Object Storage Service

User Guide

Issue 01

Date 2025-06-20

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

Notice

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

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

Contents

1 Service Overview

1.1 About OBS

OBS Overview

Object Storage Service (OBS) is a scalable service that provides secure, reliable, and cost-effective cloud storage for massive amounts of data.

OBS provides unlimited storage capacity for objects of any format, catering to the needs of common users, websites, enterprises, and developers. There is no limitation on the storage capacity of the entire OBS system or of a single bucket, and any number of objects can be stored. As a web service, OBS supports APIs over Hypertext Transfer Protocol (HTTP) and Hypertext Transfer Protocol Secure (HTTPS). You can use OBS Console or OBS tools to access and manage data stored in OBS anytime, anywhere. With OBS APIs, you can easily manage data stored in OBS and develop upper-layer applications.

Product Architecture

OBS basically consists of **buckets** and **objects**.

A bucket is a container for storing objects in OBS. Each bucket is specific to a region and has specific storage class and access permissions. A bucket is accessible through its **access domain name** over the Internet.

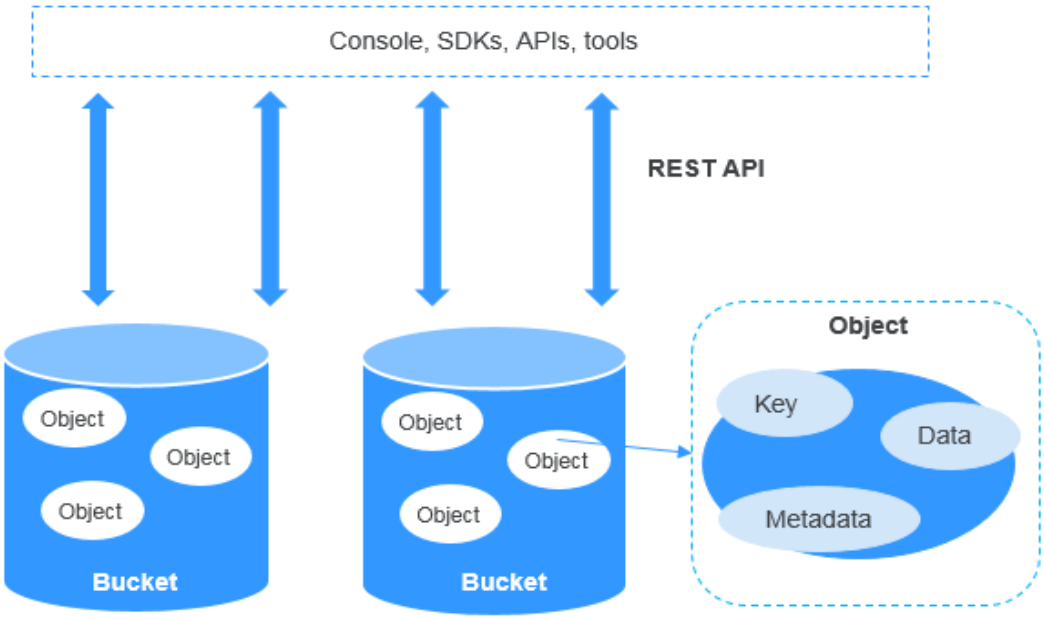
An object is the fundamental storage unit in OBS. An object consists of the following:

- A key that specifies the name of an object. An object key is a UTF-8 string up to 1,024 characters long. Each object within a bucket is uniquely identified by a key.
- Metadata that describes an object. The metadata is a set of key-value pairs that are assigned to objects stored in OBS. There are two types of metadata: system-defined metadata and custom metadata.
  - System-defined metadata is automatically assigned by OBS. Such metadata includes Date, Content-Length, Last-Modified, ETag, and more.

  - You can specify custom metadata to describe the object when you upload an object to OBS.
- Data that refers to the content of an object.

By means of secondary development based on OBS REST APIs, OBS Console and a variety of tools are provided for you to use OBS. You can also use OBS APIs to develop applications customized for your business needs.

Figure 1-1 Product architecture



Storage Classes

OBS offers the storage classes below to meet your requirements for storage performance and cost:

- Standard: The Standard storage class features low latency and high throughput. It is therefore good for storing frequently (multiple times per month) accessed files or small files (less than 1 MB). Its application scenarios include big data analytics, mobile apps, hot videos, and social apps.
- Warm: The Warm storage class is for storing data that is infrequently (less than 12 times per year) accessed, but when needed, the access has to be fast. It can be used for file synchronization, file sharing, enterprise backups, and many other scenarios. This storage class has the same durability, low latency, and high throughput as the Standard storage class, with a lower cost, but its availability is slightly lower than the Standard storage class.
- Cold: The Cold storage class is ideal for storing data that is rarely (once per year) accessed. Its application scenarios include data archive and long-term backups. This storage class is secure, durable, and inexpensive, so it can be used to replace tape libraries. To keep cost low, it may take hours to restore data from the Cold storage class.

An object uploaded to a bucket inherits the storage class of the bucket by default. You can also specify a storage class for an object when you upload it.

Changing the storage class of a bucket does not change the storage classes of existing objects in the bucket, but newly uploaded objects will inherit the new storage class.

Table 1-1 Comparison of storage classes

Compared Item	Standard	Warm	Cold
Feature	Top-notch performance, high reliability and availability	Reliable, inexpensive storage with real-time access	Long-term retention of archived data at a low cost
Application scenarios	Cloud application, data sharing, content sharing, and hot data storage	Web disk applications, enterprise backup, active archiving, and data monitoring	Archive, medical image storage, video material storage, and replacement of tape libraries
Minimum storage duration[a]	N/A	30 days	90 days
Minimum measurement object size[b]	64 KB	64 KB	64 KB

Accessing OBS

OBS provides various tools for managing data stored in it. You can use any of the tools listed in **Table 1-2** to access and manage data in OBS.

Table 1-2 OBS resource management tools

Tool	Description
OBS Console	OBS Console is a web-based GUI for you to easily manage OBS resources.
OBS Browser+	OBS Browser+ is a Windows client that lets you easily manage OBS resources from your desktop.
API	OBS offers the REST API for you to access it from web applications with ease. By making API calls, you can upload and download data anytime, anywhere, over the Internet.

1.2 Advantages

Comparison Between OBS and On-Premises Storage Servers

In this information era, it becomes increasingly difficult for conventional on-premises storage servers to deal with the fast-growing data of enterprises. **Table 1-3** compares OBS with on-premises storage servers.

Table 1-3 Comparison between OBS and on-premises storage servers

Item	OBS	On-Premises Storage Server
Storage capacity	OBS provides unlimited storage capacity. All services and storage nodes are deployed in distributed clusters. You can expand each node or cluster separately, and you never have to worry about running out of space.	Such servers provide confined storage space due to the limited capacity of the hardware devices they use. When the storage space is not sufficient, you need to buy extra disks for manual expansion.
Security	OBS uses HTTPS and SSL protocols and can encrypt data during uploads. To keep data in transit and at rest safe, OBS uses access key IDs (AKs) and secret access keys (SKs) to authenticate user identities and adopts a range of approaches including IAM policies, bucket policies, access control lists (ACLs), and uniform resource locator (URL) validation.	The owner and users are exposed to security risks from cyber attacks, technical vulnerabilities, and accidental operations.
Costs	OBS is an out-of-the-box service that has no initial capital investment or time or labor costs and frees you from O&M.	The initial deployment of on-premises servers requires high investments and a long construction period, but it quickly lags behind as enterprise businesses change so fast. Additional expenditures are required to ensure security.

OBS Advantages

- **Data durability and service continuity**: OBS supports access of massive number of users.
- **Multi-level protection and authorization management**: Measures, including versioning, server-side encryption, URL validation, virtual private

cloud (VPC)-based network isolation, access log audit, and fine-grained access control are provided to keep data secure and trusted.

- **Highly concurrent access for hundreds of billions of objects**: With intelligent scheduling and response, optimized access paths, and technologies such as transmission acceleration, event notifications, and big data vertical optimization, you can store hundreds of billions of objects in OBS and still experience smooth concurrent access with ultra-high bandwidth and low latency.
- **Easy use and management**: OBS provides standard REST APIs to help you quickly move your workloads to cloud. Storage resources are linearly, infinitely scalable, without compromising performance. You do not have to plan storage capacity beforehand or worry about expansion or reduction.

1.3 Application Scenarios

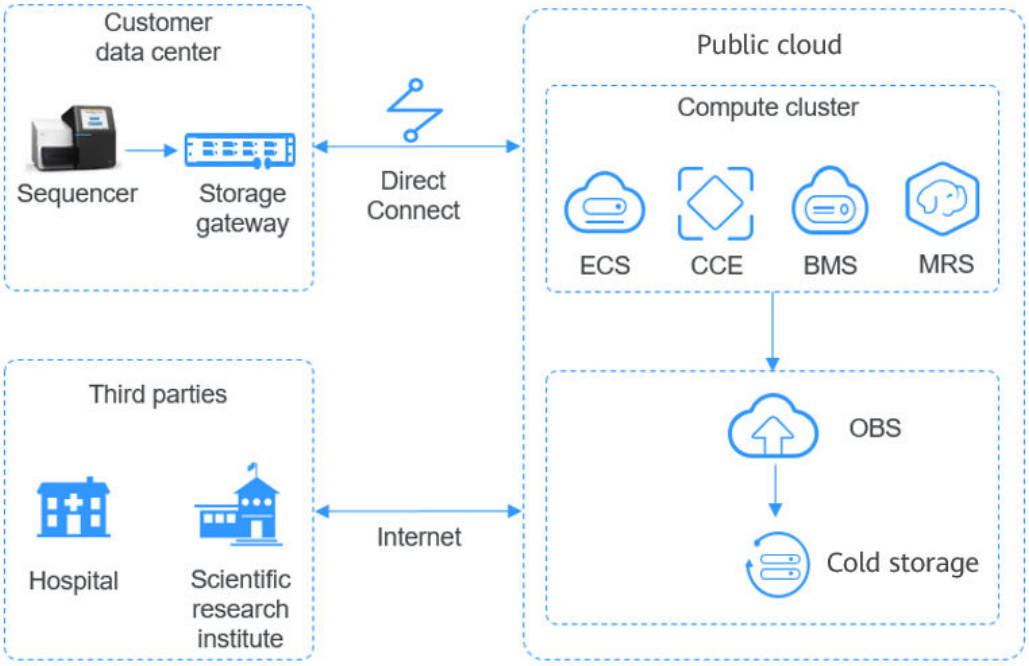
DNA Sequencing

Scenario Description

OBS is a reliable, cost-effective system for storing massive amounts of data and features high concurrency and low latency. It works with compute services to help you easily build a DNA sequencing platform.

You can use Direct Connect to automatically upload data from the sequencer in your data center to the cloud. You can then perform data analysis on the compute cluster (including ECS, CCE, and MRS services), and the analysis results will be stored in OBS. After an analysis is completed, the source DNA data will be automatically stored in the Cold storage class in OBS, and the sequencing results can be distributed to hospitals and scientific research institutes over the Internet.

Figure 1-2 DNA sequencing



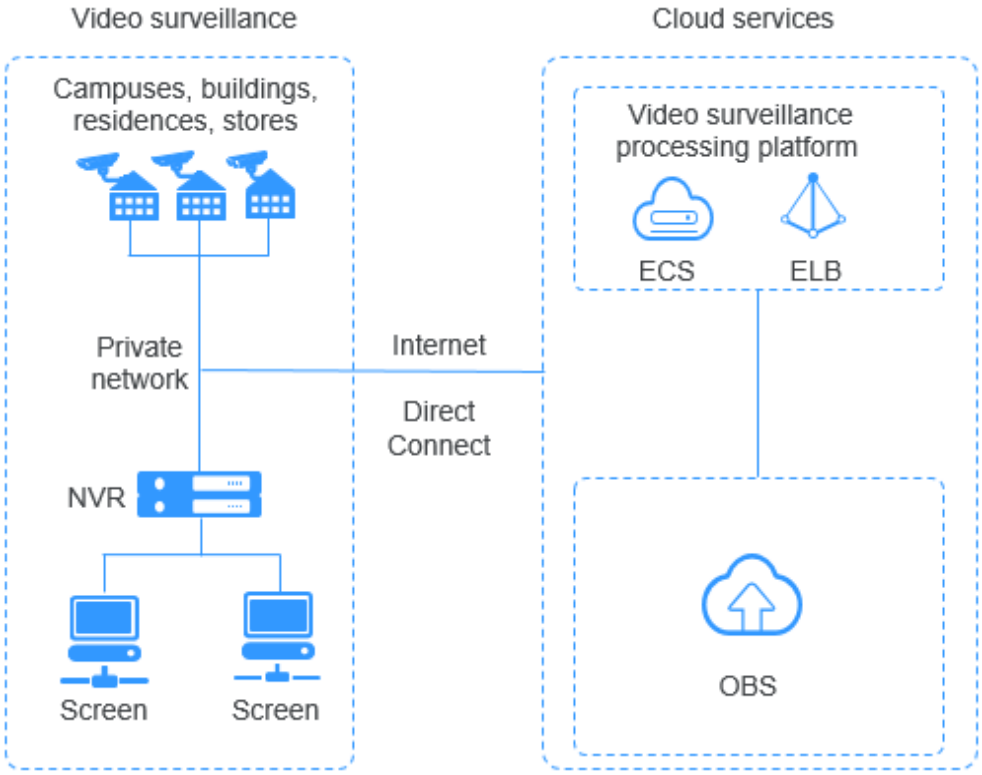
Intelligent Video Surveillance

Scenario Description

OBS provides reliable, inexpensive storage for virtually any amount of data. It features high performance and low latency and has a tiered storage class system (Standard, Warm, and Cold) to help reduce costs on storage.

You can upload surveillance videos recorded by cameras to the cloud over the Internet or using Direct Connect. Then segment the video files on the processing platform, which consists of ECS and ELB, and store video segmentation files in OBS. Later, you can download the video segmentation objects from OBS, and transfer them to terminal players.

Figure 1-3 Video surveillance

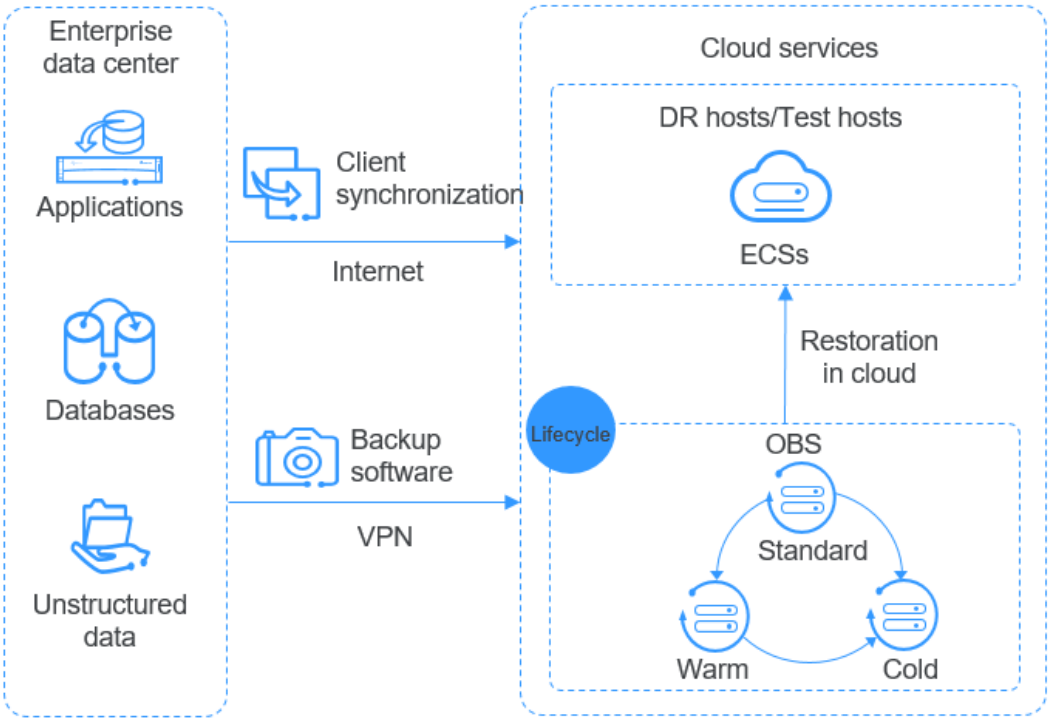


Backup and Archiving

Scenario Description

OBS offers a highly reliable, inexpensive storage system featuring high concurrency and low latency. It can hold massive amounts of data, meeting the archive needs for unstructured data of applications and databases.

