph.


Storage(byte)
1.00
0.00
-1.00
09:22:48
20:26:40
18:37:1
● Storage Size—Value—cn-zhangjiakou-standard
Metering Flow(byte)
1.00
0.00
-1.00
09:22:48
20:26:40
18:37:1
● Internet Outbound Traffic—Value—cn-zhangjiakou-standard
● CDN Outbound Traffic—Value—cn-zhangjiakou-standard
● Cross-region Replication Inbound Traffic—Value—cn-zhangjiakou-standard
▲ 1/2 ▼
Number of Billing Requests(unit)
1.00
0.10
09:22:48
20:26:40
18:37:1
● Number of GET Requests—Value—cn-zhangjiakou-standard
● Number of PUT Requests—Value—cn-zhangjiakou-standard

  The OSS monitoring service provides statistics of resources usage by user and by bucket per month. The OSS resource usage of an Alibaba Cloud account or a specific bucket is calculated and updated every hour. This way, you can calculate your storage costs during the current month.

  For more information about the billable items and billing methods of OSS, see Overview.

> **Note** The statistics provided by the monitoring service may be different from actual usage in bills. You are charged based on the actual usage in the bills provided by Billing Management.

Tracking and diagnosis

- Problem diagnosis
  - Performance diagnosis

    You must determine the baseline of your application performance based on your business requirements. Note that a failed request sent from the client may be caused by one or more elements throughout the transmission link of the request, such as excessive overloads of OSS, the configurations of TCP of the client, and the traffic bottleneck from networks.

    This way, you can locate the possible issue based on the performance metric provided by the monitoring service. Then, query the logs for detailed information and diagnose the fault further.
  - Error diagnosis

    The client application receives an error message from the server when a request error occurs. The monitoring service records and displays the counts and proportions of different error requests. You can check logs of the server, the client, and the network conditions to obtain the details about a specific request. Generally, the HTTP status code, the error code, and the error message in the response indicate the possible cause of an error request.

    For more information about error codes, see Error responses.
  - Use the logging feature to diagnose

    OSS provides the logging feature for requests from users, which can track the details about requests throughout the transmission between the client application and the server.

    For more information about how to enable logging and use this feature, see Configure logging.

    For more information about the naming conventions and formats of OSS logs, see Logging.
  - Use the logs of network conditions to diagnose

    Generally, you can determine the cause of a fault based on logs of the server and the client application. However, in some cases, to determine the cause of the fault, you also need network logs to obtain information about network conditions between the server and the client, such as the traffic and data transmitted in this process. For example, an error is reported for a user request, but no logs are generated for the request on the application server. In this case, check OSS logs to determine whether the cause is on the application user client, or use network monitoring tools to check the network conditions.

    Wireshark is one of the most common tool for log monitoring and analysis. This free tool works on the level of data packages. You can use this tool to query the details about data packages transmitted over all protocols so that you can determine whether failures are caused by package loss or network connections.

    For more information about how to use Wireshark, visit Wireshark User's Guide.
- E2E tracking and diagnosis

  In a normal process, a request is sent from the client to OSS, and the OSS server processes the request and sends a response to the client. You can track the whole process between the client and the server. You can use logs of the client application, networks, and the server to locate a potential fault.

  OSS assigns every received request a unique request ID, which can be used as the identifier to distinguish logs generated for requests. In addition, by using the timestamp of a log, you can obtain information about other events occurred on the client, in networks, and on the application server while a request is processed. This helps you analyze and investigate the cause of a fault.
  - Request ID

    In different logs, the request ID of a request is contained in different fields.
    - In OSS logs, the Request ID is in the Request ID field.
    - When you track network data, such as data flows captured by Wireshark, the Request ID is included in the response as the standard HTTP header x-oss-request-id.
    - In the client application, the Request ID is automatically displayed in the logs of the client, which is implemented by using the code of the application. The latest version of OSS SDK for Java supports displaying the Request ID of a request in the response. You can use the getRequestId method to obtain the Request ID from the response. Each version of OSS SDK for various programming languages supports displaying the Request ID of an exceptional request. You can obtain the Request ID of a request by calling the getRequestId method of the OSSException operation.

- Use the timestamp of logs

  You can use the timestamp of logs to query logs. Note that an event may occur at different time points on the client and on the application server. Therefore, the timestamps of OSS logs and logs of the client for an event may be different. When you query logs of the server based on the timestamp of logs on the client, add 15 minutes to or subtract 15 minutes from the timestamp.

Troubleshooting

- Performance-related FAQs
  - High average E2E latency and low average server latency

    The following possible causes are inferred from the preceding description of the average E2E latency and the average server latency:

    - Slow response of client applications
      - The available connections and threads are limited.
        - For limited available connections, you can run relevant commands to check whether a large number of connections is in the TIME_WAIT state. If a large number of connections is in the TIME_WAIT state, you can adjust the number of CPU cores to handle this issue.
        - For limited threads, you can view whether bottlenecks exist in resources such as the client CPU, memory, and networks. If not, you can properly increase the number of concurrent threads.
        - If the issue persists, you must optimize the code of the application. For example, you can optimize the code of the application to support asynchronous access. You can also use the performance analysis feature to identify application features that are most commonly used and then optimize the client application.
      - Limited resources of CPU, memory, or bandwidth

        In this case, you must use the monitoring feature of relevant systems to determine resource bottlenecks. Then, optimize the code of the application to adjust the limits of the resources, or you can scale up the resources of the client such as increase the number of CPU cores or increase the memory.
    - Poor network conditions

      Generally, high average E2E latency is caused by temporary poor network conditions. You can use Wireshark to analyze accidental and persistent network issues such as the loss of data packages.
  - Low average E2E latency, low average server latency, and high client request latency

    For high client request latency, the high latency most likely occurs before requests arrive at the application server. Therefore, we recommend that you analyze why requests from the client do not arrive at the server.