Figure 1-4 Backup and archiving



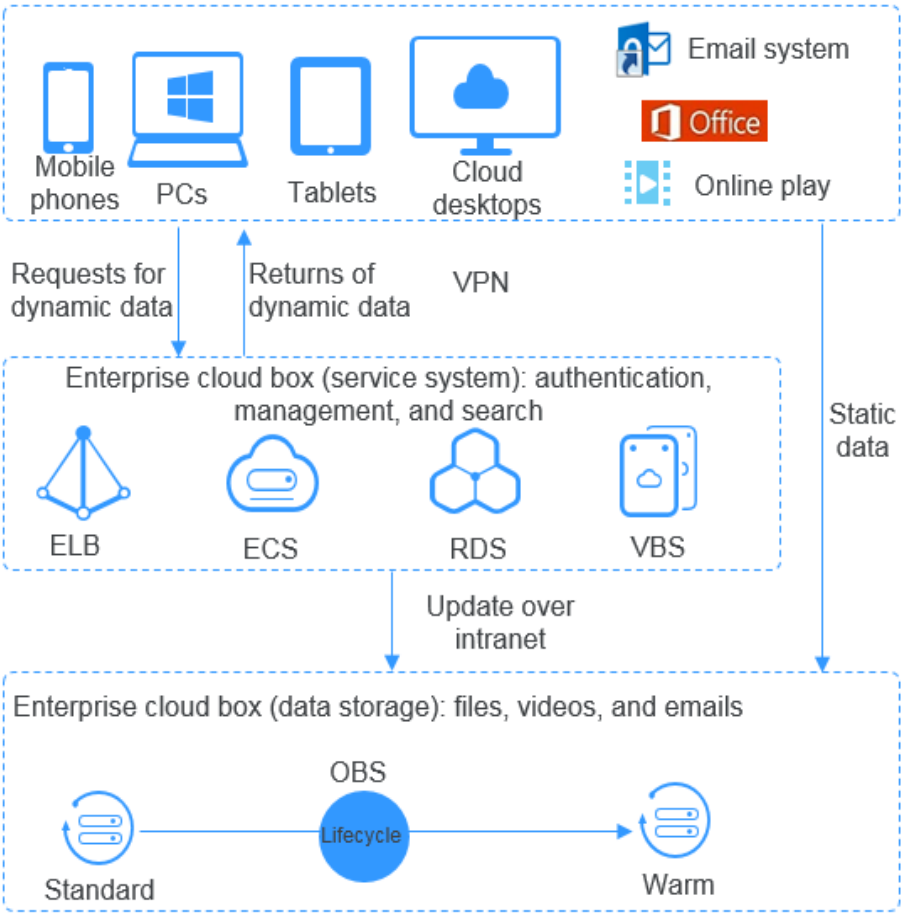
Enterprise Cloud Boxes (Web Disks)

Scenario Description

OBS works with cloud services such as ECS, ELB, RDS, and VBS to provide enterprise web disks with a reliable, inexpensive storage system featuring low latency and high concurrency. The storage capacity automatically scales as the volume of stored data grows.

Dynamic data on devices such as mobile phones, PCs, and tablets interacts with the enterprise cloud disk service system built on the cloud. Requests for dynamic data are sent to the service system for processing and then returned to devices, and the static data is stored in OBS. Service systems can process static data over the intranet. End users can directly request and read the static data from OBS. In addition, OBS provides the lifecycle management function to automatically change storage classes for objects, reducing storage costs.

Figure 1-5 Enterprise cloud boxes (web disks)



1.4 Permissions Management

You can use Identity and Access Management (IAM) to manage OBS permissions and control access to your resources. IAM provides identity authentication, permissions management, and access control.

You can create IAM users for your employees, and assign permissions to these users on a principle of least privilege (PoLP) basis to control their access to specific resource types. For example, you can create IAM users for software developers and assign specific permissions to allow them to use OBS resources but prevent them from being able to delete resources or perform any high-risk operations.

If your account does not require individual IAM users for permissions management, skip this section.

OBS Permissions

By default, new IAM users do not have any permissions assigned. You can assign permissions to these users by adding them to one or more groups and attaching policies to the groups. IAM provides preset system policies that define common permissions for different services, such as full control access and read-only. You can directly use these preset policies.

OBS is a global service deployed and accessed without specifying any physical region. OBS permissions are assigned to users in the global project, and users do not need to switch regions when accessing OBS.

Policy Types

- RBAC policy: An RBAC policy consists of permissions for an entire service. Users in a group with such a policy assigned are granted all the required permissions, including permissions for accessing and managing that service. RBAC policies do not support operation-specific permission control.
- Fine-grained policy: A fine-grained policy consists of API-based permissions for operations on specific resource types. Fine-grained policies, as the name suggests, allow for more fine-grained control than RBAC policies. Users with such fine-grained permissions can only perform specific operations on services.

NOTE

Due to data caching, an RBAC policy or a fine-grained policy involving OBS actions will take effect 10 to 15 minutes after it is attached to a user, an enterprise project, or a user group.

Table 1-4 lists all system policies of OBS.

Table 1-4 OBS system policies

Policy	Description	Policy Type
Tenant Administrator	Allows you to perform any operation on all cloud resources under the account. OBS policies are configured under **Global service** > **OBS**.	RBAC policy
Tenant Guest	Allows you to perform read-only operations on all cloud resources under the account. OBS policies are configured under **Global service** > **OBS**.	RBAC policy
OBS Buckets Viewer	Allows you to list buckets, obtain basic bucket information and bucket metadata, and list objects. OBS policies are configured under **Global service** > **OBS**.	RBAC policy
OBS Viewer	Allows you to list buckets, obtain basic bucket information and bucket metadata, and list objects. This policy is a system-defined policy of fine-grained authorization. Users with fine-grained authorization can use this policy and can create custom policy template based on this policy. OBS policies are configured under **Global service** > **OBS**.	Fine-grained policy

Policy	Description	Policy Type
OBS Operator	Allows you to perform all operations defined in OBS Viewer and to perform basic object operations, such as uploading objects, downloading objects, deleting objects, and obtaining object ACLs. This policy is a system-defined policy of fine-grained authorization. Users with fine-grained authorization can use this policy and can create custom policy template based on this policy. OBS policies are configured under **Global service** > **OBS**.	Fine-grained policy

The following table lists operations that can be performed under each set of OBS permission.

Table 1-5 Permissions and the allowed operations on OBS resources

Operation	Tenant Administrator	Tenant Guest	OBS Buckets Viewer	OBS Viewer	OBS Operator
Listing buckets	Supported	Supported	Supported	Supported	Supported
Creating buckets	Supported	Not supported	Not supported	Not supported	Not supported
Deleting buckets	Supported	Not supported	Not supported	Not supported	Not supported
Obtaining basic bucket information	Supported	Supported	Supported	Supported	Supported
Controlling bucket access	Supported	Not supported	Not supported	Not supported	Not supported
Managing bucket policies	Supported	Not supported	Not supported	Not supported	Not supported

Operation	Tenant Administrator	Tenant Guest	OBS Buckets Viewer	OBS Viewer	OBS Operator
Changing bucket storage classes	Supported	Not supported	Not supported	Not supported	Not supported
Listing objects	Supported	Supported	Supported	Supported	Supported
Listing objects with multiple versions	Supported	Supported	Not supported	Not supported	Not supported
Uploading files	Supported	Not supported	Not supported	Not supported	Supported
Creating folders	Supported	Not supported	Not supported	Not supported	Supported
Deleting objects	Supported	Not supported	Not supported	Not supported	Supported
Deleting folders	Supported	Not supported	Not supported	Not supported	Supported
Downloading objects	Supported	Supported	Not supported	Not supported	Supported
Deleting object versions	Supported	Not supported	Not supported	Not supported	Supported
Downloading object versions	Supported	Supported	Not supported	Not supported	Supported
Changing object storage classes	Supported	Not supported	Not supported	Not supported	Not supported
Restoring objects	Supported	Not supported	Not supported	Not supported	Not supported

Operation	Tenant Administrator	Tenant Guest	OBS Buckets Viewer	OBS Viewer	OBS Operator
Undeleting objects	Supported	Not supported	Not supported	Not supported	Supported
Deleting fragments	Supported	Not supported	Not supported	Not supported	Supported
Controlling object access	Supported	Not supported	Not supported	Not supported	Not supported
Configuring object metadata	Supported	Not supported	Not supported	Not supported	Not supported
Obtaining object metadata	Supported	Supported	Not supported	Not supported	Supported
Managing versioning	Supported	Not supported	Not supported	Not supported	Not supported
Managing logging	Supported	Not supported	Not supported	Not supported	Not supported
Managing event notifications	Supported	Not supported	Not supported	Not supported	Not supported
Managing tags	Supported	Not supported	Not supported	Not supported	Not supported
Managing lifecycle rules	Supported	Not supported	Not supported	Not supported	Not supported
Managing static website hosting	Supported	Not supported	Not supported	Not supported	Not supported
Managing CORS rules	Supported	Not supported	Not supported	Not supported	Not supported

Operation	Tenant Administrator	Tenant Guest	OBS Buckets Viewer	OBS Viewer	OBS Operator
Managing URL validation	Supported	Not supported	Not supported	Not supported	Not supported
Managing domain names	Supported	Not supported	Not supported	Not supported	Not supported
Appending data to objects	Supported	Not supported	Not supported	Not supported	Supported
Configuring an object ACL	Supported	Not supported	Not supported	Not supported	Not supported
Configuring the ACL for an object version	Supported	Not supported	Not supported	Not supported	Not supported
Obtaining object ACL information	Supported	Supported	Not supported	Not supported	Supported
Obtaining the ACL of a specific object version	Supported	Supported	Not supported	Not supported	Supported
Initiating a multipart upload	Supported	Not supported	Not supported	Not supported	Supported
Listing uploaded parts	Supported	Supported	Not supported	Not supported	Supported
Canceling multipart uploads	Supported	Not supported	Not supported	Not supported	Supported

OBS Resource Permissions Management

Access to OBS buckets and objects can be controlled by IAM user permissions, bucket policies, and ACLs.

For more information, see **Overview**.

1.5 Constraints

This section describes the constraints on the use of OBS features.

Table 1-6 OBS use constraints

Item	Description
Bandwidth	By default, the maximum bandwidth for read/write (GET/PUT) requests of a single account is 16 Gbit/s. If the bandwidth reaches this upper limit, flow control will be triggered.
Queries per second (QPS)	Default maximum QPS allowed by a single account: • 6,000 write requests (PUT Object) per second • 10,000 read requests (GET Object) per second • 1,000 listing requests (LIST) per second **NOTE** If you use sequential prefixes (sorted by timestamp or in alphabetical order) for object naming, object access requests may be concentrated in a specific partition, resulting in access hotspots. This limits the request rate in the hot partition and increases access latency. Random prefixes are recommended for naming objects so that requests are evenly distributed across partitions, achieving horizontal expansion.
Access rules	In consideration of the DNS resolution performance and reliability, OBS requires that the bucket name must precede the domain when a request carrying a bucket name is constructed to form a three-level domain name, also mentioned as virtual-hosted-style access domain name. For example, you have a bucket named **test-bucket** in the **my-kualalumpur-1** region, and you want to access the ACL of the **test-object** object in the bucket. The correct access URL is **https://test-bucket.obs.my-kualalumpur-1.alphaedge.tmone.com.my/test-object?acl**.

Item	Description
Buckets	• On OBS, each bucket name must be unique and cannot be changed. • After you create a bucket, its name and region cannot be changed. • An account (including all IAM users under this account) can create a maximum of 100 buckets and parallel file systems. You can use the fine-grained access control of OBS to properly plan and use buckets. For example, you can create folders in a bucket for storing objects with different prefixes and use **fine-grained permission control** to implement permission isolation between departments. • By default, there is no limit on the storage capacity of the entire OBS system or a single bucket, and any number of objects can be stored. • A bucket can be deleted only after all objects in the bucket have been deleted. • The name of a deleted bucket can be reused for another bucket or a parallel file system at least 30 minutes after the deletion.
Bucket inventories	See **Bucket Inventory Overview**.

Item	Description
Uploading objects	• OBS Console supports uploading files in a batch. A maximum of 100 files can be uploaded in a batch with the total size of no more than 5 GB. If you upload only one file in a batch upload, it cannot exceed 5 GB in size. • If you use OBS Browser+ or an API, you can upload a single object of up to 48.8 TB. • Batch upload is available only when: The bucket version is 3.0. • If versioning is disabled for your bucket and you upload a new file with the same name as the one you previously uploaded to your bucket, the new file automatically overwrites the previous one and does not retain its ACL information. If you upload a new folder using the same name that was used with a previous folder in the bucket, the two folders will be merged, and files in the new folder will overwrite those with the same name in the previous folder. • After versioning is enabled for your bucket, if the new file you upload has the same name as the one you previously uploaded to the bucket, a new file version will be added in the bucket. • Though any UTF-8 characters can be used in object keys (object names), it is recommended that object keys be named according to the **object key naming guidelines**. These guidelines help object key names substantially meet the requirements of DNS, web security characters, XML analyzers, and other APIs.
Deleting objects	If versioning is not enabled for a bucket, deleted objects cannot be recovered.
Restoring Cold objects	• If a Cold object is being restored, you cannot suspend or delete the restore task. • You cannot restore an object in the **Restoring** state. • After an object is restored, an object copy in the Standard storage class will be generated. This way, there is a Cold object and a Standard object copy in the bucket at the same time. The Standard object copy will be automatically deleted upon its expiration.
Lifecycle management	There is no limit on the number of lifecycle rules in a bucket, but the total size of XML descriptions about all lifecycle rules in a bucket cannot exceed 20 KB.

Item	Description
User-defined domain name binding	• Only buckets whose version is 3.0 or later support the binding of user-defined domain names. • By default, user-defined domain names allow requests for OBS over only HTTP. • A user-defined domain name can be bound to only one bucket. • Currently, the suffix of a user-defined domain name can contain 2 to 6 uppercase or lowercase letters.
ACLs	• A bucket ACL can have up to 100 grants. The total bucket ACL size cannot exceed 50 KB. • An object ACL can have up to 100 grants. The total object ACL size cannot exceed 50 KB.
Bucket policies	There is no limit on the number of bucket policies (statements) for a bucket, but the JSON descriptions of all bucket policies in a bucket cannot exceed 20 KB in total.
Parallel file systems	See the **Parallel File System Feature Guide**.

1.6 Related Services

Table 1-7 Related services

Function	Related Service	Reference
IAM provides the following functions: • User identity authentication • IAM user permission control • IAM agency configuration	Identity and Access Management (IAM)	**Permissions Management** **Configuring User Permissions**
Cloud Eye monitors OBS buckets, to collect statistics about the upload traffic, download traffic, the number of GET and PUT requests, the average Time to First Byte (TTFB) of GET requests, and the number of 4*xx* and 5*xx* errors.	Cloud Eye	**OBS Monitoring Metrics**

Function	Related Service	Reference
SMN sends OBS related alarms and event notifications, and triggers workflows.	Simple Message Notification (SMN)	SMN-Enabled Event Notifications
Tags are used to identify and organize buckets in OBS.	Tag Management Service (TMS)	Tag Overview
KMS encrypts files uploaded to the OBS.	Key Management Service (KMS)	Server-Side Encryption Overview
DNS resolves domain names configured for static website hosting in OBS.	Domain Name Service (DNS)	Using a User-Defined Domain Name to Configure Static Website Hosting

OBS can serve as a storage resource pool for other cloud services such as Relational Database Service (RDS) and Cloud Trace Service (CTS).

1.7 Basic Concepts

1.7.1 Objects

Objects are basic units stored in OBS. An object contains both data and the metadata that describes data attributes. Data uploaded to OBS is stored in buckets as objects.

An object consists of the following:

- A key that specifies the name of an object. An object key is a UTF-8 string up to 1,024 characters long. Each object within a bucket is uniquely identified by a key.
- Metadata that describes an object. The metadata is a set of key-value pairs that are assigned to objects stored in OBS. There are two types of metadata: system-defined metadata and custom metadata.
  - System-defined metadata is automatically assigned by OBS. Such metadata includes Date, Content-Length, Last-Modified, ETag, and more.
  - You can specify custom metadata to describe the object when you upload an object to OBS.
- Data that refers to the content of an object.

Objects are generally managed as files, but OBS, as an object-based service, has no concept of files and folders. For easy data management, OBS provides a

method to simulate folders. By adding a slash (/) to an object name, for example, **test/123.jpg**, you can specify **test** as a folder and **123.jpg** as the name of a file in the **test** folder. The key of the object is **test/123.jpg**.

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

On OBS Console, you can use folders the same way you use them in a file system.

Object Key Naming Guidelines

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

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

Alphanumeric characters (also known as unreserved characters)	0–9, a–z, and A–Z
Special characters (also known as reserved characters)	Exclamation mark (!) Hyphen (-) Underscore (_) Period (.) Asterisk (*) Single quote (') Left parenthesis (() Right parenthesis ())

The following are examples of valid object key names:

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

1.7.2 Buckets

Buckets are containers for storing objects. OBS provides flat storage in the form of buckets and objects. Unlike the conventional multi-layer directory structure of file systems, all objects in a bucket are stored at the same logical layer.

Each bucket has its own attributes, such as access permissions, storage class, and the region. You can specify access permissions, storage class, and regions when creating buckets. You can also configure advanced attributes to meet storage requirements in different scenarios.