    The following scenarios may cause high client request latency:

    - The available connections and threads are limited.
      - For limited available connections, you can run relevant commands to check whether a large number of connections is in the TIME_WAIT state. If a large number of connections is in the TIME_WAIT state, you can adjust the number of CPU cores to handle this issue.
      - For limited threads, you can view whether bottlenecks exist in the client CPU, memory, and network resources. If not, you can properly increase the number of concurrent threads.
      - If the issue persists, you must optimize the code of the application, such as access by using the asynchronous method. You can also use the performance analysis feature to identify application features that are most commonly used and then optimize the client application.
    - Multiple retries occur for request in the client. In this case, you must analyze the cause based on the retry information. You can determine whether retries occur in the client by using the following methods:
      - Check the client logs to view whether retries have occurred. Take OSS SDK for Java as an example. You can query log prompts of the warn- or info- level: If similar logs are recorded, retries may occur in the client or in the server.

        ```
        [Server]Unable to execute HTTP request:
          or
          [Client]Unable to execute HTTP request:
        ```
      - Take OSS SDK for Java as an example. You can query the following log if the level of the client log is debug. If similar logs are recorded, retries must have occurred.

        ```
        Retrying on
        ```

    If the client has no faults, consider potential issues of networks such as the loss of data packages. You can use tools such as Wireshark to analyze the cause of network issues.
  - High average server latency

    For high average server latency of downloads and uploads, consider the following possible causes:

    - A large number of clients frequently access an object.

      In this case, you can view OSS logs to determine whether an object or a group of objects is frequently accessed.

      For the scenario of downloads, we recommend that you activate Content Delivery Network (CDN) for the bucket to improve the performance and reduce the generated traffic. For the scenario of uploads, we recommend that you modify the access control list (ACL) of the bucket or the object so that users cannot write data to the bucket or the object if this does not affect your business requirements.
    - Internal issues of OSS

      For internal issues of OSS, they may not be solved by optimizing the code of the application. In this case, contact technical support to provide the client logs or the Request ID of the failed request in OSS logs.
- Errors of the application server

  If the errors of the server increase, consider the following possible causes:

  - Temporary increase

    In this case, you must adjust the retry policy of the client application and use proper concession mechanisms such as exponential backoff. This way, you can prevent service unavailability caused by the optimization, upgrade, and data migration for OSS load balance. In addition, the pressure of your business at peak times can be reduced.

    - Permanent increase

      If server errors remain high, contact technical support to provide the client logs or the Request ID of the failed request in OSS logs.

- Network errors

  A network error occurs when the server fails to respond to a request because of disconnection from the client or networks while the server processes the request. In this case, the HTTP status code 499 is recorded for the request. The status code 499 has the following possible causes:

    - Before the server receives requests to read and write data, the server checks whether the connection is available. If not, the HTTP status code 499 is recorded for the request.
    - The HTTP status code 499 is recorded for a request when the server is processing the request but the client disables the connection.

  A network error occurs when the client cancels the request or loses the connection during a request process. If the client cancels the request, you must check the code of the application and obtain why and when the client disconnects with OSS. If the client loses the connection, you can use tools such as Wireshark to analyze possible causes.

- Client errors

    - Increase of client authorization error requests

      When the client authorization error request increases, or the client application receives a large amount of the HTTP status code 403, consider the following possible causes:

        - Invalid bucket domain name

            - If users use the third-level domain or the second-level domain to access the bucket, the region contained in the domain name may be different from the region in which the bucket is located. For example, the accessed bucket is located in the China (Hangzhou) region, but the accessed domain name is Bucket.oss-cn-shanghai.aliyuncs.com. In this case, you must confirm the region in which the bucket is located and correct the accessed domain name.
            - If you have enabled CDN, the origin mapped to CDN may be the wrong domain name of the bucket. In this case, check whether the origin is the third-level domain of the bucket.
            - If users use the client of JavaScript and a 403 error is returned, check whether CORS is configured for the web browser that users use to access the bucket. In this case, check the CORS settings of the bucket and correct the settings so that users can use the web browser to access the bucket. For more information about how to configure cross-origin resource sharing (CORS), see Configure CORS.

        - Access control

            - If you use the AccessKey pair of an Alibaba Cloud account to access the bucket, check the validity of your AccessKey pair.
            - If you use the AccessKey pair of a RAM user to access the bucket, check the validity or the authorization of the AccessKey pair.
            - If you use a token generated by Security Token Service (STS), check whether the temporary token expires. If the token has expired, apply for a new token.
            - If the access control list (ACL) is configured for the accessed bucket or object, check whether users are allowed to perform specific operations based on the ACL settings.

        - Expired URL

          If a 403 error occurs when the third party accesses the bucket by using a signed URL, the most possible cause is that the signed URL has expired.

        - A 403 error may occur when RAM users log on to OSS tools such as ossftp, ossbrowser, and the OSS console. In this case, check whether you enter the correct AccessKey pair or whether you have the permission to call the GetService operation if your account is a RAM user.