Each bucket name in OBS is globally unique and cannot be changed after the bucket is created. The region where a bucket resides cannot be changed once the

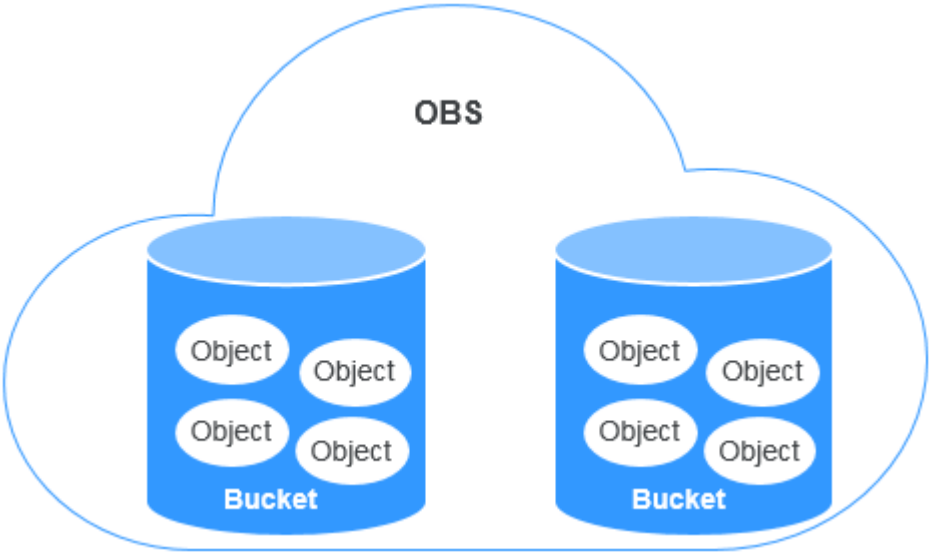
bucket is created. When you create a bucket, OBS creates a default access control list (ACL) that grants users permissions (such as read and write permissions) on the bucket. Only authorized users can perform operations such as creating, deleting, viewing, and configuring buckets.

An account (including all IAM users under this account) can create a maximum of 100 buckets and parallel file systems. However, there is no restriction on the number and total size of objects in a bucket.

OBS adopts the REST architectural style, and is based on HTTP and HTTPS. You can use URLs to locate resources.

Figure 1-6 illustrates the relationship between buckets and objects in OBS.

Figure 1-6 Relationship between objects and buckets



1.7.3 Parallel File System

Parallel File System, a sub-product of OBS, is a high-performance file system, with only milliseconds of access latency in the case of suitable networking and computing packages. OBS parallel file systems can support a TB/s-level bandwidth and handle millions of IOPS on the storage side, making it ideal for processing high-performance computing (HPC) workloads.

For details about PFS, see the *Parallel File System Feature Guide*.

1.7.4 Access Keys (AK/SK)

OBS uses access keys to authenticate the identity of a request sender.

Access keys comprise two parts: an access key ID (AK) and a secret access key (SK). AKs are used together with SKs to sign requests cryptographically, ensuring that the requests are confidential, complete, and correct.

When you use OBS APIs for secondary development and use an AK and SK pair for authentication, the signature must be calculated based on the algorithm defined by OBS and added to the request.

The authentication can be based on a permanent AK and SK pair, or based on a temporary AK/SK pair and security token.

Permanent AK/SK Pairs

NOTE

To access OBS in the AP-Kuala Lumpur-OP6 region, contact the administrator to obtain the AK and SK by referring to the **access key obtaining method**.

- Access key ID (AK): It is a unique identifier associated with a secret access key and is used to identify the sender of a request.
- Secret access key (SK): It is used in combination with the access key ID to sign requests. It can prevent requests from being tampered with and ensures the confidentiality and integrity of the requests.

Temporary AK/SK Pairs

A temporary AK/SK pair and security token assigned by OBS comply with the principle of least privilege and are for temporarily accessing OBS. They are valid from 15 minutes to 24 hours, and need to be obtained again once they expire. If the security token is missing from your request, a 403 error will be returned.

- Temporary access key ID (AK): It is a unique identifier associated with a temporary secret access key and is used to identify the sender of a request.
- Temporary secret access key (SK): It is used in combination with the temporary access key ID to sign requests. It can prevent requests from being tampered with and ensures the confidentiality and integrity of the requests.
- Security token: It is used together with the temporary AK and SK to access all resources of a specified account.

When using the following tools to access OBS resources, you need to use the AK/SK pair for security authentication.

Table 1-8 OBS resource management tools

Tool	AK/SK Configuration
OBS Browser+	Configure the AK and SK during account configuration.
obsfs	Configure the AK and SK during initial configuration.
APIs	Add the AK/SK pair to the request when computing the signature.

1.7.5 Endpoints and Domain Names

Endpoint: OBS provides an endpoint for each region. An endpoint is considered a domain name to access OBS in a region and is used to process requests of that region.

Endpoints vary depending on services and regions. The following table lists OBS endpoints.

Table 1-9 OBS endpoints

Region Name	Region	Endpoint	Protocol
AP-Kuala Lumpur-OP6	my-kualalumpur-1	obs.my-kualalumpur-1.alphaedge.tmone.com.my	HTTPS/HTTP

Bucket domain name: Each bucket in OBS has a domain name. A domain name is the address of a bucket and can be used to access the bucket over the Internet. It is applicable to cloud application development and data sharing.

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

Table 1-10 lists the bucket domain name and other domain names in OBS, including their structure and protocols.

Table 1-10 OBS domain names

Type	Structure	Description	Protocol
Regional domain name	**Endpoint**	Each region has an endpoint, which is the domain name of the region. For more information about OBS endpoints, see **Table 1-9**.	HTTPS HTTP
Bucket domain name	**BucketName.Endpoint**	After a bucket is created, you can use the domain name to access the bucket. You can compose the domain name according to the structure of bucket domain names, or you can obtain it from basic information of the bucket on OBS Console or OBS Browser+.	HTTPS HTTP

Type	Structure	Description	Protocol
Object domain name	**BucketName.Endpoint/ObjectName**	After an object is uploaded to a bucket, you can use the object domain name to access the object. You can spell out the domain name according to the structure of object domain names, or you can obtain it from the object details on OBS Console or OBS Browser+.	HTTPS HTTP
Static website domain name	**BucketName.obs-website.Endpoint**	A static website domain name is a bucket domain name when the bucket is configured to host a static website.	HTTPS HTTP
User-defined domain name	Self-owned domain name registered with a domain name provider	You can bind a user domain name to a bucket so that you can access the bucket through the user domain name.	HTTP

1.7.6 Region and AZ

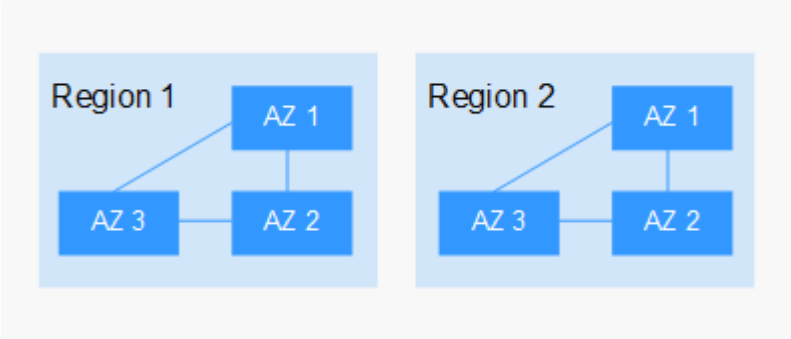
Concept

A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

- A region is a physical data center. Each region is completely independent, improving fault tolerance and stability. After a resource is created, its region cannot be changed.
- An AZ is a physical location using independent power supplies and networks. Faults in an AZ do not affect other AZs. A region can contain multiple AZs, which are physically isolated but interconnected through internal networks. This ensures the independence of AZs and provides low-cost and low-latency network connections.

Figure 1-7 shows the relationship between regions and AZs.

Figure 1-7 Regions and AZs



How Do I Select a Region?

You are advised to select a region close to you or your target users. This reduces network latency and improves access speed.

How Do I Select an AZ?

When determining whether to deploy resources in the same AZ, consider your applications' requirements for disaster recovery (DR) and network latency.

- For high DR capability, deploy resources in different AZs in the same region.
- For low network latency, deploy resources in the same AZ.

Regions and Endpoints

Before using an API to call resources, you must specify its region and endpoint.

2 OBS Console Operation Guide

2.1 Console Function Overview

Table 2-1 lists functions provided by OBS Console.

Table 2-1 OBS Console functions

Function	Description
Basic bucket operations	Allow you to create and delete buckets of different storage classes in specified regions (service areas), as well as change bucket storage classes.
Basic object operations	Allow you to manage objects, including uploading (multipart uploads included), downloading, and deleting objects, as well as changing object storage classes and restoring Cold objects.
Server-side encryption	Encrypts objects on the server side to enhance security of objects stored on OBS.
WORM	Protects objects from being deleted or tampered with within a specified period.
Object metadata	Allows you to set properties for objects.
Monitoring	• Cloud Eye can monitor the following OBS metrics: – Download Traffic – Upload Traffic – GET Requests – PUT Requests – First Byte Download Delay – 4*xx* Errors – 5*xx* Errors

Function	Description
Fragment management	Manages and clears fragments generated due to object upload failures.
Versioning	Stores multiple versions of an object in the same bucket.
Logging	Logs bucket access requests for analysis and auditing.
Event notification	Allows you to receive messages and emails from OBS.
Permission control	Controls access to OBS using IAM policies, bucket/object policies, and bucket/object access control lists (ACLs).
Lifecycle management	Allows you to configure lifecycle rules to periodically expire and delete objects or transition objects between storage classes.
Tags	Help you identify and classify buckets in OBS.
Static website hosting	Supports the hosting of static websites in buckets and the redirection of access requests for buckets.
User-defined domain name configuration	Enables you to bind your website domain name to a bucket domain name. If you want to migrate files from your website to OBS while keeping the website address unchanged, you can use this function.
URL validation	Prevents object links in OBS from being stolen by other websites.
Cross origin resource sharing (CORS)	Allows a web client in one origin to interact with resources in another one. Cross origin resource sharing (CORS) is a browser-standard mechanism defined by the World Wide Web Consortium (W3C). For general web page requests, website scripts and contents in one origin cannot interact with those in another because of Same Origin Policies (SOPs).
Bucket inventory	Periodically provides CSV files that list object information in the bucket and delivers the CSV files to the specified bucket.

2.2 Restrictions

Table 2-2 lists the web browser versions compatible with OBS Console.

Table 2-2 Supported web browser versions

Web Browser	Version
Internet Explorer	• Internet Explorer 9 (IE9) • Internet Explorer 10 (IE10) • Internet Explorer 11 (IE11)
Firefox	Firefox 55 and later
Chrome	Chrome 60 and later

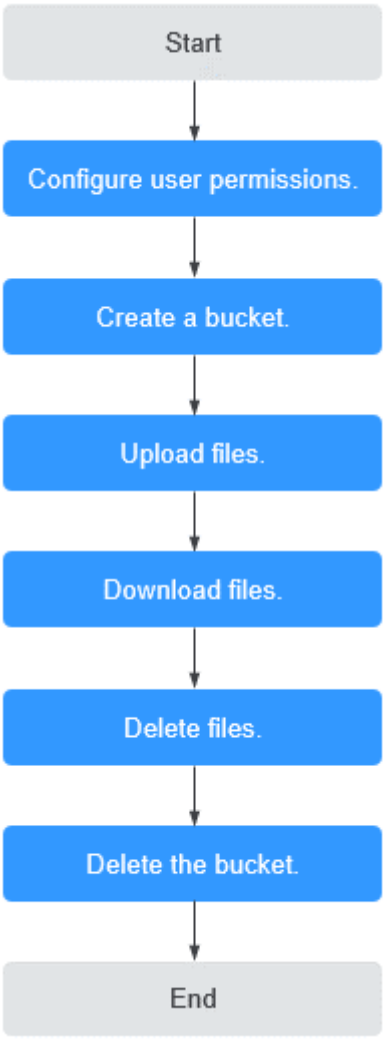
2.3 Getting Started

2.3.1 Process Description

OBS basic operations include bucket creation, object upload, and object download.

The follow-up sections describe how to complete the tasks illustrated in **Figure 2-1**.

Figure 2-1 OBS Console quick start



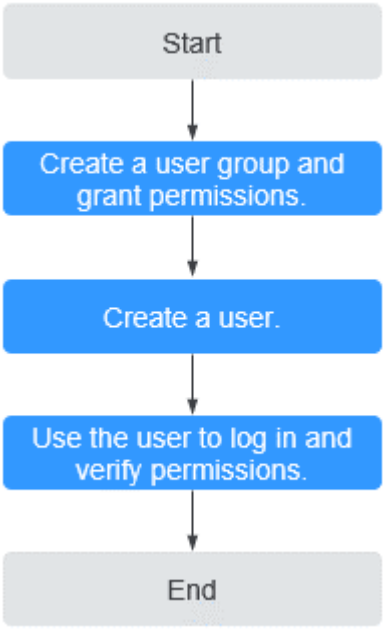
2.3.2 Configuring User Permissions

If your cloud service account does not need individual IAM users, then you may skip this section. Your permissions to use OBS functions are not affected.

OBS is separately deployed from other cloud resources. If IAM users are required, you need to grant them access permissions for OBS.

Process

Figure 2-2 Process of granting an IAM user the OBS permissions



Procedure

Step 1 Log in to the management console with your account.

Step 2 On the top menu bar, choose **Service List** > **Management & Deployment** > **Identity and Access Management**. The IAM console is displayed.

Step 3 Create a user group and assign OBS permissions to it.

A user group is a collection of users. By assigning permissions to a user group, you assign permissions to the users in this group. After you create an IAM user, add it to one or more user groups, so that it can inherit the permissions from the groups.

1. In the navigation pane, choose **User Groups**. The **User Groups** page is displayed.
2. Click **Create User Group**.
3. Enter a user group name and click **OK**.

   The user group is displayed in the user group list once the creation is complete.
4. Locate the user group you created and click **Modify** in the **Operation** column of the row.
5. In the **Group Permissions** area, locate the row that displays **Global service** > **OBS**, click **Attach Policy** in the **Operation** column, select the policy name, and click **OK**.

NOTE

In the **Policy Information** area, you can view the details about the policy.

Due to data caching, an RBAC policy or a fine-grained policy involving OBS actions will take effect 10 to 15 minutes after it is attached to a user, an enterprise project, or a user group.

Step 4 Create an IAM user. For details, see section "Creating an IAM User" in the *Identity and Access Management User Guide*.

Step 5 Use the created IAM user to log in to OBS Console and verify the user permissions.

----End

2.3.3 Creating a Bucket

This section describes how to create a bucket on OBS Console. A bucket is a container that stores objects in OBS. Before you can store data in OBS, you must create a bucket.

NOTE

An account can create a maximum of 100 buckets and parallel file systems.

Procedure

Step 1 In the upper right corner of the OBS Console homepage, click **Create Bucket**.

Step 2 Configure bucket parameters.

Table 2-3 Bucket parameters

Parameter	Description
Region	Geographic area where a bucket resides. For low latency and faster access, select the region nearest to you. Once the bucket is created, its region cannot be changed.

Parameter	Description
Bucket Name	Name of the bucket. A bucket name must be unique across all accounts and regions. Once a bucket is created, its name cannot be changed. According to the globally applied DNS naming rules, an OBS bucket name: • Must be unique across all accounts and regions. The name of a deleted bucket can be reused for another bucket or a parallel file system at least 30 minutes after the deletion. • Must be 3 to 63 characters long. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. • Cannot start or end with a period (.) or hyphen (-), and cannot contain two consecutive periods (..) or contain a period (.) and a hyphen (-) adjacent to each other. • Cannot be formatted as an IP address. **NOTE** When you access OBS through HTTPS using virtual hosted-style URLs, if the bucket name contains a period (.), the certificate verification will fail. To work around this issue, you are advised not to use periods (.) in bucket names.
Storage Class	Storage classes of a bucket. Different storage classes meet different requirements for storage performance and costs. • The Standard storage class is for storing a large number of hot files or small files that are frequently accessed (multiple times per month on average) and require quick retrieval. • The Warm storage class is for storing data that is less frequently accessed (less than 12 times per year on average) and requires quick retrieval. • The Cold storage class is for archiving data that is rarely accessed (once a year on average) and has no requirements for quick retrieval. For details, see **Storage Classes**.
Bucket Policy	Controls read and write permissions for buckets. • **Private**: No access beyond the bucket ACL settings is granted. • **Public Read**: Anyone can read objects in the bucket. • **Public Read and Write**: Anyone can read, write, or delete objects in the bucket.
Default Encryption	After you enable default encryption for the bucket, any object you upload to it will inherit the KMS encryption from the bucket by default. After you enable default encryption for the bucket, any object you upload to it will be encrypted with the **obs/default** key by default. You can also click **Create KMS Key** to create a key on the KMS console. Then select the created key on OBS Console for encryption.

Parameter	Description
WORM	When you enable write-once-read-many (WORM), you can configure a retention policy for the current bucket. The object version which the retention policy is applied to cannot be deleted within a specified period. You can only enable WORM when you create a bucket. Once enabled for a bucket, WORM cannot be disabled. When you enable WORM, OBS automatically enables versioning for the bucket, and versioning cannot be suspended later for that bucket.
Enterprise Project	You can add a bucket to an enterprise project for unified management. Create an enterprise project on the enterprise project page. The default enterprise project is named **default**. On the **Enterprise Project Management** page, create an enterprise project, create a user group and add users to this group, and then add the user group to the enterprise project. By doing so, users in this user group obtain the operation permissions for the buckets and objects in the enterprise project. **NOTE** Only an enterprise account can configure enterprise projects. OBS Viewer and OBS Operator are the fine-grained authorizations of the enterprise project user group in OBS.
Tags	Optional. Tags are used to identify and classify buckets in OBS. Each tag is represented by a key-value pair. For more information, see **Tag Overview**.

Step 3 Click **Create Now**.

----End

2.3.4 Uploading an Object

This section describes how to upload local files to OBS over the Internet. These files can be texts, images, videos, or any other type of files.

NOTE

OBS Console allows you to upload files in a batch. Up to 100 files can be uploaded at a time, with the total size of no more than 5 GB. If the size of a file exceeds 5 GB, use the OBS API for multipart upload.

If versioning is disabled for your bucket and you upload a new file with the same name as the one you previously uploaded to your bucket, the new file automatically overwrites the previous one and does not retain its ACL information. If you upload a new folder using the same name that was used with a previous folder in the bucket, the two folders will be merged, and files in the new folder will overwrite those with the same name in the previous folder.

After versioning is enabled for your bucket, if the new file you upload has the same name as the one you previously uploaded to the bucket, a new file version will be added in the bucket. For details about versioning, see **Versioning Overview**.

Prerequisites

- At least one bucket has been created.
- If you want to classify files, you can create folders and upload files to different folders. For details, see **Creating a Folder**.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

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

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

NOTE

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

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

You can also click **add files** to select files.

Step 6 (Optional) Select **KMS encryption** to encrypt the uploaded file. For details, see **Enabling Server-Side Encryption When Uploading an Object**.

NOTE

If the bucket has default encryption enabled, any object you upload will inherit the KMS encryption from the bucket by default.

Step 7 Click **Upload**.

----End

2.3.5 Downloading an Object

You can download files from OBS Console to your local computer.

Constraints

Objects in the Cold storage class can be downloaded only when they are in the **Restored** state.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 Select the file you want to download, and click **Download** or choose **More** > **Download As** on the right.

NOTE

In the **Download As** dialog box, right-click the object and choose **Copy Link Address** from the shortcut menu to obtain the object's download address.

----End

2.3.6 Deleting an Object

You can delete unnecessary files one by one or in a batch on OBS Console to save space and money.

NOTE

When WORM has been enabled for a bucket, versioning is also enabled for the bucket by default. If an object version has any WORM retention policy configured, this object version cannot be permanently deleted during the retention period. On the **Versions** tab of the object details page, you can choose **More** > **Extend Retention Period** in the **Operation** column in the row of the object version to check whether this version is within the retention period. If no WORM retention policy is configured for an object version, you can delete it on the **Versions** tab of the object details page.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 Select the file you want to delete, and choose **More** > **Delete** on the right.

You can select multiple files and click **Delete** above the file list to batch delete them.

Step 4 Click **Yes** to confirm the deletion.

----End

Important Notes

In big data scenarios, parallel file systems usually have deep directory levels and each directory has a large number of files. In such case, deleting directories from parallel file systems may fail due to timeout. To address this problem, you are advised to configure **a lifecycle rule** for directories so that they can be deleted in background based on the preset lifecycle rule.

2.3.7 Deleting a Bucket

You can delete unwanted buckets on OBS Console to free up the quota of buckets.

Prerequisites

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

NOTICE

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

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

Procedure

Step 1 In the bucket list on OBS Console, select the bucket you want to delete, and then click **Delete** on the right.

NOTE

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

Step 2 Click **Yes** to confirm the deletion.

----End

2.4 Storage Classes

2.4.1 Overview

Scenarios

As Internet develops, data storage scenarios become increasingly diverse. Limited storage classes cannot meet diverse storage and cost management requirements. OBS provides the following storage classes: Standard, Warm, and Cold.

Standard

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

  Objects smaller than 64 KB are regarded as 64 KB in size.
- Data restoration: N/A

  NOTE

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

Warm

- Scenarios: The Warm storage class is for storing data that is infrequently (less than 12 times a year) accessed, but when needed, the access has to be fast. It can be used for file synchronization, file sharing, enterprise backups, and many other scenarios.
- Redundancy: The Warm storage class provides two redundancy options. Multi-AZ storage means data is stored in multiple AZs, which enables more reliability. Single-AZ storage means data is stored in a single AZ, which is more cost-effective.
- Specifications: The minimum measurement object size is 64 KB. The minimum storage duration is 30 days.

  Objects smaller than 64 KB are regarded as 64 KB in size. For objects stored for less than 30 days, their storage duration is still 30 days.
- Data restoration: The system automatically restores data.

  NOTE

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

Cold

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

  Objects smaller than 64 KB are regarded as 64 KB in size. For objects stored for less than 90 days, their storage duration is still 90 days.
- Data restoration: Cold objects can be accessed only after being restored. The Cold storage class supports standard and expedited restoration. The standard restoration takes 3 to 5 hours, and the expedited restoration takes 1 to 5 minutes.

  NOTE

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

Comparison of Storage Classes

Item	Standard	Warm	Cold
Feature	Top-notch performance, high reliability and availability	Reliable, inexpensive storage with real-time access	Long-term storage for archived data at a low cost
Use cases	Cloud applications, data sharing, content sharing, and hot data storage	Web disk applications, enterprise backups, active archiving, and data monitoring	Storage of archives, medical imaging data, and videos, as well as replacement of tape libraries
Designed durability (single-AZ)	99.999999999% (11 nines)	99.999999999% (11 nines)	99.999999999% (11 nines)
Designed availability (single-AZ)	99.99%	99%	99%
Data access speed	Real-time access	Real-time access	Objects must be restored before you can access them. Standard restoration: 3 to 5 hours; Expedited restoration: 1 to 5 minutes.

2.4.2 Configuring Storage Classes for Buckets and Objects

Scenarios

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

Bucket and Object Storage Classes

You can specify the storage class for a bucket when creating the bucket. You can also change the storage class of a bucket after the bucket is created.

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

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

Configuring a Storage Class During Bucket Creation

Step 1 In the upper right corner of the OBS Console page, click **Create Bucket**.

Step 2 Configure bucket parameters.

Table 2-4 Bucket parameters

Parameter	Description
Region	Region where the bucket is located. For low latency and faster access, select the region nearest to you. Once the bucket is created, its region cannot be changed.
Bucket Name	Name of the bucket. A bucket name must be unique across all accounts and regions. Once a bucket is created, its name cannot be changed. According to the globally applied DNS naming rules, an OBS bucket name: • Must be unique across all accounts and regions. The name of a deleted bucket can be reused for another bucket or a parallel file system at least 30 minutes after the deletion. • Must be 3 to 63 characters long. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. • Cannot start or end with a period (.) or hyphen (-), and cannot contain two consecutive periods (..) or contain a period (.) and a hyphen (-) adjacent to each other. • Cannot be formatted as an IP address. **NOTE** When you access OBS through HTTPS using virtual hosted-style URLs, if the bucket name contains a period (.), the certificate verification will fail. To work around this issue, you are advised not to use periods (.) in bucket names.
Storage Class	The storage class of a bucket. These storage classes can meet different needs for storage performance and costs. • The Standard storage class is for storing a large number of hot files or small files that are frequently accessed (multiple times per month on average) and require quick retrieval. • The Warm storage class is for storing data that is less frequently accessed (less than 12 times per year on average) but requires quick retrieval. • The Cold storage class is for archiving data that is rarely accessed (once a year on average) and has less demanding requirements for quick retrieval. For details, see **Overview**.

Parameter	Description
Bucket Policy	Controls read and write permissions for buckets. • **Private**: Only users granted permissions by the bucket ACL can access the bucket. • **Public Read**: Anyone can read objects in the bucket. • **Public Read and Write**: Anyone can read, write, or delete objects in the bucket.
Enterprise Project	You can add a bucket to an enterprise project for unified management. Create an enterprise project on the enterprise project page. The default enterprise project is named **default**. On the **Enterprise Project Management Service** page, create an enterprise project, create a user group and add users to this group, and then add the user group to the enterprise project. By doing so, users in this user group obtain the operation permissions for the buckets and objects in the enterprise project. **NOTE** Only an enterprise account can configure enterprise projects. OBS Viewer and OBS Operator are the fine-grained authorizations of the enterprise project user group in OBS.
Default Encryption	When is enabled for a bucket, you can configure an object to inherit the bucket's KMS encryption settings when you upload the object to the bucket. After is enabled for a bucket, all objects uploaded to the bucket are automatically encrypted. The **obs/default** key is used by default. You can also click **Create KMS Key** to switch to the management console and create a customer master key. Then go back to OBS Console and select the key from the drop-down list.
WORM	When you enable write-once-read-many (WORM), you can configure a retention policy for the current bucket. The object version which the retention policy is applied to cannot be deleted within a specified period. You can only enable WORM when you create a bucket. Once enabled for a bucket, WORM cannot be disabled. When you enable WORM, OBS automatically enables versioning for the bucket, and versioning cannot be suspended later for that bucket.
Tags	Optional. Tags are used to identify and classify buckets in OBS. Each tag is represented by a key-value pair. For more information, see **Tag Overview**.

Step 3 Click **Create Now**.

----End

Configuring a Storage Class During Object Uploads

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

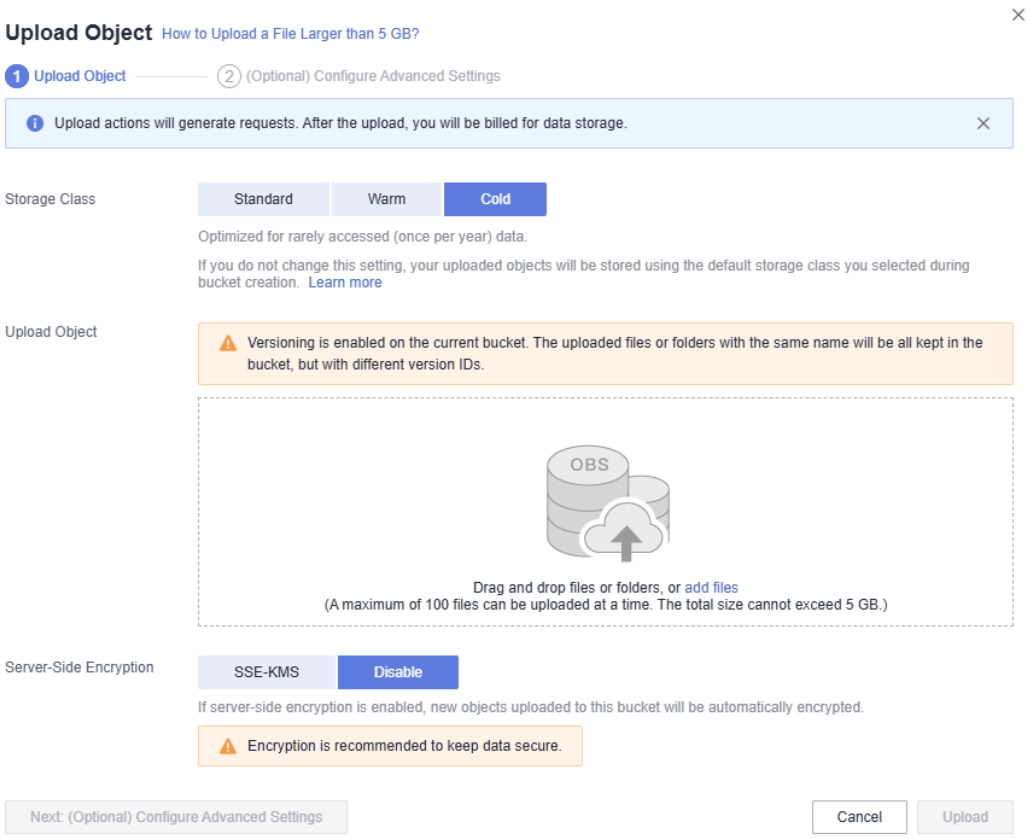
Step 2 In the navigation pane, choose **Objects**.

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

NOTE

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

Figure 2-3 Uploading objects



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

NOTE

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

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

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

Step 6 Click **Upload**.

----End

2.4.3 Changing the Storage Classes of Buckets and Objects

Scenarios

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

Constraints

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

Manually Changing the Storage Class of a Bucket

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

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

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

----End

Manually Changing the Storage Class of an Object

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

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

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

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

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

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

----End

Changing the Storage Class of an Object Using Lifecycle Rules

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Basic Configurations** > **Lifecycle Rules**.

Step 3 Click **Create**.

Step 4 Configure a lifecycle rule.

Basic Information:

- **Status**: Select **Enable** to enable this lifecycle rule after the configuration.
- **Rule Name**: It identifies a lifecycle rule. The rule name must be no longer than 255 characters.
- **Applies To**: It can be set to **Object name prefix** or **Bucket**.
  - **Object name prefix**: Objects with this specified prefix will be managed by the lifecycle rule. The prefix cannot start with a slash (/) or contain two consecutive slashes (//), and cannot contain the following special characters: \:*?"<>|
  - **Bucket**: All objects in the bucket will be managed by the lifecycle rule.

NOTE

- If the specified prefix overlaps with the prefix of an existing lifecycle rule, OBS regards these two rules as one and forbids you to configure the one you are configuring. For example, if there is already a rule with prefix **abc** in OBS, you cannot configure another rule whose prefix starts with **abc**.
- If there is a lifecycle rule based on prefix, you cannot create any rule to apply to the entire bucket.
- If there is a lifecycle rule applied to the entire bucket, you cannot create any rule based on prefix.

Current Version or **Historical Version**:

- If **Versioning** is disabled for a bucket, only **Current Version** can be configured.
- If **Versioning** was ever enabled for a bucket, both **Current Version** and **Historical Version** can be configured.

  The **Historical Version** appears only when versioning is enabled or suspended for the bucket.

NOTE

- **Current Version** and **Historical Version** are two concepts related to versioning. If versioning is enabled for a bucket, uploading objects with the same name to the bucket creates different object versions. The last uploaded object is called the current version, while those previously uploaded are called historical versions. For details, see *Versioning*.
- You can configure either **Current Version** or **Historical Version** or both of them.

- **Transition to Warm After (Days)**: After a specified number of days have passed since the last update, objects meeting specified conditions will be transitioned to Warm. This number must be at least 30.
- **Transition to Cold After (Days)**: After a specified number of days have passed since the last update, objects meeting specified conditions will be transitioned to Cold. If you configure objects to transition first to Warm and then to Cold, make sure the objects stay in Warm at least 30 days before they are transitioned to Cold. There are, however, no such constraints on time if you configure objects to transition to only Cold.
- **Delete Objects After (Days)**: After this number of days since the last update, objects meeting certain conditions will be expired and then deleted. This

number must be an integer larger than that specified for any of the transition operations.

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

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

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

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

On January 10, 2015, you created a rule to delete the objects prefixed with **log** one day later. You might encounter the following situations:

- Objects **log/test1.log** and **log/test2.log** uploaded on January 7, 2015 might be deleted upon the next system scan. The deletion could happen on January 10, 2015 or January 11, 2015, depending on the time of the system scan.
- For objects **log/clientlog.log** and **log/serverlog.log** uploaded on January 10, 2015, each system scan would determine whether one full day had passed since their last update. If any scan determined one full day had passed, those objects would be deleted upon that scan. The deletion might happen on January 11, 2015 or January 12, 2015.

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

NOTE

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

Step 5 Click **OK** to complete the lifecycle rule configuration.

----End

Precautions

- **Minimum measurement object size**

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

- **Minimum storage duration**

  This means that the minimum storage duration will be used even if objects are not stored for that long. For example, if an object is transitioned to Cold after being stored in Warm for 20 days, its minimum storage duration is still 30 days (the minimum storage duration for Warm).

Item	Standard	Warm	Cold
Minimum storage duration	N/A	30 days	90 days

- **Object restoration duration**

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

  Table 2-5 Object restoration duration

Restoration Mode	Duration of Restoration from Cold
Standard	3 to 5 hours
Expedited	1 to 5 minutes

2.5 Managing Buckets

2.5.1 Creating a Bucket

A bucket is a container that stores objects in OBS. Before you store data in OBS, you need to create a bucket.

NOTE

An account can create a maximum of 100 buckets and parallel file systems.

Procedure

Step 1 In the upper right corner of the OBS Console homepage, click **Create Bucket**.

Step 2 Configure bucket parameters.

Table 2-6 Bucket parameters

Parameter	Description
Region	Geographic area where a bucket resides. For low latency and faster access, select the region nearest to you. Once the bucket is created, its region cannot be changed.