    - Increase of 404 errors for client requests

      A 404 error for a client request indicates that the data that users access does not exist. When the number of 404 errors for client requests increases, consider the following possible causes:

        - The business logic of the application. For example, an application calls the doesObjectExist method provided by OSS SDK for Java to check whether an object exists before further actions. If the object does not exist, the value of false is returned to the client and a 404 error message is generated on the server. Therefore, in the business scenario, a 404 status code does not indicate an error.
        - The accessed object is deleted by the client application or other processes. In this case, query OSS logs for the accessed object.
        - Repeat delete operations caused by network failure. For example, the client initiates an operation to delete an object. The request arrives at the server, and the object is deleted. However, the response does not arrive at the client due to network failure. As a result, the client sends another request to delete the object, and a 404 error occurs. In this case, you can query and view the client logs and OSS logs to determine the cause of the 404 error.
            - Query the client logs and check whether a repeated request is sent from the client.
            - Query OSS logs. Then, check whether two delete operations are initiated on the object and the HTTP status code of the first operation is 2xx.

    - Low valid request proportion and high client error requests

      Valid request proportion is the proportion of successful requests whose responses are the HTTP status code 2xx or 3xx to total requests. Requests whose responses are the HTTP status code 4xx or 5xx are counted as failed requests and lower the valid request proportion. Client other error requests by user refer to error requests other than server error requests, network error requests, client authorization error requests, resource not found error requests and client timeout error requests. The responses to the preceding error requests are respectively 5xx, 499, 403, 404, 408, or 400 whose corresponding OSS error code is RequestTimeout.

      You can query OSS logs to determine the error type. Then, refer to OSS error codes and solve the errors by modifying the code of the application. For more information, see Error responses.

- Exceptional increase in storage usage

  When your storage usage dramatically increases, the cause may be the cleaning operation failure. Troubleshoot in the following aspects:

    - If the client application uses specific processes to perform regular cleaning operations to release storage, find out the cause in the following steps:

      a. Check whether the valid request proportion decrease because failed cleaning operations lower the valid request proportion.

      b. Locate and determine the specific cause of the decrease in valid request proportion and the type of error requests. Then, you can obtain the details about errors from the client logs.

    - The client application clears your bucket storage by configuring lifecycle rules. For cleaning operations triggered by lifecycle rules, you must use the OSS console or call API operations to check whether the lifecycle rules are correctly configured. If not, you can modify the lifecycle rules in the OSS console. To check whether the lifecycle rules were modified, query OSS logs. If the lifecycle rules are correctly configured but do not take effect, contact OSS technical support for help.

- Other storage service issues

  If the troubleshooting section in this topic does not solve your storage service issues, use the following methods to diagnose and troubleshoot your issues:

i. View the monitoring service of OSS in the Cloud Monitor console and check whether the baseline of metrics has been changed. You may determine whether the issue is temporary or permanent and which storage operations are affected by this issue.

ii. Query OSS logs to obtain all errors that occur at the same time based on the monitoring information which may help you locate and solve the issue.

iii. If the logs of the OSS server cannot provide sufficient information for troubleshooting, use the client logs to investigate the client application or network tools such as Wireshark to investigate network failures.

11.7. FAQ

This topic provides answers to frequently asked questions about OSS monitoring service.

OSS and Cloud Monitor are two Alibaba Cloud services. OSS sends data to Cloud Monitor for analysis and processing. Statistics on the storage and traffic usage displayed in the OSS console are collected by Cloud Monitor.

Latencies of two to three hours exist when OSS sends data to Cloud Monitor. The interval at which OSS sends data cannot exceed five minutes. If the interval exceeds five minutes, Cloud Monitor rejects the data and the data cannot be sent again. Therefore, we recommend that you do not calculate the fees based on the data from Cloud Monitor. To check fees, we recommend that you click here to submit a ticket.

What do I do if the status of the alert rule displays Insufficient Data?

Solution: You can view data on the **Monitoring Service Overview** tab of the **Users** tab. The alert rule displays Insufficient Data if no data is generated.

What do I do if the data of object upload or download is not updated in real time in the Cloud Monitor console?

Cause: The latest data that is visible in the Cloud Monitor console is detected by sending requests, during which latencies exist.

Solution: Perform the following operations to troubleshoot the problem:

1. Check whether latencies of access to the client exist.
2. Capture packets to analyze data if latencies exist when users access the corresponding bucket.
3. Analyze access logs to confirm whether latencies exist.

What do I do if latencies of access from the monitoring system of a company to OSS exist?

A company sets up a monitoring system to monitor OSS data. They can perform the following operations to troubleshoot the problem when long latencies of access to OSS exist:

1. Check whether the network works properly. You can ping the IP addresses of other networks to test whether latencies exist.
2. Create an ECS server that is located in the same region as the OSS bucket. Start access from the ECS server to the bucket to test whether latencies exist.
3. Send the ID of the request with latencies to after-sales technical support. Check whether access latencies exist.
4. Capture packets to obtain upload data. Use the following parameters to analyze packets:

```
tcpdump -i <Outbound port name> -s0 (Public IP address of the host and OSS endpoint) -w result.pcap
```

What do I do if the valid request rate is low?

Problem description: Cloud Monitor displays a message, indicating that OSS bucket p2xxx of 135114002 has a valid request rate of 30.51%, which is lower than 90%, and the error lasts zero minutes.

Solution: The abnormal request rate is calculated based on the formula: Abnormal request rate = Count of 2xx and 3xx responses/Total count of responses. You can view the ratios of abnormal responses including 2xx and 3xx responses, and confirm whether the increase of abnormal HTTP status codes leads to the decrease of valid request rates. You can also enable OSS real-time analysis.