Parameter	Description
Bucket Name	Name of the bucket. A bucket name must be unique across all accounts and regions. Once a bucket is created, its name cannot be changed. According to the globally applied DNS naming rules, an OBS bucket name: • Must be unique across all accounts and regions. The name of a deleted bucket can be reused for another bucket or a parallel file system at least 30 minutes after the deletion. • Must be 3 to 63 characters long. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. • Cannot start or end with a period (.) or hyphen (-), and cannot contain two consecutive periods (..) or contain a period (.) and a hyphen (-) adjacent to each other. • Cannot be formatted as an IP address. **NOTE** When you access OBS through HTTPS using virtual hosted-style URLs, if the bucket name contains a period (.), the certificate verification will fail. To work around this issue, you are advised not to use periods (.) in bucket names.
Storage Class	Storage classes of a bucket. Different storage classes meet different requirements for storage performance and costs. • The Standard storage class is for storing a large number of hot files or small files that are frequently accessed (multiple times per month on average) and require quick retrieval. • The Warm storage class is for storing data that is less frequently accessed (less than 12 times per year on average) and requires quick retrieval. • The Cold storage class is for archiving data that is rarely accessed (once a year on average) and has no requirements for quick retrieval. For details, see **Storage Classes**.
Bucket Policy	Controls read and write permissions for buckets. • **Private**: No access beyond the bucket ACL settings is granted. • **Public Read**: Anyone can read objects in the bucket. • **Public Read and Write**: Anyone can read, write, or delete objects in the bucket.
Default Encryption	After you enable default encryption for the bucket, any object you upload to it will inherit the KMS encryption from the bucket by default. After you enable default encryption for the bucket, any object you upload to it will be encrypted with the **obs/default** key by default. You can also click **Create KMS Key** to create a key on the KMS console. Then select the created key on OBS Console for encryption.

Parameter	Description
WORM	When you enable write-once-read-many (WORM), you can configure a retention policy for the current bucket. The object version which the retention policy is applied to cannot be deleted within a specified period. You can only enable WORM when you create a bucket. Once enabled for a bucket, WORM cannot be disabled. When you enable WORM, OBS automatically enables versioning for the bucket, and versioning cannot be suspended later for that bucket.
Enterprise Project	You can add a bucket to an enterprise project for unified management. Create an enterprise project on the enterprise project page. The default enterprise project is named **default**. On the **Enterprise Project Management** page, create an enterprise project, create a user group and add users to this group, and then add the user group to the enterprise project. By doing so, users in this user group obtain the operation permissions for the buckets and objects in the enterprise project. **NOTE** Only an enterprise account can configure enterprise projects. OBS Viewer and OBS Operator are the fine-grained authorizations of the enterprise project user group in OBS.
Tags	Optional. Tags are used to identify and classify buckets in OBS. Each tag is represented by a key-value pair. For more information, see **Tag Overview**.

Step 3 Click **Create Now**.

----End

Related Operations

After the bucket is created, you can change its storage class by performing the following steps:

Step 1 In the bucket list on OBS Console, select the target bucket and click **Change Storage Class** on the right.

Step 2 Select the desired storage class and click **OK**.

NOTE

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

----End

2.5.2 Viewing Basic Information of a Bucket

On OBS Console, you can view details about a bucket, including basic bucket statistics and information.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 Under **Basic Information**, view the basic bucket information.

NOTE

The statistics of **Used Capacity** and **Objects** are not real-time data, which are usually updated 15 minutes in delay.

----End

2.5.3 Searching for a Bucket

You can search for a bucket by characters contained in its name.

Procedure

Step 1 In the search box in the upper right corner of the OBS Console homepage, enter characters contained in the name of the bucket you want to search for.

Step 2 Click  .

Buckets that meet the search criteria are displayed in the bucket list.

For example, if you want to search for buckets whose names contain **test**, you only need to enter **test** in the search box and click  . Then, all buckets that contain **test** in their names are displayed.

----End

Related Operations

In the bucket list, click  next to the bucket name, storage class, region, used capacity, number of objects, or creation time to sort buckets.

2.5.4 Deleting a Bucket

You can delete unwanted buckets on OBS Console to free up the quota of buckets.

Prerequisites

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

NOTICE

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

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

Procedure

Step 1 In the bucket list on OBS Console, select the bucket you want to delete, and then click **Delete** on the right.

NOTE

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

Step 2 Click **Yes** to confirm the deletion.

----End

2.6 Managing Objects

2.6.1 Creating a Folder

This section describes how to create a folder on OBS Console. Folders facilitate data management in OBS.

Background Information

- Unlike a file system, OBS does not involve the concepts of file and folder. For easy data management, OBS provides a method to simulate folders. In OBS, an object is simulated as a folder by adding a slash (/) to the end of the object name on OBS Console. If you call the API to list objects, paths of objects are returned. In an object path, the content following the last slash (/) is the object name. If a path ends with a slash (/), it indicates that the object is a folder. The hierarchical depth of the object does not affect the performance of accessing the object.
- OBS Console does not support the download of folders. You can use OBS Browser+ to download folders.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 Click **Create Folder**, or click a folder in the object list to open it and click **Create Folder**.

Step 4 In the **Folder Name** text box, enter a name for the folder.

- You can create single-level or multi-level folders.
- The name cannot contain the following special characters: \:*?"<>|
- The name cannot start or end with a period (.) or slash (/).
- The folder's absolute path cannot exceed 1,023 characters.
- Any single slash (/) separates and creates multiple levels of folders at once.
- The name cannot contain two or more consecutive slashes (/).

Step 5 Click **OK**.

----End

Follow-up Procedure

You can click **Copy Path** on the right to copy the path of the folder and share it with others. Then they can open the bucket where the folder is stored and enter the path in the search box above the object list to find the folder.

2.6.2 Uploading an Object

This section describes how to upload local files to OBS over the Internet. These files can be texts, images, videos, or any other type of files.

Constraints

- OBS Console allows you to upload files in a batch. Up to 100 files can be uploaded at a time, with the total size of no more than 5 GB. If the size of a file exceeds 5 GB, use the OBS API for multipart upload.
- If versioning is disabled for your bucket and you upload a new file with the same name as the one you previously uploaded to your bucket, the new file automatically overwrites the previous one and does not retain its ACL information. If you upload a new folder using the same name that was used with a previous folder in the bucket, the two folders will be merged, and files in the new folder will overwrite those with the same name in the previous folder.
- After versioning is enabled for your bucket, if the new file you upload has the same name as the one you previously uploaded to the bucket, a new file version will be added in the bucket. For details, see **Versioning Overview**.

Prerequisites

- At least one bucket has been created.
- If you want to classify files, you can create folders and upload files to different folders. For details, see **Creating a Folder**.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

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

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

NOTE

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

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

You can also click **add files** to select files.

Step 6 (Optional) Select **KMS encryption** to encrypt the uploaded file. For details, see **Enabling Server-Side Encryption When Uploading an Object**.

NOTE

If the bucket has default encryption enabled, any object you upload will inherit the KMS encryption from the bucket by default.

Step 7 Click **Upload**.

----End

Related Operations

When uploading an object, you can specify a storage class for it. After the object is uploaded, you can also change its storage class by doing as follows:

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 Select the target object and choose **More** > **Change Storage Class** on the right.

Step 4 Select the desired storage class and click **OK**.

----End

NOTE

- You can manually change objects between storage classes.
  - From Standard to Warm or Cold
  - From Warm to Standard or Cold
  - From Cold to Standard or Warm. Before changing Cold objects, you must restore them first.
- After an object is changed to Cold, its restore status changes to **Unrestored**.
- You can also configure a lifecycle rule to change the storage class of an object. For details, see **Configuring a Lifecycle Rule**.

Follow-up Procedure

You can click **Copy Path** on the right of an object to copy its path.

You can share the path with others. Then they can open the bucket where the object is stored and enter the path in the search box above the object list to find the object.

2.6.3 Downloading an Object

You can download files from OBS Console to the system default path or a custom download path on your local computer.

Constraints

Objects in the Cold storage class can be downloaded only when they are in the **Restored** state.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 Select the file you want to download. Then, click **Download** or **More** > **Download As** on the right.

NOTE

In the **Download As** dialog box, right-click the object and choose **Copy Link Address** from the shortcut menu to obtain the object's download address.

----End

2.6.4 Searching for an Object or Folder

On OBS Console, you can search for files or folders by prefix.

Searching by Prefixes of Object Names

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 In the search box above the object list, enter the name prefix of the file or folder that you want to search for.

In the root directory of the bucket, files and folders whose name starts with the specified prefix are displayed.

NOTE

To search for objects within a folder, use either of the following methods:

- In the search box of the root directory, enter *folder path*/*object name prefix*. For example, if you enter **abc/123/example**, all files and folders with the **example** prefix in the **abc/123** folder will be displayed.
- Open the folder, and enter the object name prefix in the search box. For example, after you open the **abc/123** folder and enter **example** in the search box, all files and folders with the **example** prefix in the **abc/123** folder will be displayed.

Step 4 Click  . The search results are displayed in the object list.

----End

Related Operations

In the object list, click  next to the size or last modified time to sort objects.

2.6.5 Accessing an Object Using Its URL

You can grant anonymous users the read permission for an object so they can access the object using the shared object URL.

Prerequisites

Anonymous users have the read permission for the object. For details about permission granting, see **Granting Anonymous Users Permission to Access Objects**.

NOTE

Encrypted objects cannot be shared.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 Click the object to be shared. On the top part of the page, you can find the object's sharing link in the **Link** area.

Anonymous users can access the object by clicking this link. An object link (URL) is in the format of **https://***Bucket name*.*Domain name*/*Directory level*/*Object name*. If the object is in the root directory of the bucket, its URL does not contain any directory level.

NOTE

- To allow anonymous users to access objects in Cold storage using URLs, ensure that these objects are in the **Restored** state.

----End

2.6.6 Restoring an Object from Cold Storage

You must restore a Cold object before you can download it, access it with a URL, or configure its ACL or metadata.

Constraints

- If a Cold object is being restored, its restore task cannot be suspended or deleted.

- An object being restored cannot be restored again.
- After an object is restored, an object copy in the Standard storage class will be generated. This way, there is a Cold object and also its Standard copy in the bucket. The copy will be automatically deleted once the restoration expires.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 Select the file you want to restore, and click **Restore** on the right.

You can select multiple files and click **Restore** above the file list to batch restore the files.

NOTE

Objects that are being restored cannot be added for batch restore.

Step 4 Configure the validity period and speed of the restore. The following table describes the parameters.

Table 2-7 Parameters for restoring objects

Parameter	Description
Validity Period	How long the object will remain in the **Restored** state. It starts once the object is restored. The value is an integer ranging from 1 to 30 (days). The default value is **30**. For example, if you set **Validity Period** to **20** when restoring an object, 20 days after the object is successfully restored, its status will change from **Restored** to **Unrestored**.
Speed	How fast an object will be restored. • **Expedited**: Cold objects can be restored within 1 to 5 minutes. • **Standard**: Cold objects can be restored within 3 to 5 hours.

Step 5 Click **OK**.

The **Restoration Status** column in the object list displays the restore statuses of objects.

You can click  to manually refresh the restore status.

NOTE

The system checks the file restore status at UTC 00:00 every day. The system starts counting down the expiration time from the time when the latest check is complete.

----End

Related Operations

Within the validity period of a restored object, you can restore the object again. The validity period is then extended because it will start again when the latest restore is complete.

NOTE

If a restored object is restored again, its expiration time should be later than the time set for the previous restore. Assume that an object is restored on January 1 and will expire 30 days later (on January 30). If the object is restored again on January 10 and is made to be expired earlier than January 30 (less than 20 days later), this restore action is considered invalid.

2.6.7 Deleting an Object or Folder

Scenarios

On OBS Console, you can manually delete unneeded files or folders to release space and reduce costs.

Alternatively, you can configure lifecycle rules to periodically, automatically delete some or all of the files and folders from a bucket. For details, see **Configuring a Lifecycle Rule**.

In big data scenarios, parallel file systems usually have deep directory levels and each directory has a large number of files. In such case, deleting directories from parallel file systems may fail due to timeout. To address this problem, you are advised to delete directories in either of the following ways:

1. On the Hadoop client that has OBSA, an OBS client plugin, embedded, run the **hadoop fs - rmr obs://{***Name of a parallel file system***}/{***Directory name***}** command.
2. Configure **a lifecycle rule** for directories so that they can be deleted in background based on the preset lifecycle rule.

Background Information

Object Deletion with Versioning Enabled

When versioning is enabled for a bucket, OBS works slightly different when deleting different objects.

- Deleting a file or folder: The file or folder is not permanently deleted, but is retained in the **Deleted Objects** list and marked with the **Delete Marker**. In **Deleted Objects**, click the object name. On the **Versions** tab, you can see that the latest object version has the delete marker.
  - To permanently delete the file or folder, delete it again from the **Deleted Objects** list. For details, see **Procedure**.
  - To recover the deleted file, undelete it from the **Deleted Objects** list. For details, see **Undeleting an Object**.
- Deleting an object version: The version will be permanently deleted and cannot be recovered. If the deleted version is the latest one, the next latest version becomes the latest version.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 Select the file or folder you want to delete and choose **More** > **Delete** on the right.

You can select multiple files or folders and click **Delete** above the object list to batch delete them.

Step 4 Click **Yes** to confirm the deletion.

CAUTION

If you delete an object from a bucket with versioning enabled, the object is not permanently deleted but retained in the **Deleted Objects** list. All versions of the object are still kept in the bucket and are billed for storage. If you need to permanently delete the object, complete the following steps:

Step 5 If versioning is enabled for the bucket, delete the files or folders again from the **Deleted Objects** list to permanently delete them.

NOTE

In a bucket with WORM enabled, objects cannot be permanently deleted from the **Deleted Objects** list. You can permanently delete an object on its details page. For details, see **Related Operations**.

Likewise, folders cannot be permanently deleted from the **Deleted Objects** list either. To permanently delete a folder, you can only **configure a lifecycle rule**.

1. Click **Deleted Objects**.
2. In the **Operation** column of the file or folder to be deleted, click **Permanently Delete**.

   You can also select multiple files or folders and click **Permanently Delete** above the object list to batch delete them.

----End

Related Operations

When versioning is enabled, files in the **Deleted Objects** list also have multiple versions. Note the following points when deleting different versions of files:

- Deleting a version with the **Delete Marker** actually recovers this version instead of permanently deleting it. For details, see **Undeleting an Object**.
- Deleting a version without the **Delete Marker** permanently deletes this version. This version will not be recovered even if the object is recovered later.

2.6.8 Undeleting an Object

Scenarios

If a bucket has **versioning** enabled, you can recover a deleted object by undeleting it.

Background Information

Object Deletion with Versioning Enabled

When versioning is enabled for a bucket, OBS works slightly different when deleting different objects.

- Deleting a file or folder: The file or folder is not permanently deleted, but is retained in the **Deleted Objects** list and marked with the **Delete Marker**.
  - To permanently delete the file or folder, delete it again from the **Deleted Objects** list. For details, see **Deleting an Object or Folder**.
  - To recover the deleted file, undelete it from the **Deleted Objects** list. For details, see **Procedure**.
- Deleting an object version: The version will be permanently deleted and cannot be recovered. If the deleted version is the latest one, the next latest version becomes the latest version.

Object Recovery with Versioning Enabled

When a bucket has the versioning function enabled, deleting a file from the **Objects** list does not permanently delete it. The deleted file will be retained with the **Delete Marker** in the **Deleted Objects** list. You can recover the deleted file using the **Undelete** operation.

Note the following points when you undelete objects:

1. Only files can be undeleted but not folders.

   After you undelete a deleted file, the file is recovered and will appear in the **Objects** list. Then you can perform basic operations on the file as you normally do on other objects. If the file was stored in a folder before the deletion, it will be recovered to its original path after you undelete it.
2. Deleted files in the **Deleted Objects** also keep multiple versions. When deleting different versions of files, note the following points:
   - If you delete a version with the **Delete Marker**, it actually recovers this version instead of permanently deleting it. For details, see **Related Operations**.
   - If you delete a version without the **Delete Marker**, that version is permanently deleted. This version will not be recovered, even if the object is recovered later.
3. A deleted object must have at least one version without the **Delete Marker** in the **Deleted Objects** list. Otherwise, the object cannot be undeleted.

Prerequisites

- Versioning has been enabled for the bucket. For details, see **Configuring Versioning**.

- The file to be recovered is in the **Deleted Objects** list, and has at least one version without the **Delete Marker**.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 Click **Deleted Objects**.

Step 4 In the row of the deleted object that you want to recover, click **Undelete** on the right.

You can select multiple files and click **Undelete** above the object list to batch recover them.

----End

Related Operations

Recover a file by deleting its version with the Delete Marker:

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 Click **Deleted Objects**.

Step 4 Click the deleted file that you want to recover. The file information is displayed.

Step 5 On the **Versions** tab page, view all versions of the file.

- If you delete a version with the **Delete Marker**, the file will be recovered and retained in the **Objects** list.
- If you delete a version without the **Delete Marker**, that version will be permanently deleted.

----End

2.6.9 Managing Fragments

Background Information

Data can be uploaded to OBS using multipart uploads. There will be fragments generated, if a multipart upload fails because of the following causes (included but not limited to):

- The network is in poor conditions, and the connection to the OBS server is interrupted frequently.
- The upload task is manually suspended.
- The device is faulty.
- The device is powered off suddenly.

On OBS Console, storage used by fragments is charged. Clear fragments when they are not needed. If a file upload task fails, upload the file again.

NOTICE

Generated fragments take up storage space that is billable.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 Click **Fragments**, select the fragment that you want to delete, and click **Delete** on the right.

You can also select multiple fragments and click **Delete** above the fragment list to batch delete them.

Step 4 Click **Yes** to confirm the deletion.

----End

2.7 Server-Side Encryption

2.7.1 Server-Side Encryption Overview

After server-side encryption is enabled, objects to be uploaded will be encrypted and stored on the server. When objects are downloaded, they will be decrypted on the server first and then returned in plaintext to you.

Key Management Service (KMS) uses Hardware Secure Modules (HSMs) to ensure key security, enabling users to easily create and manage encryption keys. Keys are not displayed in plaintext outside HSMs, which prevents key disclosure. All operations performed on keys are controlled and logged, and usage of all keys is recorded, meeting regulatory compliance requirements.

The objects to be uploaded can be encrypted from the server side using the encryption service provided by KMS. You need to create a key using KMS or use the default key provided by KMS. Then you can use the key to perform server-side encryption when uploading objects to OBS.

OBS supports both SSE-KMS and server-side encryption with customer-provided keys (SSE-C) by calling APIs. In SSE-C mode, OBS encrypts objects on the server side using the keys and MD5 values provided by customers. Both methods use the AES-256 encryption algorithm.

2.7.2 Bucket Default Encryption

You can configure default encryption for an OBS bucket. Once configured, any objects you upload to the bucket will be encrypted with the specified KMS key by default.

You can enable default encryption when creating a bucket (see **Creating a Bucket**). You can also enable or disable default encryption for an existing bucket.

OBS only encrypts the objects uploaded after default encryption is enabled for the bucket, and does not encrypt those uploaded before. After default encryption is disabled, encryption status of existing objects in the bucket remains unchanged, and you can still encrypt objects when you upload them.

Enabling Default Encryption for a Bucket

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the **Basic Configurations** area, click **Default Encryption**. The **Default Encryption** dialog box is displayed.

Step 3 Select **Enable**.

Key **obs/default** is selected by default for KMS encryption. You can also click **Create KMS Key** to switch to the KMS management console and create a customer master key. Then go back to OBS Console and select the key from the drop-down list.

Step 4 Click **OK**.

----End

Disabling Default Encryption for a Bucket

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the **Basic Configurations** area, click **Default Encryption**. The **Default Encryption** dialog box is displayed.

Step 3 Select **Disable**.

Step 4 Click **OK**.

----End

2.7.3 Enabling Server-Side Encryption When Uploading an Object

OBS allows you to encrypt objects with server-side encryption so that the objects can be securely stored in OBS.

When you upload an object to a bucket with default encryption disabled, you can separately configure default encryption for the object. If the bucket has default encryption enabled, the object you upload inherits encryption from the bucket by default. You can also configure new encryption for the object.

Constraints

- The object encryption status cannot be changed.
- A key in use cannot be deleted. Otherwise, the object encrypted with this key cannot be downloaded.

Prerequisites

In the region where OBS is deployed, the **KMS Administrator** permission has been added to the user group. For details about how to add permissions, see the *IAM User Guide*.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 Click **Upload Object**. The **Upload Object** dialog box is displayed.

Step 4 Add the files to be uploaded.

Step 5 Enable **KMS encryption** and select a key that you have created on KMS.

NOTE

If the bucket has default encryption enabled, any object you upload will inherit the KMS encryption from the bucket by default.

After **KMS encryption** is selected, **obs/default** is selected by default as the key for the encryption. You can also click **Create KMS Key** to switch to the KMS management console and create a customer master key. Then go back to OBS Console and select the key from the drop-down list.

Step 6 Click **Upload**.

After the object is uploaded, you can view its encryption status on its details page.

----End

2.8 WORM

2.8.1 WORM Overview

OBS provides write-once-read-many (WORM) to protect objects from being deleted or tampered with within a specified period. WORM works at both the bucket and object levels in compliance mode.

Scenarios

In compliance mode, a WORM-protected object version cannot be overwritten or deleted by anyone, including the root user in your account.

When WORM is configured for a bucket, the protection applies to all objects in the bucket. When WORM is configured for an object version, the protection applies to the current object version only. No matter which type of WORM protection you want to use, you must enable WORM for the bucket first. A bucket-level WORM retention policy takes effect only for objects uploaded after the policy was configured. If an object is protected by a bucket-level WORM policy and an object-

level WORM policy at the same time, the object-level WORM policy takes precedence.

Important Notes

- When you enable WORM for a bucket, OBS automatically enables versioning and versioning cannot be suspended later for that bucket. WORM protects objects based on the object version IDs. Only object versions with any WORM retention policy configured can be protected. Assume that object **test.txt 001** is protected by WORM. If another file with the same name is uploaded, a new object version **test.txt 002** with no WORM policy configured will be generated. In such case, **test.txt 002** is not protected and can be deleted. If you download an object without specifying a version ID, the current object version (**test.txt 002**) will be downloaded.
- A lifecycle rule cannot delete WORM-protected objects, but can transition their storage class. After an object is no longer protected, it will be deleted when meeting the expiration rule in a lifecycle configuration.
- If you do not enable WORM when creating a bucket, you cannot enable or configure it for that bucket later. If you cannot configure WORM for a bucket, it may be because you did not enable WORM when you created the bucket or your bucket was created before this feature was available. In such case, to use WORM, you need to create a new bucket and enable WORM for it.
- Once you enable WORM for a bucket, you cannot disable it or suspend versioning for the bucket, but you can disable the default WORM policy for the bucket.
- If you have deregistered your account, the WORM-protected objects will be permanently deleted.
- WORM-based protection is not available for migration.
- The metadata of a WORM-protected object can still be modified.

2.8.2 Configuring WORM Retention

You can determine whether to enable WORM when creating a bucket. For details, see **Creating a Bucket**. When creating a bucket, if you enable WORM, you can continue to configure WORM after the bucket is created; if you do not enable it, you are not allowed to enable or configure it for that bucket later.

The following describes how to configure WORM retention after you create a bucket with WORM enabled.

Prerequisites

You have enabled WORM for the bucket when you create it.

Configuring WORM for a Bucket

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the **Basic Configurations** area, click **WORM Retention**. The **Configure WORM Retention** dialog box is displayed.

Step 3 Choose **Configure**. Keep the default **Compliance** retention mode and specify a default retention period.

NOTE

- Only the compliance retention mode is currently supported. In this mode, no users can delete protected object versions or change their retention mode during the specified retention period.
- During the specified default retention period, OBS prevents WORM-protected object versions from being deleted. You can configure a retention period in either days (from **1** to **36500**) or years (from **1** to **100**). The upper limit is 100 years.
- When you upload an object to a WORM-protected bucket, the object inherits the WORM retention from the bucket by default. You can also configure a different WORM retention for the object under advanced settings. If both a bucket-level and object-level WORM retention policy are applied to an object, the object-level retention policy will be used.

Step 4 Click **OK**.

----End

Skipping the WORM Retention Configuration

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the **Basic Configurations** area, click **WORM Retention**. The **Configure WORM Retention** dialog box is displayed.

Step 3 Choose **Skip**.

Step 4 Click **OK**.

----End

Extending the Retention Period

After WORM is configured for an object, you can go to the object details page and extend the retention period of an object version on the **Versions** page. Before the specified date, OBS prevents protected object versions from being deleted.

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 In the object list, click the object you want to go to the object details page.

Step 4 On the **Versions** tab page, view all versions of the object.

Step 5 Locate the object version for which you want to extend the retention period, choose **Extend Retention Period**, and select a date.

NOTE

A retention period can only be extended, but not shortened.

Assume that an object version was configured to be protected until March 30, 2023. If you want to extend the retention period on March 1, 2023, you can extend it to March 31, 2023 or a later date. If you extend the retention period on April 1, 2023, you can extend it to the current day (April 1, 2023) or a later date. If the current day is used, the object version will no longer be protected by WORM after 24:00 on that day.

----End

Related Operations

When uploading an object, configure a retention policy for the object. For details, see **Uploading an Object**.

2.9 Object Metadata

2.9.1 Object Metadata Overview

Object metadata is a set of name-value pairs that describe the object and is used for object management.

Currently, only the metadata defined by the system is supported.

The metadata defined by the system is classified into the following types: system-controlled and user-controlled. For example, metadata such as **Last-Modified** is controlled by the system and cannot be modified. You can call the API to modify the metadata such as **ContentLanguage**. The metadata that can be modified is described as follows:

Table 2-8 OBS metadata

Name	Description
ContentDisposition	Provides a default file name for the object that is being requested. When an object is being downloaded or accessed, the file with the default file name is directly displayed in the browser or a download dialog box is displayed if the file is being accessed. For example, select **ContentDisposition** as the metadata name and enter **attachment;filename="testfile.xls"** as the metadata value for an object. If you access the object through a link, a dialog box is directly displayed for downloading objects, and the object name is changed to **testfile.xls**. For details, see the definition about ContentDisposition in HTTP.

Name	Description
ContentLanguage	Indicates the language or languages intended for the audience. Therefore, a user can differentiate according to the user's preferred language. For details, see the definition about ContentLanguage in HTTP.
WebsiteRedirectLocation	Redirects an object to another object or an external URL. The redirection function is implemented using static website hosting. For example, you can perform the following operations to implement object redirection: 1. Set metadata of object **testobject.html** in the root directory of bucket **testbucket**. Select **WebsiteRedirectLocation** for **Name** and enter **http://www.example.com** for **Value**. **NOTE** OBS only supports redirection for objects in the root directory of a bucket. Redirection for objects located in folders of a bucket is not supported. 2. Configure static website hosting for bucket **testbucket**, and set the object **testobject.html** in the bucket as the default home page of the hosted static website. 3. If you access object **testobject.html** through the URL link provided on the **Configure Static Website Hosting** page, the access request is redirected to **http://www.example.com**.
ContentEncoding	Content encoding format when an object is downloaded. The options are as follows: ● Standard: **compress**, **deflate**, **exi**, **identity**, **gzip**, and **pack200-gzip** ● Others: **br**, **bzip2**, **lzma**, **peerdist**, **sdch**, **xpress**, **xz**
CacheControl	Cache behavior of the web page when the specified object is downloaded. ● Cacheability: **public**, **private**, **no-cache**, and **only-if-cached** ● Expiration time: **max-age=<*seconds*>**, **s-maxage=<*seconds*>**, **max-stale[=<*seconds*>]**, **min-fresh=<*seconds*>**, **stale-while-revalidate=<*seconds*>**, **stale-if-error=<*seconds*>** ● Re-verification and reloading: **must-revalidate**, **proxy-revalidate**, **immutable** ● Others: **no-store**, **no-transform**
Expires	Cache expiration time (GMT)

Name	Description
ContentType	File type of an object. For details, see **About Object Metadata Content-Type**.

NOTE

- When versioning is enabled for a bucket, you can set metadata for objects which are **Latest Version**, but cannot set metadata for objects which are **Historical Version**.
- Metadata cannot be configured for Cold objects.

2.9.2 About Object Metadata Content-Type

When an object is uploaded to OBS, the system automatically matches the value of **Content-Type** based on the file name extension of the object. When you access an object through a web browser, the system specifies an application to open the object according to the value of **Content-Type**. You can modify the **Content-Type** of an object based on its file name extension.

Table 2-9 Common Content-Type values

File Name Extension	Content-Type	File Name Extension	Content-Type
.* (binary stream, unknown file type)	application/octet-stream	.tif	image/tiff
.001	application/x-001	.301	application/x-301
.323	text/h323	.906	application/x-906
.907	drawing/907	.a11	application/x-a11
.acp	audio/x-mei-aac	.ai	application/postscript
.aif	audio/aiff	.aifc	audio/aiff
.aiff	audio/aiff	.anv	application/x-anv
.asa	text/asa	.asf	video/x-ms-asf
.asp	text/asp	.asx	video/x-ms-asf
.au	audio/basic	.avi	video/avi
.awf	application/vnd.adobe.workflow	.biz	text/xml
.bmp	application/x-bmp	.bot	application/x-bot
.c4t	application/x-c4t	.c90	application/x-c90

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

File Name Extension	Content-Type	File Name Extension	Content-Type
.fax	image/fax	.fdf	application/vnd.fdf
.fif	application/fractals	.fo	text/xml
.frm	application/x-frm	.g4	application/x-g4
.gbr	application/x-gbr	.	application/x-
.gif	image/gif	.gl2	application/x-gl2
.gp4	application/x-gp4	.hgl	application/x-hgl
.hmr	application/x-hmr	.hpg	application/x-hpgl
.hpl	application/x-hpl	.hqx	application/mac-binhex40
.hrf	application/x-hrf	.hta	application/hta
.htc	text/x-component	.htm	text/html
.html	text/html	.htt	text/webviewhtml
.htx	text/html	.icb	application/x-icb
.ico	image/x-icon	.ico	application/x-ico
.iff	application/x-iff	.ig4	application/x-g4
.igs	application/x-igs	.iii	application/x-iphone
.img	application/x-img	.ins	application/x-internet-signup
.isp	application/x-internet-signup	.IVF	video/x-ivf
.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

File Name Extension	Content-Type	File Name Extension	Content-Type
.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

File Name Extension	Content-Type	File Name Extension	Content-Type
.nws	message/rfc822	.odc	text/x-ms-odc
.out	application/x-out	.p10	application/pkcs10
.p12	application/x-pkcs12	.p7b	application/x-pkcs7-certificates
.p7c	application/pkcs7-mime	.p7m	application/pkcs7-mime
.p7r	application/x-pkcs7-certreqresp	.p7s	application/pkcs7-signature
.pc5	application/x-pc5	.pci	application/x-pci
.pcl	application/x-pcl	.pcx	application/x-pcx
.pdf	application/pdf	.pdf	application/pdf
.pdx	application/vnd.adobe.pdx	.pfx	application/x-pkcs12
.pgl	application/x-pgl	.pic	application/x-pic
.pko	application/vnd.ms-pki.pko	.pl	application/x-perl
.plg	text/html	.pls	audio/scpls
.plt	application/x-plt	.png	image/png
.png	application/x-png	.pot	application/vnd.ms-powerpoint
.ppa	application/vnd.ms-powerpoint	.ppm	application/x-ppm
.pps	application/vnd.ms-powerpoint	.ppt	application/vnd.ms-powerpoint
.ppt	application/x-ppt	.pr	application/x-pr
.prf	application/pics-rules	.prn	application/x-prn
.prt	application/x-prt	.ps	application/x-ps
.ps	application/postscript	.ptn	application/x-ptn

File Name Extension	Content-Type	File Name Extension	Content-Type
.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

File Name Extension	Content-Type	File Name Extension	Content-Type
.sdw	application/x-sdw	.sit	application/x-stuffit
.slb	application/x-slb	.sld	application/x-sld
.slk	drawing/x-slk	.smi	application/smil
.smil	application/smil	.smk	application/x-smk
.snd	audio/basic	.sol	text/plain
.sor	text/plain	.spc	application/x-pkcs7-certificates
.spl	application/futuresplash	.spp	text/xml
.ssm	application/streamingmedia	.sst	application/vnd.ms-pki.certstore
.stl	application/vnd.ms-pki.stl	.stm	text/html
.sty	application/x-sty	.svg	text/xml
.swf	application/x-shockwave-flash	.tdf	application/x-tdf
.tg4	application/x-tg4	.tga	application/x-tga
.tif	image/tiff	.tif	application/x-tif
.tiff	image/tiff	.tld	text/xml
.top	drawing/x-top	.torrent	application/x-bittorrent
.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

File Name Extension	Content-Type	File Name Extension	Content-Type
.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
.xdp	application/vnd.adobe.xdp	.xdr	text/xml
.xfd	application/vnd.adobe.xfd	.xfdf	application/vnd.adobe.xfdf
.xhtml	text/html	.xls	application/vnd.ms-excel
.xls	application/x-xls	.xlw	application/x-xlw
.xml	text/xml	.xpl	audio/scpls

File Name Extension	Content-Type	File Name Extension	Content-Type
.xq	text/xml	.xql	text/xml
.xquery	text/xml	.xsd	text/xml
.xsl	text/xml	.xslt	text/xml
.xwd	application/x-xwd	.x_b	application/x-x_b
.sis	application/vnd.symbian.install	.sisx	application/vnd.symbian.install
.x_t	application/x-x_t	.ipa	application/vnd.iphone
.apk	application/vnd.android.package-archive	.xap	application/x-silverlight-app

2.9.3 Configuring Object Metadata

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 Click the object to be operated, and then click the **Metadata** tab.

Step 4 Click **Add** and specify the metadata information.

Step 5 Click **Save**.

----End

2.10 Bucket Inventories

2.10.1 Bucket Inventory Overview

The bucket inventory function periodically generates lists of metadata information of objects in a bucket. Inventories help you better understand object statuses in the bucket.

An inventory is a CSV file. Inventory files are automatically uploaded to the specified bucket.

You specify that inventories are generated for objects with the same object name prefix. You can also determine the inventory generation interval and whether to list all object versions in the inventory file. The object metadata you specify in the

inventory include the file size, last modification time, storage class, ETag, multipart upload, encryption status, and replication status.