What do I do if Cloud Monitor returns HTTP status code 404?

Problem description: Cloud Monitor displays a message, indicating that the number of error requests to resources in OSS bucket ***-ali of 197*****745 that do not exist is 30, and the error lasts five minutes and is fixed at 11:45.

Cause: The requested resources do not exist. The response is normal.

What do I do if Cloud Monitor returns NoSuchWebSiteConfigration?

Cause: The loaded feature configurations do not exist when the client requests OSS data. As a result, HTTP status code 404 is returned. HTTP status code 200 is returned if the feature is configured. These cases are normal.

Why does the OSS console display no statistics on API operations?

Cause: Statistics on API operations are displayed the next day. For example, statistics on API operations on October 12 are displayed on October 13.in OSS, which is normal.

What do I do if data is inaccurate when I check bills generated for OSS monitoring service?

Latencies of two to three hours exist when OSS sends data to Cloud Monitor. The intervals at which OSS sends data cannot exceed five minutes. If the interval exceeds five minutes, Cloud Monitor rejects the data and the data cannot be sent again. Solution: We recommend that you do not use the data from Cloud Monitor when you check bills. We recommend that you use one of the following methods to check bills:

- Enable OSS logging and check bills based on statistics on OSS logs.
- Enable OSS real-time log analysis to import, analyze, and process logs and view the results.

What do I do if Cloud Monitor shows that the valid request rate during a certain period is decreased to 0 but the data in OSS logs and the OSS console is reasonable?

Cause: Cloud Monitor calculates the valid request rate based on the formula: 100% - Count of 2xx and 3xx responses/Total count of responses. Solution: Check whether exceptions exist in the OSS console or OSS logs.

Cause: The time OSS uses to send the entire cluster logs to Cloud Monitor exceeds the time window predetermined by Cloud Monitor. However, Cloud Monitor rejects data that fails to be received for the first time.

12.Event notifications

12.1. Overview

You can configure event notification rules for objects that you want to monitor in the Object Storage Service (OSS) console. If the events that you specified in the rules occur on these objects, you can be immediately notified.

Prerequisites

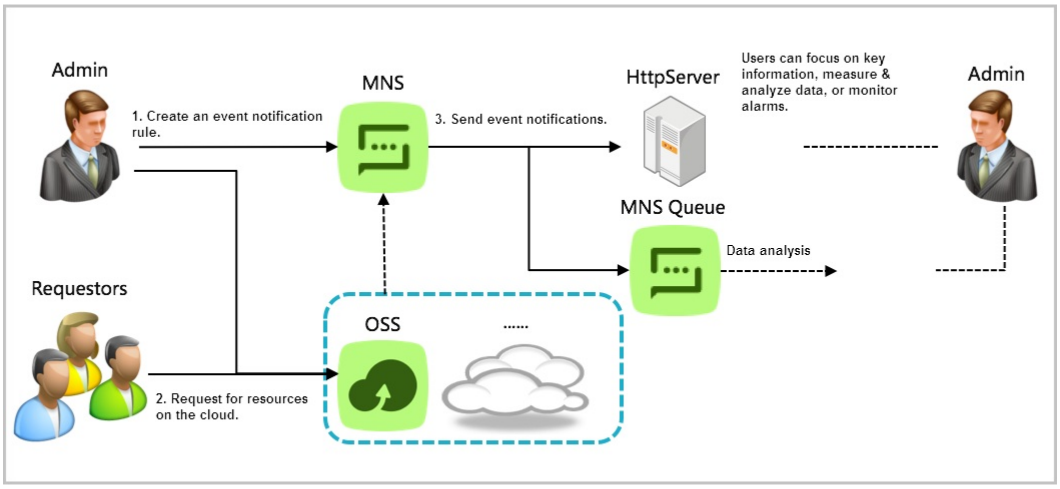
Message Service (MNS) is activated. You can activate MNS on the MNS product page.

Usage notes

- You are charged by MNS when you use the event notification feature. For more information about the pricing, see Pricing.
- The event notification feature is not supported in the following regions: China (Heyuan), China (Guangzhou), China (Hohhot), China (Ulanqab), UAE (Dubai), and Malaysia (Kuala Lumpur).
- You can configure up to 10 event notification rules in a region.
- Notifications are not sent for TS and M3U8 objects that are generated by ingesting streams over Real-Time Messaging Protocol (RTMP). For more information about RTMP-based stream ingest, see Overview.

Description

If an operation performed on your OSS resources triggers an event notification rule that you configure, MNS sends a notification about the operation to the specified HTTP server or MNS queue. The following figure shows the event notification process in detail.


Admin
1. Create an event notification rule.
MNS
3. Send event notifications.
HttpServer
Users can focus on key information, measure & analyze data, or monitor alarms.
Admin
MNS Queue
Data analysis
Requestors
2. Request for resources on the cloud.
OSS
......

Events

Event	Description
PutObject	An object is created or overwritten by using simple upload.
PostObject	An object is created or overwritten by using form upload.
CopyObject	An object is created or overwritten by copying an object.
InitiateMultipartUpload	A multipart upload task is initiated.
UploadPart	An object is created or overwritten by uploading parts.
UploadPartCopy	An object is created or overwritten by using multipart copy.
CompleteMultipartUpload	A multipart upload task is completed.
AppendObject	An object is created or appended by using append upload.
GetObject	An object is obtained by using simple download.
DeleteObject	A single object is deleted.
DeleteObjects	Multiple objects are deleted.
ObjectReplication:ObjectCreated	An object is created by using cross-region replication (CRR).
ObjectReplication:ObjectRemoved	An object is deleted by using CRR.
ObjectReplication:ObjectModified	An object is overwritten by using CRR.
ObjectCreatedGroup	An object is created or overwritten.
ObjectDownloadedGroup	An object is downloaded.
ObjectRemovedGroup	An object is deleted.

Notice The ObjectCreatedGroup, ObjectDownloadedGroup, and ObjectRemovedGroup events are available only in the China (Hong Kong), US (Silicon Valley), US (Virginia), Germany (Frankfurt), Australia (Sydney), Singapore (Singapore), and UK (London) regions.

Notifications

The content of OSS-based event notifications is Base64-encoded. After the content is decoded, the content is in the JSON format. The following code provides an example on the decoded content:

```
{"events": [
      {
        "eventName": "",  // The event name.
        "eventSource": "", // The resources that trigger an event notification. Valid value: acs:oss.
        "eventTime": "", // The time when the event occurred. The returned data is a timestamp that complies with ISO 8601.
        "eventVersion": "", // The version of an event notification. The current version is 1.0.
        "oss": {
            "bucket": {
                "arn": "", // The Alibaba Cloud Resource Name (ARN) of the bucket, which is in the following format: acs:oss:region:uid:bucketname.
                "name": "", // The name of the bucket for which the event notification rule is configured.
                "ownerIdentity": "" // The owner of the bucket.
            },
            "object": {
                "deltaSize": "", // The size difference between the new object and the original object. If you create an object, the value of this pa
rameter indicates the size of the object. If you overwrite an object, the value indicates the difference between the sizes of the new object and the
original object. Therefore, the value may be negative.
                "eTag": "", // The ETag value of the object.
                "key": "", // The name of the object.
                "position": "", // This parameter applies only to the ObjectCreated:AppendObject event and indicates the position from which the Appe
ndObject operation starts. The first AppendObject operation on an object starts from byte 0 of the object.
                "readFrom": "", // This parameter applies only to the ObjectDownloaded:GetObject event and indicates the position from which the GetO
bject operation starts. If the GetObject operation is not performed by range, the value of this parameter is 0. Otherwise, the value refers to the po
sition of the byte from which the GetObject operation starts.
                "readTo": "", // This parameter applies only to the ObjectDownloaded:GetObject event and indicates the position at which the last Get
Object operation ends. If the GetObject operation is not performed by range, the value of this parameter is the object size. Otherwise, the value ref
ers to the position of the byte next to the end byte of the GetObject operation.
                "size": "" // The size of the object.
                },
        "ossSchemaVersion": "", // The version of the schema. The current value is 1.0.
        "ruleId": "GetObject", // The ID of the rule that matches the event.
        "region": "", // The region in which the bucket is located.
        "requestParameters": {
            "sourceIPAddress": "" // The source IP address from which the request is sent.
            },
        "responseElements": {
            "requestId": "" // The ID of the request.
            },
        "userIdentity": {
            "principalId": "" // The UID of the requester.
            },
        "xVars": {  // Custom parameters for OSS callback.
            "x:callback-var1":"value1",
            "x:vallback-var2":"value2"
            }
        }
     }
  ]
}
```

The following code provides an example on the content of a notification:

```
{"events": [
      {
        "eventName": "ObjectDownloaded:GetObject",
        "eventSource": "acs:oss",
        "eventTime": "2016-07-01T11:17:30.000Z",
        "eventVersion": "1.0",
        "oss": {
            "bucket": {
                "arn": "acs:oss:cn-shenzhen:114895646818****:event-notification-test-shenzhen",
                "name": "event-notification-test-shenzhen",
                "ownerIdentity": "114895646818****"},
            "object": {
                "deltaSize": 0,
                "eTag": "0CC175B9C0F1B6468E1199E269772661",
                "key": "test",
                "readFrom": 0,
                "readTo": 1,
                "size": 1
            },
        "ossSchemaVersion": "1.0",
        "ruleId": "GetObjectRule",
        "region": "cn-shenzhen",
        "requestParameters": {
            "sourceIPAddress": "198.51.100.1"
            },
        "responseElements": {
            "requestId": "5FF16B65F05BC932307A3C3C"
            },
        "userIdentity": {
            "principalId": "114895646818****"
            },
        "xVars": {
            "x:callback-var1":"value1",
            "x:vallback-var2":"value2"
            }
        }
     }
  ]
}
```

Use the OSS console

1. Log on to the OSS console.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure event notification rules.
3. In the left-side navigation pane, choose **Basic Settings > Event Notification**.
4. In the Event Notification section, click **Configure**. On the page that appears, click **Create Rule**.
5. In the **Create Rule** panel, configure the rule parameters that are described in the following table.

Parameter	Description
Rule Name	Specify the name of the event notification rule. The name of each event notification rule that is created by using the same Alibaba Cloud account must be unique in the same region. The name of an event notification rule must start with a letter and can contain only letters, digits, and hyphens (-). The name cannot exceed 85 characters in length.
Events	Select one or more events that can trigger the event notification rule from the drop-down list. For example, if you want to receive a notification when a specific object is created or overwritten by copying an object, select CopyObject. You can configure an event notification rule for a specific object and specify multiple types of events that can trigger the rule. You can also configure multiple event notification rules for an object. When you configure multiple event notification rules, take note of the following items: ○ If the multiple event notification rules apply to the same object, the values of this parameter in these rules must be different. For example, if you select CopyObject for Events when you create an event notification rule for objects whose names contain the `images` prefix, CopyObject cannot be selected for Events when you create another event notification rule for objects whose names contain the same prefix. ○ If the multiple event notification rules apply to different objects, the values of this parameter in these rules can be the same. For example, if you select PutObject for Events when you create an event notification rule for objects whose names contain the `images` prefix and the `.png` suffix, you can select PutObject or DeleteObject for Events when you create another event notification rule for objects whose names contain the `log` prefix and the `.jpg` suffix. Notice If you do not specify the version ID when you delete an object from a versioned bucket, the DeleteObject or DeleteObjects event notification is not triggered. In this case, no version of the object is deleted. The current version of the object is converted into a previous version and a delete marker is added to the object. For more information about the object operations that correspond to the event types, see Events.

Parameter	Description
Resource Description	Specify the objects to which the event notification rule applies. ○ Select **Full Name** to apply the rule to an object whose name matches the specified name. ■ To create a rule that applies to an object named exampleobject.txt in the root directory of the bucket, enter *exampleobject.txt*. ■ To create a rule that applies to an object named myphoto.jpg in the destdir directory within the root directory of the bucket, enter *destdir/myphoto.jpg*. ○ Select **Prefix and Suffix** to apply the rule to objects whose names contain the specified prefix and suffix. ■ To create a rule that applies to all objects in the bucket, leave Prefix and Suffix empty. ■ To create a rule that applies to all objects in the examplefolder directory within the root directory of the bucket, set Prefix to *examplefolder/* and leave Suffix empty. ■ To create a rule that applies to all JPG objects in the bucket, leave Prefix empty and set Suffix to *.jpg*. ■ To create a rule that applies to all MP3 objects in the *examplefolder* directory within the root directory of the bucket, set Prefix to *examplefolder/* and Suffix to *.mp3*. To create a Resource Description entry, click **Add**. You can create up to five **Resource Description** entries.
Endpoint	Specify the endpoint to which notifications are sent. Valid values: **HTTP** and **Queue**. ○ **HTTP**: Enter the address of the HTTP endpoint to which notifications are sent. Example: `http://198.51.100.1:8080` . For more information about how to enable an HTTP endpoint, see Manage topics and HttpEndpoint. ○ **Queue**: Enter the name of an MNS queue. For more information about how to create a queue, see Create a queue. To create an endpoint, click **Add**. You can create up to five **endpoints**.

6. Click **OK**.

   After you configure the event notification rule, the rule takes effect after approximately 10 minutes.

Troubleshooting

- Issue description: A bucket has an event notification rule configured for DeleteObject and DeleteObjects events. However, no notifications are sent when delete operations are performed on objects in the bucket.
- Cause: The bucket has versioning enabled. However, no object version IDs are specified in the delete operation. If you do not specify an object version ID when you perform a delete operation, the current version is converted to a previous version and a delete marker is added to the object. Therefore, the delete operation does not trigger the event notification rule configured for DeleteObject and DeleteObjects events.

References

You can configure event notification rules for objects that you want to monitor. If the events that you specified in the rules occur on these objects, you are immediately notified from the HTTP servers or MNS queues that you specified in the rules. For more information about the procedure, see Tutorial: Use MNS to send notifications for OSS events.

12.2. Tutorial: Use MNS to send notifications for OSS events

You can configure event notification rules in the Object Storage Service (OSS) console for objects that you want to monitor. If the events that you specified in the rules occur on these objects, you can immediately receive a notification from the HTTP servers or Message Service (MNS) queues that you specify in the rules.

Scenario

You create a bucket named srcbucket in the China (Hangzhou) region for your company, and then create two directories named log/ and destdir/ in srcbucket. The log/ directory is used to store log objects that are continuously generated. The names of the log objects contain the log/ prefix and indicate the date when the objects are generated. Examples: log/date1.txt, log/date2.txt, and log/date3.txt. The destdir/ directory is used to store image objects of user cases, which are collected on a weekly basis. The names of the image objects contain the destdir/ prefix. Examples: destdir/photo1.jpg and destdir/photo2.jpg. The following structure shows the directories and objects in srcbucket:

```
srcbucket
    └── log/
       ├── date1.txt
       ├── date2.txt
       ├── date3.txt
       ├── ...
    └── destdir/
       ├── photo1.jpg
       ├── photo2.jpg
       ├── ...
```

A bucket that belongs to your parent company named destbucket is located in the UK (London) region. You want to synchronize the log and image objects and the operations performed on these objects, such as creation, modification, and deletion, from srcbucket to destbucket in real time. In addition, you want the employees of your company and parent company to be notified of any changes to objects in the log/ and destdir/ directories of srcbucket and destbucket.

In this case, you can configure a cross-region replication (CRR) rule for srcbucket and event notification rules for srcbucket and destbucket.

Step 1: Create MNS queues

1. Log on to the MNS console.
2. In the left-side navigation pane, click **Queues**.
3. In the top navigation bar, select the **China (Hangzhou)** region.
4. On the **Queues** page, click **Create Queue**.
5. Click Create Queue. In the **Create Queue** panel, set Name to *myqueue1* and keep the default settings of other parameters.
6. Click **OK**.

7. Repeat the preceding steps to create a queue named *myqueue2* in the **UK (London)** region to receive notifications sent for the destination bucket destbucket.

Step 2: Configure a CRR rule for srcbucket

1.
2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click srcbucket.
3. In the left-side navigation pane, choose **Redundancy for Fault Tolerance > Cross-Region Replication**.
4. In the **Cross-Region Replication** section, click **Configure**.
5. Click **Cross-Region Replication**. In the **Cross-Region Replication** panel, configure the parameters described in the following table.

Parameter	Description
Destination Region	Select **UK (London)**.
Destination Bucket	Select **destbucket**.
Acceleration Type	To accelerate data transmission between srcbucket in the China (Hangzhou) region and destbucket in the UK (London) region during CRR, we recommend that you select **Transfer Acceleration**. If you enable transfer acceleration, you are charged for this feature. For information about the billing methods, see Transfer acceleration fees.
Applied To	Select **Files with Specified Prefix**, and then specify the following two prefixes: *destdir/* and *log/*.
Operations	Select **Add/Delete/Change**.
Replicate Historical Data	Select **Yes**.

6. Click **OK**.

Step 3: Configure event notification rules for srcbucket and destbucket

Notice The CRR rule configured in Step 2 specifies that historical data in srcbucket is synchronized to destbucket. If the amount of the historical data is large, a large number of notifications are sent after you configure event notification rules for srcbucket and destbucket. If you do not want event notification rules to be triggered when historical data is synchronized, we recommend that you configure event notification rules for srcbucket and destbucket after all historical data is synchronized.

1. Configure an event notification rule for srcbucket.
   i. Log on to the OSS console.
   ii. In the left-side navigation pane, click **Buckets**. On the Buckets page, click srcbucket.
   iii. In the left-side navigation pane, choose **Basic Settings > Event Notification**.
   iv. In the Event Notification section, click **Configure**. On the page that appears, click **Create Rule**.
   v. In the **Create Rule** panel, configure the parameters described in the following table. Then, click **OK**.

Parameter	Description
Rule Name	Set the rule name to *notification1*.
Events	Select **PutObject**, **CopyObject**, **DeleteObject**, and **DeleteObjects**.
Resource Description	Select **Prefix and Suffix**. Then, specify the following prefixes: *log/* and *destdir/*.
Endpoint	Select **Queue**. Then, enter *myqueue1* that is created in Step 1.

2. Configure an event notification rule for destbucket.
   i. In the left-side navigation pane, click **Buckets**. On the Buckets page, click destbucket.
   ii. In the **Create Rule** panel, configure the parameters described in the following table. Then, click **OK**.

Parameter	Description
Rule Name	Set the rule name to *notification2*.
Events	Select **ObjectReplication:ObjectCreated**, **ObjectReplication:ObjectRemoved**, and **ObjectReplication:ObjectModified**.
Resource Description	Select **Prefix and Suffix**. Then, specify the following prefixes: *log/* and *destdir/*.
Endpoint	Select **Queue**. Then, enter *myqueue2* that is created in Step 1.

After you configure the event notification rules, the rules take effect in about 10 minutes.

Step 4: Receive notifications

When the event notification rule that you configure for srcbucket or destbucket is triggered, MNS automatically creates a topic whose name is in the following format: `mns-en-topics-[Product]-[RuleName]-[Timestamp]`. Example: `mns-en-topics-oss-notification1-96828124450125`. To receive notifications, you must create a subscription to the topic and specify an endpoint in the subscription rule.

1. Create subscriptions to the topics.
   i. Log on to the MNS console.
   ii. In the left-side navigation pane, click **Topics**.
   iii. In the top navigation bar, select the **China (Hangzhou)** region.
   iv. Click **View Subscriptions** in the Actions column that corresponds to the topic that is automatically created by MNS.
   v. In the View Subscriptions to Topic panel, click **Subscriptions**. On the page that appears, click **Create Subscription**.

vi. In the **Create Subscription** panel, configure parameters to create a subscription to the topic that is created by MNS for srcbucket. Set **Name** to *mysubscription1*, select **Queue** for **Push Type**, and set **Receiver Endpoint** to *myqueue1*. Keep the default settings of other parameters.

vii. Click **OK**.

viii. In the **UK (London)** region, repeat the preceding steps to create a subscription named *mysubscription2* to the topic that is created by MNS for destbucket. Select **Queue** for Push Type, and set Receiver Endpoint to *myqueue2*.

2. Receive notifications.

   i. In the left-side navigation pane, click **Queues**.

   ii. Choose **More > Send Messages** in the Actions column that corresponds to *myqueue1*.

   iii. In the **Receive Message** section of the Quick Experience page, click **Receive Message** in the upper-right corner.

   This notifies you when objects in the log/ and destdir/ directories of srcbucket are created, modified, or deleted.

   iv. Repeat the preceding steps to receive notifications from *myqueue2* that is created for destbucket. This notifies you when objects in destbucket are created, modified, or deleted based on the CRR rule.

If you no longer need to receive notifications sent based on an event notification rule, delete the event notification rule. However, the topic that was automatically created by MNS based on the deleted event notification rule is not deleted. To avoid unnecessary charges, delete topics that you no longer use.

13.DLA

13.1. Overview of the OSS-HDFS service

OSS-HDFS (JindoFS service) is a cloud-native data lake storage service. OSS-HDFS is built on unified metadata management capabilities and is fully compatible with Hadoop Distributed File System (HDFS) API operations. The Portable Operating System Interface (POSIX) is supported in OSS-HDFS. This helps OSS-HDFS handle data lake computing scenarios in the big data and AI fields.

Benefits

You can use the OSS-HDFS service without the need to modify the Hadoop and Spark applications. It can be easily configured to access and manage data in a similar way as in HDFS. In addition, you can take advantage of Object Storage Service (OSS) characteristics such as unlimited storage space, elastic scalability, and high security, reliability, and availability.

As the foundation of cloud-native data lake, OSS-HDFS can analyze exabytes of data, manage hundreds of millions of objects, and achieve terabytes of throughput. OSS-HDFS provides flat and hierarchical namespace features to meet the requirements for big data storage. The hierarchical namespace feature allows you to manage objects in a hierarchical directory structure. In addition, unified metadata management enables automatic switchovers between OSS and HDFS. Users of Hadoop can access their objects in OSS-HDFS without the need to copy and convert the format of the objects. This improves efficiency and reduces maintenance costs.

Characteristics

The OSS-HDFS service provides the following characteristics:

- The OSS-HDFS service is fully compatible with HDFS API operations and supports directory-level operations. JindoSDK allows Apache Hadoop-based computing and analysis applications, such as MapReduce, Hive, Spark, and Flink, to access HDFS. This way, you can access and manage your data in OSS-HDFS the same way as in HDFS. For more information, see OSS-HDFS服务快速入门.

Compatibility with HDFS

- OSS-HDFS supports POSIX by using JindoFuse. This allows you to mount objects in OSS-HDFS to the local file system so that you can manage objects in OSS-HDFS the same way as in the local file system. For more information, see Use JindoFuse to access the OSS-HDFS service.

Support for POSIX

- Self-managed Hadoop clusters are restricted by physical resources and difficult to achieve resource scalability. For example, NameNode hits a bottleneck when the number of nodes of a Hadoop cluster exceeds several hundred, and the files reaches about 400 million. As the size of metadata increases, the queries per second (QPS) of the cluster decrease.

  The OSS-HDFS service is designed to support multiple tenants and storage of a huge amount of data. You can scale out the capability of metadata management. In addition, OSS-HDFS supports high concurrency and high throughput with low latency. OSS-HDFS performance remains stable and highly available even if the number of objects exceeds one billion. OSS-HDFS provides unified metadata management capabilities to handle ultra-large files and supports multiple hierarchy policies. This helps OSS-HDFS make better use of system resources, lower costs, and adapt to business workload.

High performance, high scalability, and low costs

- OSS-HDFS stores data in OSS, which is the core infrastructure of data storage for Alibaba Cloud. OSS is a proven success in smoothly handling peak traffic during the Double 11 shopping festival and provides high availability and high reliability. OSS provides the following features:
  - 99.995% service availability (designed for).
  - Data durability of at least 99.9999999999% (twelve 9's) (designed for).
  - Automatic scalability without services interruptions.
  - Automatic backup for redundancy.

Data durability and service availability

Scenarios

OSS-HDFS works well in the big data and AI fields. The service is used in the following scenarios:

- OSS-HDFS provides native support for files, directories, and related operations. In addition, OSS-HDFS supports atomic operations on directories and rename operations within milliseconds. This makes OSS-HDFS applicable to Hive and Spark for offline data warehousing. When you use the extract, transform, and load (ETL) feature to handle streaming data, OSS-HDFS delivers better performance than OSS buckets.

Hive and Spark for offline data warehousing

- OSS-HDFS provides basic file-related operations, such as append, truncate, flush, and pwrite. OSS-HDFS supports POSIX by using JindoFuse. This way, when you use ClickHouse for online analytical processing (OLAP), local disks can be replaced to achieve the decoupling of storage from computing. In addition, the caching system helps speed up operations and improve performance at low costs.

OLAP

- OSS-HDFS provides basic file-related operations, such as append, truncate, flush, and pwrite. OSS-HDFS supports POSIX by using JindoFuse. Therefore, OSS-HDFS can seamlessly work with AI programs and existing training and reasoning programs in Python.

AI training and reasoning

- OSS-HDFS provides native support for files, flush operations, directories, and related operations. You can use OSS-HDFS instead of HDFS to decouple storage from computing for HBase. Compared with the storage for HBase by using OSS Standard buckets, OSS-HDFS can store WAL logs for HBase without dependence on HDFS. This streamlines the architecture.

Decoupling of storage from computing for HBase

- OSS-HDFS supports flush and truncate operations. You can use OSS-HDFS instead of HDFS to store sinks and checkpoints in real-time computing scenarios of Flink.

Real-time computing

- As a novel cloud-native data lake storage service, OSS-HDFS allows HDFS to migrate data to the cloud and optimize the experience of HDFS users. This way, OSS-HDFS provides scalability and cost-effective storage services. You can use Jindo DistCp, which is a data copy tool provided by Alibaba Cloud, to migrate data from HDFS to OSS-HDFS. During data migration, HDFS checksum can be used to verify data integrity.

Data migration

Features

The following table describes features of the OSS-HDFS service in different scenarios.

Scenario	Feature
Work with Hive and Spark for data warehousing	Native support for files, directories, and related operations
	Add permissions on files and directories
	Atomic operations on directories and rename operations within milliseconds
	Set time by using setTimes
	Extended attributes
	Access control list (ACL)
	Accelerate local read caching
Replace HDFS	Snapshots
	File-related operations such as flush, sync, truncate, and append
	Checksum verification
	Automatic clean-up of the HDFS recycle bin
Support for POSIX	Random writes to files
	File-related operations such as truncate, append, and flush