Constraints

- A bucket can have a maximum of 10 inventory rules.
- The source bucket (for which the inventory is configured) and the destination bucket (that stores the generated inventory files) must belong to the same account.
- The source and destination buckets must be in the same region.
- Inventory files must be in the CSV format.
- OBS can generate inventory files for all objects in a bucket or a group of objects whose names begin with the same prefix.
- If a bucket has multiple inventory rules, overlaps between the inventory rules are not allowed.
  - If a bucket already has an inventory rule for the entire bucket, new inventory rules that filter objects by prefixes cannot be created. If you need an inventory rule that covers only a subset of objects in the bucket, delete the inventory rule configured for the entire bucket.
  - If an inventory rule that filters objects by a specified prefix already exists, you cannot create an inventory rule for the entire bucket. To create an inventory rule for the entire bucket, make sure that the bucket has no other inventory rules that filter objects by specified prefixes.
  - If a bucket already has an inventory rule that filters objects by the object name prefix **ab**, the filter of a new inventory rule cannot start with **a** or **abc**. To create such a rule, you need to first delete the existing inventory rule that conflicts with the rule you will create.
- Bucket inventory files can be encrypted only in the SSE-KMS mode.
- The destination bucket cannot have default encryption enabled.

Content in an Inventory File

Table 2-10 lists all possible metadata fields that an inventory file can contain.

Table 2-10 Object metadata fields allowed in an inventory file

Metadata	Description
Bucket	Name of the source bucket
Key	Name of an object. Each object in a bucket has a unique key. Object names in the inventory file are URL-encoded using UTF-8 and must be decoded before you can use them.
VersionId	Object version ID. This field is not included in the inventory file if **ObjectVersions** in the inventory configuration is set to **Current version only**.

Metadata	Description
IsLatest	This field is set to **True** if the object version is the latest. This field is not included in the inventory file if **ObjectVersions** in the inventory configuration is set to **Current version only**.
IsDeleteMarker	When versioning is enabled for the source bucket, deleting an object will create a new piece of object metadata and set **IsDeleteMarker** of the metadata to **true**. This field is not included in the inventory file if **ObjectVersions** in the inventory configuration is set to **Current version only**.
Size	Object size, in bytes
LastModifiedDate	Object creation date or the last modification date
ETag	Hexadecimal digest of the object MD5. ETag is the unique identifier of the object content. It reflects whether the object content is changed. For example, if the ETag value is **A** when an object is uploaded but changes to **B** when the object is downloaded, it means that the object content has been changed.
StorageClass	Storage class of an object
IsMultipartUploaded	Whether an object is uploaded using multipart upload
ReplicationStatus	Cross-region replication status of an object
EncryptionStatus	Encryption status of an object

Inventory File Name

The name of an inventory file is in the following format:

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

- *destinationPrefix* indicates the prefix specified in the inventory configuration, which can be used to group inventory files. If no prefix is specified, the default prefix is **BucketInventory**.
- *sourceBucketName* indicates the source bucket for which the inventory is configured. This field can prevent conflicts when inventory files of different source buckets are saved to the same destination bucket.
- *inventoryId* can prevent conflicts when multiple inventory files of the same source bucket are sent to the same destination bucket.
- *yyyy-MM-dd'T'HH-mm'Z'* indicates the start time and date when the inventory generation begins scanning the bucket. Objects uploaded to the source bucket after this time may not be listed in the inventory file.
- **UUID_index.csv** indicates one of the inventory files.

The manifest.json File

If there are a large number of objects in a bucket, multiple inventory files may be generated for a single inventory configuration. It takes some time to generate these files. For example, if there are 200,000 objects in a bucket, it will take about 1.5 minutes to generate all inventory files. One or two hours after all inventory files are generated, a **manifest.json** file will be generated. The **manifest.json** file contains information about all inventory files generated this time, including:

- **sourceBucket** that indicates the name of the source bucket
- **destinationBucket** that indicates the name of the destination bucket
- **version** that indicates the inventory version
- **fileFormat** that indicates the inventory file format
- **fileSchema** that indicates the object metadata fields contained in the inventory files
- **files** that indicates the list of all inventory files
- **key** that indicates the inventory file name
- **size** that indicates the inventory file size, in bytes
- **inventoriedRecord** that indicates the number of inventory records

The following is an example of a **manifest.json** file.

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

"fileSchema":"Bucket,Key,Size,LastModifiedDate,ETag,StorageClass,IsMultipartUploaded,ReplicationStatus,EncryptionStatus",
	"files":[
		{
			"key":"inventory%2Fuser001%2Ftest_id%2F2019-01-03T12-28Z%2Ffiles%2F0000016813AF58E66806C1E2D7F15155_1.csv",
			"size":6705647390,
			"inventoriedRecord":70585762,
		}
	]
}
```

The name of the **manifest.json** file is as follows (for details about each field, see **Inventory File Name**):

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

The symlink.txt File

The **symlink.txt** file records the path of an inventory file. It helps quickly find all inventory files in big data scenarios. Apache Hive is compatible with the **symlink.txt** file. Hive can automatically find the **symlink.txt** file and the inventory files recorded in it.

The name of the **symlink.txt** file is as follows (for details about each field, see **Inventory File Name**):

```
destinationPrefix/sourceBucketName/inventoryId/hive/dt=YYYY-MM-DD-00-00/symlink.txt
```


2.10.2 Configuring a Bucket Inventory

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Inventories**. The inventory list is displayed.

Step 3 Click **Create**. The **Create Inventory** dialog box is displayed.

Step 4 Configure required parameters.

Table 2-11 Parameters for configuring a bucket inventory

Parameter	Description
Inventory Name	Name of a bucket inventory
Filter	Filter of an inventory. You can enter an object name prefix for OBS to create an inventory for objects with the specified prefix. Currently, only a prefix can be used as a filter. If the filter is not specified, the inventory covers all objects in the bucket. If a bucket has multiple inventories, their filters cannot overlap with each other.
Save Inventory Files To	Select a bucket (destination bucket) for saving generated inventory files. This bucket must be in the same region as the source bucket.
Inventory File Name Prefix	Prefix of the inventory file path. An inventory file will be saved in the following path: *Inventory file name prefix/Source bucket name/Inventory name/Date and time/files/*. If this parameter is not specified, OBS automatically adds **BucketInventory** as the prefix to inventory file's path.
Frequency	How frequently inventory files are generated. It can be set to **Daily** or **Weekly**.
Status	Inventory status. You can enable or disable the generation of inventories.

Step 5 Click **Next** to go to the **Configure Report** page.

Step 6 Configure the report.

Table 2-12 Report related parameters

Parameter	Description
Inventory Format	Inventory files can only be saved in CSV format.
Object Versions	Object versions that you want to list in an inventory file. It can be set to **Current version only** or **Include all versions**.
Optional Fields	Object information fields that can be contained in an inventory file, including **Size**, **Last modified date**, **Storage class**, **ETag**, **Multipart upload**, **Encryption status**, and **Replication status**.
Send Notification	If there is a new inventory file generated, a notification will be sent to the email address or mobile number specified in the SMN topic. If you enable the notification function, an SMN event notification rule will be created in the bucket where inventory files are stored. You can view details about the rule on the **Event Notification** page of the bucket. If you disable the notification function or modify the SMN topic, the SMN event notification rule will also be deleted or modified.

Step 7 Click **Next** to confirm the bucket policy.

OBS then automatically creates a bucket policy on the destination bucket to grant OBS permission to write inventory files to the bucket.

Step 8 Click **Create**.

----End

2.11 Permissions Control

2.11.1 Overview

OBS supports the following permission control mechanisms:

- IAM policies: IAM policies define the actions that can be performed on your cloud resources. In other words, IAM policies specify what actions are allowed or denied.
- Bucket policies and object policies:

  A bucket policy applies to the configured bucket and objects in the bucket. A bucket owner can use a bucket policy to grant permissions of buckets and objects in the buckets to IAM users or other accounts.

  NOTE

  In a bucket policy applied to a VDC read-only administrator, only read permissions (such as the permissions for listing or downloading objects) take effect. VDC read-only administrators cannot modify resources.

An object policy applies to specified objects in a bucket.

- Access control lists (ACLs): Control the read and write permissions for accounts. You can set ACLs for buckets and objects.

2.11.2 Permission Control Mechanisms

2.11.2.1 IAM Policies

You can create IAM users under a registered cloud service account, and then use IAM policies to control users' access permissions to cloud resources.

IAM policies define the actions that can be performed on your cloud resources. In other words, IAM policies specify what actions are allowed or denied.

IAM policies with OBS permissions take effect on all OBS buckets and objects. To grant an IAM user the permission to operate OBS resources, you need to assign one or more OBS permission sets to the user group to which the user belongs.

For details about OBS permissions controlled by IAM policies, see **Permissions Management**.

IAM policies Application Scenarios

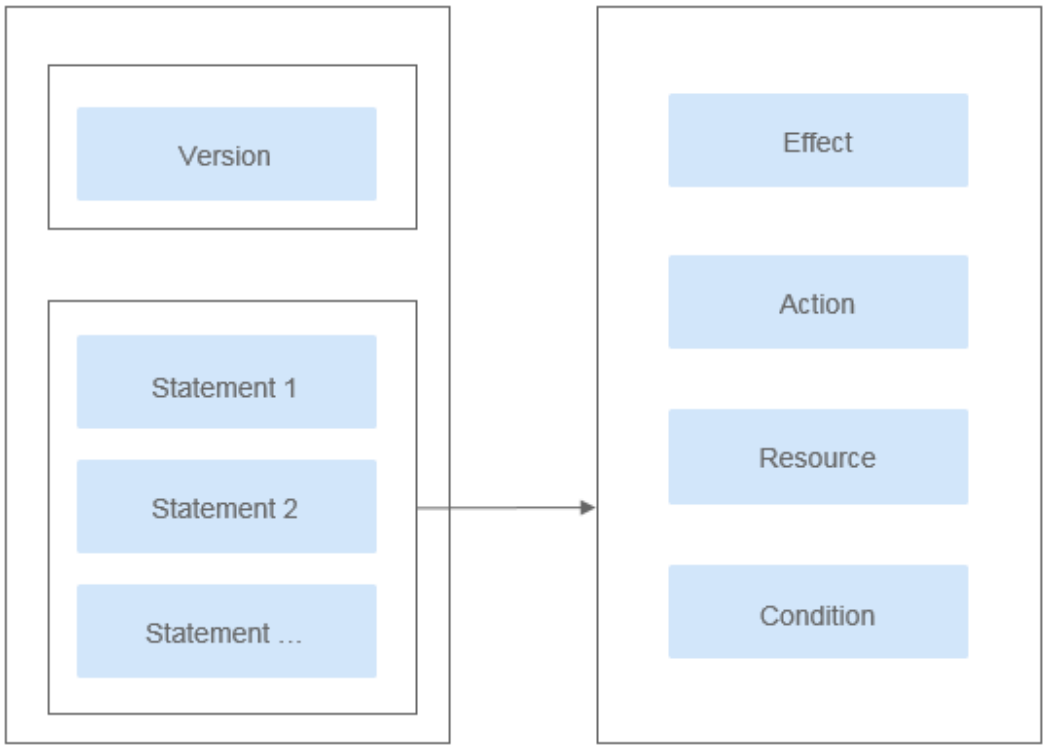
IAM policies are used to authorize IAM users under an account.

- Controlling permissions to cloud resources as a whole under an account
- Controlling permissions to all OBS buckets and objects under an account

Policy Structure and Syntax

A policy consists of a version and statements. Each policy can have multiple statements.

Figure 2-4 Policy structure



Policy syntax example:

```
{
   "Version": "1.1",
   "Statement": [
            {
         "Effect": "Allow",
         "Action": [
            "obs:bucket:HeadBucket",
            "obs:bucket:ListBucket",
            "obs:bucket:GetBucketLocation"
         ],
         "Resource": [
            "obs:*:*:bucket:*"
         ],
         "Condition": {
            "StringEndWithIfExsits": {
               "g:UserName": ["specialCharacter"]
            },
            "Bool": {
               "g:MFAPresent": ["true"]
            }
         }
      }
   ]
}
```


Table 2-13 Policy syntax parameters

Parameter	Description
Version	The version number of a policy. • **1.0**: RBAC policies. An RBAC policy consists of permissions for an entire service. Users in a group with such a policy assigned are granted all of the permissions required for that service. • **1.1**: Fine-grained policies. A fine-grained policy consists of API-based permissions for operations on specific resource types. Fine-grained policies, as the name suggests, allow for more fine-grained control on specific operations and resources than RBAC policies. For example: You can restrict an IAM user to access only the objects in a specific directory of an OBS bucket.

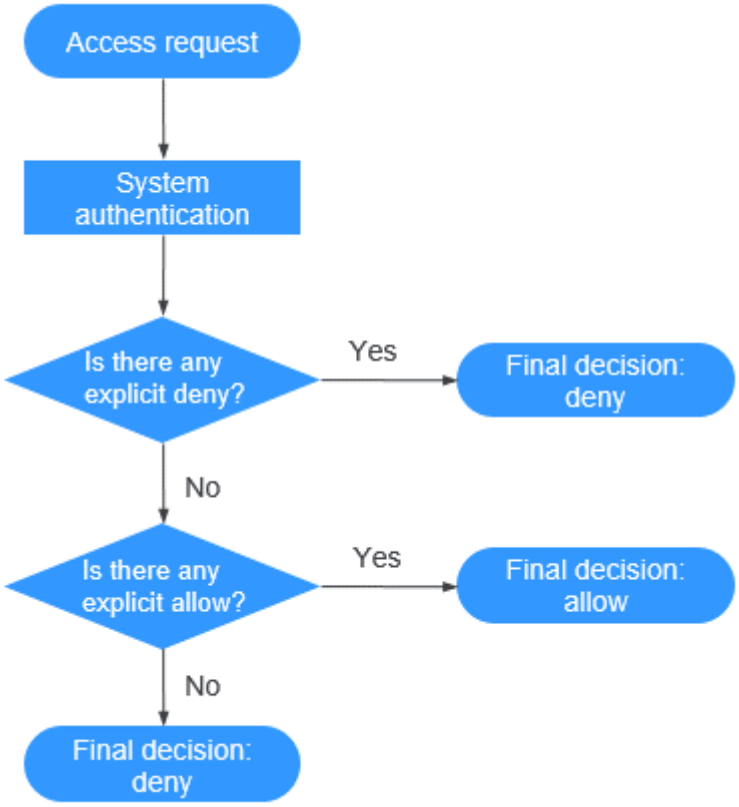
Parameter	Description
Statement	Permissions defined by a policy, including **Effect**, **Action**, **Resource**, and **Condition**. **Condition** is optional. • **Effect** The valid values for **Effect** are **Allow** and **Deny**. System policies contain only **Allow** statements. For custom policies containing both **Allow** and **Deny** statements, the **Deny** statements take precedence. • **Action** Permissions of specific operations on resources in the format of *Service name*:*Resource type*:*Operation*. A policy can contain one or more permissions. The wildcard (*) is allowed to indicate all of the services, resource types, or operations depending on its location in the action. OBS has two resource types: bucket and object. For details about actions, see the topics of "Bucket Actions" and "Object Actions" in the "IAM Policies and Supported Actions" section of the *Object Storage Service API Reference*. • **Resource** Resources on which the policy takes effect in the format of *Service name*:*Region*:*Domain ID*:*Resource type*:*Resource path*. The wildcard (*) is allowed to indicate all of the services, regions, resource types, or resource paths depending on its location in the action. In the JSON view, if **Resource** is not specified, the policy takes effect for all resources. The value of **Resource** supports uppercase (A to Z), lowercase (a to z) letters, digits (0 to 9), and the following characters: -_*./\. If the value contains invalid characters, use the wildcard character (*). OBS is a global service. Therefore, set **Region** to *****. **Domain ID** indicates the ID of the resource owner. Set it to ***** to indicate the ID of the account to which the resources belong. Examples: – **obs:*:*:bucket:***: all OBS buckets – **obs:*:*:object:my-bucket/my-object/***: all objects in the **my-object** directory of the **my-bucket** bucket • **Condition** Conditions for the policy to take effect (Optional). Format: *Condition operator*:*{Condition key:[Value 1, Value 2]}* The condition includes the global service condition name and cloud service condition name. The condition names supported by OBS are the same as those in the bucket policy. When configuring in IAM, add **obs:**. For details, see **Conditions**.

Parameter	Description
	The value of **Condition** can contain only uppercase (A to Z), lowercase (a to z) letters, digits (0 to 9), and the following characters: -,./_@#$%&. If the value contains unsupported characters, consider using the condition operator for fuzzy match, such as StringLike and StringStartWith. Examples: – **StringEndWithIfExists":{"g:UserName":["specialCharacter"]}**: The statement is valid for users whose names end with **specialCharacter**. – **"StringLike":{"obs:prefix":["private/"]}**: When listing objects in a bucket, you need to set prefix to **private/** or include **private/**.

Authentication of IAM policies

The authentication of IAM policies starts from the Deny statements. The following figure shows the authentication logic for resource access.

Figure 2-5 Authentication logic



NOTE

The actions in each policy are in the OR relationship.

1. A user accesses the system and makes an operation request.
2. The system evaluates all the permission policies assigned to the user.
3. In these policies, the system looks for explicit deny permissions. If the system finds an explicit deny that applies, it returns a decision of Deny, and the authentication ends.
4. If no explicit deny is found, the system looks for allow permissions that would apply to the request. If the system finds an explicit allow permission that applies, it returns a decision of Allow, and the authentication ends.
5. If no explicit allow permission is found, IAM returns a decision of Deny, and the authentication ends.

2.11.2.2 Bucket Policies and Object Policies

Bucket Owner and Object Owner

The owner of a bucket is the account that created the bucket. If the bucket is created by an IAM user under the account, the bucket owner is the account instead of the IAM user.

The owner of an object is the account that uploads the object, who may not be the owner of the bucket to which the object belongs. For example, account **B** is granted the permission to access a bucket of account **A**, and account **B** uploads a file to the bucket. In that case, instead of the bucket owner account **A**, account **B** is the owner of the object.

Bucket Policies

Bucket policies apply to buckets and the objects in them. By leveraging bucket policies, the owner of a bucket can grant IAM users or other accounts the permissions to operate the bucket and objects in the bucket.

NOTE

In a bucket policy applied to a VDC read-only administrator, only read permissions (such as the permissions for listing or downloading objects) take effect. VDC read-only administrators cannot modify resources.

Application Scenarios

- If no IAM policies are used for access control and you want to grant other accounts the permissions to access your OBS resources, you can use bucket policies.
- You can configure bucket policies to grant IAM users different access permissions on buckets.
- You can also use bucket policies to grant other accounts the permissions to access your buckets.

Standard Bucket Policies

There are three options for standard bucket policies.

- **Private**: No access beyond the bucket ACL settings is granted.
- **Public Read**: Anyone can read objects in the bucket.
- **Public Read and Write**: Anyone can read, write, or delete objects in the bucket.

After a bucket is created, the default bucket policy is **Private**. Only the bucket owner has the full control permissions over the bucket. To ensure data security, it is recommended that you do not use the **Public Read** or **Public Read and Write** policies.

Table 2-14 Standard bucket policies

Parameter	Private	Public Read	Public Read and Write
Effect	N/A	Allow	Allow
Principal	N/A	* (Any user)	* (Any user)
Resources	N/A	* (All objects in a bucket)	* (All objects in a bucket)
Actions	N/A	• GetObject • GetObjectVersion • ListBucket	• GetObject • GetObjectVersion • PutObject • DeleteObject • DeleteObjectVersion • ListBucket
Conditions	N/A	N/A	N/A

Custom Bucket Policies

The following three modes are provided to facilitate quick configuration:

- **Read-only**: With the **Read-only** mode, you only need to specify the **Principal** (authorized users). Then the authorized users have the read permission for the bucket and objects in the bucket, and can perform all GET operations on these resources.
- **Read and write**: With the **Read and write** mode, you only need to specify the **Principal** (authorized users). Then the authorized users have the full control permissions for the bucket and objects in the bucket, and can perform any operation on these resources.
- **Customized**: With the **Customized** mode, you can define the specific operation permissions that you want to grant to users and accounts by configuring the **Effect**, **Principal**, **Resources**, **Actions**, and **Conditions** parameters.

NOTE

On OBS Console, when you use a custom bucket policy to grant other users the permissions to operate resources in a bucket, you also need to grant these users the bucket read permission **ListBucket** (leaving the resource name blank indicates that the policy is applied to the entire bucket). Otherwise, the users may have no permission to access the bucket from OBS Console.

Object Policies

Object policies apply to objects in a bucket. A bucket policy is applicable to a set of objects (with the same object name prefix) or to all objects (specified by an asterisk *) in the bucket. To configure an object policy, select an object, and then configure a policy for it.

2.11.2.3 Bucket ACLs and Object ACLs

Access control lists (ACLs) enable you to manage access to buckets and objects, and define grantees and their granted access permissions. Each bucket and object has its own ACL that defines which accounts or groups are granted access and the type of access. When a request is received against a resource, OBS checks the ACL of the resource to verify whether the requester has necessary access permissions.

When you create a bucket or an object, OBS creates a default ACL that grants the resource owner full control (FULL_CONTROL) over the bucket or object.

An ACL supports up to 100 grants.

Who Is a Principal?

A principal can be an account or one of the predefined OBS groups. For details, see **Table 2-15**.

Table 2-15 Users supported by OBS

Principal	Description
Specific User	You can grant accounts access permissions to a bucket or an object using ACLs. Once a specific account is granted the access permissions, all IAM users who have OBS resource permissions under this account can have the same access permissions to operate the bucket or object. If you need to grant different access permissions to different IAM users, configure bucket policies. For details, see **Granting an IAM User Permissions to Operate a Specific Bucket**.

Principal	Description
Owner	The owner of a bucket is the account that created the bucket. The bucket owner has all bucket access permissions by default. The read and write permissions for the bucket ACL are permanently available to the bucket owner, and cannot be modified. The owner of an object is the account that uploads the object, who may not be the owner of the bucket to which the object belongs. The object owner has the read access to the object, as well as the read and write access to the object ACL, and such access permissions cannot be modified. **NOTICE** Do not modify the bucket owner's read and write access permissions for the bucket.
Anonymous User	If anonymous users are granted access to a bucket or an object, anyone can access the object or bucket without identity authentication.
Log Delivery User **NOTE** Only the bucket ACL supports authorizing permissions to the log delivery user.	A log delivery user only delivers access logs of buckets and objects to the specified target bucket. OBS does not create or upload any file to a bucket automatically. Therefore, if you want to record bucket access logs, you need to grant the permission to the log delivery user who will deliver the access logs to your specified target bucket. The user only delivers logs within the service scope of OBS. **NOTICE** After logging is enabled, the log delivery user group will be automatically granted the permission to read the bucket ACL and write the bucket where logs are saved. If you manually disable such permissions, bucket logging will fail.

What Permissions Can I Grant Using an ACL?

Table 2-16 lists the permissions you can grant using a bucket ACL.

Table 2-16 Access permissions controlled by a bucket ACL

Permission	Option	Description
Access to Bucket	READ	Used to obtain the list of objects in a bucket and the bucket metadata.
	WRITE	Used to upload, overwrite, and delete any object in a bucket.
Access to ACL	READ_ACP	Used to obtain the ACL of a bucket. The bucket owner has this permission permanently by default.

Permission	Option	Description
	WRITE_ACP	Used to update the ACL of a bucket. The bucket owner has this permission permanently by default.

Table 2-17 lists the permissions you can grant using an object ACL.

Table 2-17 Access permissions controlled by an object ACL

Permission	Option	Description
Access to Object	READ	Used to obtain the content and metadata of an object.
Access to ACL	READ_ACP	Used to obtain the ACL of an object. The object owner has this permission permanently by default.
	WRITE_ACP	Used to update the ACL of an object. The object owner has this permission permanently by default.

NOTE

Every time you change the bucket or object access permission setting in an ACL, it overwrites the existing setting instead of adding a new access permission to the bucket or object.

You can also set an ACL through a header when invoking the API for creating a bucket or uploading an object. Six types of predefined permissions can be set. Even with the predefined permissions configured, the bucket or object owner still has the full control over the resource. Table 2-18 lists the predefined permissions.

Table 2-18 Predefined access permissions in OBS

Predefined Access Permission	Description
private	Indicates that the owner of a bucket or an object has the full control over the resource. Any other users cannot access the bucket or object. This is the default access control policy.
public-read	If this permission is granted on a bucket, anyone can obtain the object list, multipart tasks, metadata, and object versions in the bucket. If it is granted on an object, anyone can obtain the content and metadata of the object.

Predefined Access Permission	Description
public-read-write	If this permission is granted on a bucket, anyone can obtain the object list, multipart tasks, metadata, and object versions in the bucket, and can upload or delete objects, initialize multipart upload tasks, upload parts, merge parts, copy parts, and cancel multipart upload tasks. If it is granted on an object, anyone can obtain the content and metadata of the object.
public-read-delivered	If this permission is granted on a bucket, anyone can obtain the object list, multipart tasks, metadata, and object versions, and obtain the object content and metadata in the bucket. It does not apply to objects.
public-read-write-delivered	If this permission is granted on a bucket, anyone can obtain the object list, multipart tasks, metadata, and object versions in the bucket, and can upload or delete objects, initialize multipart upload tasks, upload parts, merge parts, copy parts, and cancel multipart upload tasks. You can also obtain object content and metadata in the bucket. It does not apply to objects.
bucket-owner-full-control	If this permission is granted on a bucket, the bucket can be accessed only by its owner. If it is granted on an object, only the bucket or object owner has the full control over the object.

Bucket ACL Application Scenarios

ACLs control the read and write permissions for accounts and groups. ACL permission granularity is not as fine as bucket policies and IAM policies. Generally, it is recommended that you use IAM policies and bucket policies for access control.

You can configure the bucket ACL to:

- Grant an account the read and write access to the bucket, so that data in the bucket can be shared.

Object ACL Application Scenarios

ACLs control the read and write permissions for accounts and groups. ACL permission granularity is not as fine as bucket policies and IAM policies. Generally, it is recommended that you use IAM policies and bucket policies for access control.

It is recommended that you use object ACLs in the following scenarios:

- Object-level access control is required. A bucket policy can control access permissions for an object or a set of objects. If you want to further specify an

access permission for an object in the set of objects for which a bucket policy has been configured, then the object ACL is recommended for easier access control over single objects.

- An object is accessed through a URL. Generally, if you want to grant anonymous users the permission to read an object through a URL, use the object ACL.

2.11.2.4 Relationship Between a Bucket ACL and a Bucket Policy

Mapping Between Bucket ACLs and Bucket Policies

Bucket ACLs are used to control basic read and write access to buckets. Custom settings of bucket policies support more actions that can be performed on buckets. Bucket policies supplement bucket ACLs. In most cases (granting permissions to log delivery user groups excluded), you can use bucket policies to manage access to buckets. **Table 2-19** shows the mapping between bucket ACL access permissions and bucket policy actions.

Table 2-19 Mapping between bucket ACL access permissions and bucket policy actions

ACL Permission	Option	Mapped Action in a Custom Bucket Policy
Access to bucket	Read	• ListBucket • ListBucketVersions • ListBucketMultipartUploads
	Write	• PutObject • DeleteObject • DeleteObjectVersion
Access to ACL	Read	GetBucketAcl
	Write	PutBucketAcl

Mapping Relationship Between Object ACLs and Bucket Policies

Object ACLs are used to control basic read and write access permissions for objects. The custom settings of bucket policies support more actions that can be performed on objects. **Table 2-20** describes the mapping relationship between object ACL access permissions and bucket policy actions.

Table 2-20 Mapping relationship between object ACLs and bucket policies

Object ACL	Option	Mapped Action in a Custom Bucket Policy
Access to Object	Read	• GetObject • GetObjectVersion

Object ACL	Option	Mapped Action in a Custom Bucket Policy
Access to ACL	Read	• GetObjectAcl • GetObjectVersionAcl
	Write	• PutObjectAcl • PutObjectVersionAcl

2.11.2.5 How Does Authorization Work When Multiple Access Control Mechanisms Co-Exist?

- Based on the principle of least privilege, the default access control result is always deny, and an explicit deny statement always take precedence over an allow statement.

  Suppose that IAM policies grant a user the access to an object, a bucket policy denies the user's access to that object, and there is no ACL configured. Then user's access to the object will be denied.
- If no method specifies an allow statement, then the request will be denied by default. Only if no method specifies a deny statement and one or more methods specify an allow statement, will the request be allowed.

  For example, if a bucket has multiple bucket policies with allow statements, the adding of a new bucket policy with an allow statement will simply add the allowed permissions to the bucket, but the adding of a new bucket policy with a deny statement will result in a re-arrangement of the permissions. The deny statement will take precedence over allowed statements, even the denied permissions are allowed in other bucket policies.

Figure 2-6 Authorization process

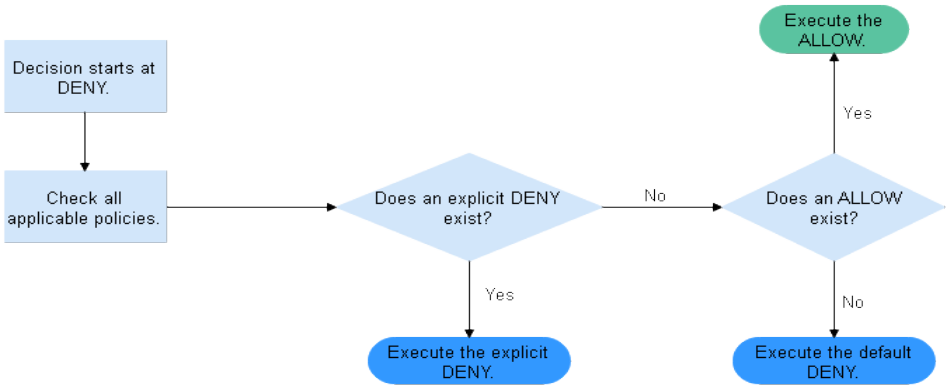


Figure 2-7 is a matrix of the IAM policies, bucket policies, and ACLs (allow and deny effects).

Figure 2-7 Matrix of the IAM policies, bucket policies, and ACLs (allow and deny effects)

<table>
<tr><td rowspan="2">Bucket Policy</td><td colspan="3">IAM Policy</td><td rowspan="2">ACL</td></tr>
<tr><td>Deny</td><td>Allow</td><td>Default Deny</td></tr>
<tr><td rowspan="2">Deny</td><td colspan="3" rowspan="2">Deny</td><td>Allow</td></tr>
<tr><td>Default Deny</td></tr>
<tr><td rowspan="2">Allow</td><td rowspan="3">Deny</td><td colspan="2" rowspan="2">Allow</td><td>Allow</td></tr>
<tr><td>Default Deny</td></tr>
<tr><td rowspan="2">Default Deny</td><td>Allow</td><td>Deny</td><td>Allow</td></tr>
<tr><td>Deny</td><td>Deny</td><td>Default Deny</td></tr>
</table>

2.11.3 Bucket Policy Parameters

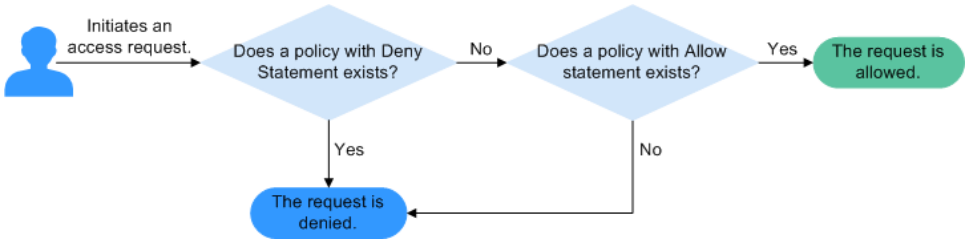
2.11.3.1 Effect

A bucket policy can either allow or deny requests.

- **Allow**: The policy allows the matched requests.
- **Deny**: The policy denies the matched requests.

When a bucket policy contains both the allow and deny effects, the deny effect prevails. The following figure shows the judgment process.

Figure 2-8 Determining a bucket policy when the allow and deny statements conflict



1. A user initiates an access request.
2. OBS preferentially searches for bucket policies that have the deny (explicit deny) effect. If a deny statement is found, OBS directly rejects the access. The access request ends.
3. If there is no deny statement, OBS searches for allow statements.
   - If an allow statement is found, OBS allows the access.
   - If no allow statement is found, OBS rejects the access. The access request ends.
4. If an error occurs during the judgment, an error message is generated and returned to the user who initiates the access request.

2.11.3.2 Principals

The principals indicate the users bucket policies apply to. These users can be accounts and IAM users. Target users can be specified in either of the following ways:

- **Include**: The policy applies to specified users.
- **Exclude**: The policy applies to users except the specified ones.

NOTE

In a bucket policy applied to a VDC read-only administrator, only read permissions (such as the permissions for listing or downloading objects) take effect. VDC read-only administrators cannot modify resources.

2.11.3.3 Resources

The resources a bucket policy is applied to can be the current entire bucket or objects in the bucket.

Resources can be specified in either of the following ways:

- **Include**: The bucket policy applies to specified OBS resources.
- **Exclude**: The bucket policy applies to OBS resources except the specified ones.

Applying a Bucket Policy to a Bucket

To specify the current bucket as the resource, select **Entire bucket**. When configuring actions for the policy, select bucket related actions.

Applying a Bucket Policy to Specified Objects

To apply the bucket policy to specified objects in a bucket, object-related actions must be configured in the policy. The configuration format is as follows:

- For an object, enter the object name (including its folder name if any). If you want to specify the **example.jpg** file in the **imgs-folder** folder in the bucket, enter the following content in the resource text box:

  imgs-folder/example.jpg
- For an object set, the wildcard asterisk (*) should be used. The asterisk * indicates an empty string or any combination of multiple characters. The format rules are as follows:
  - Use only one asterisk (*) to indicate all objects in a bucket.
  - Use *Object name prefix** to indicate objects starting with this prefix in a bucket. For example,

    imgs*
  - Use **Object name suffix* to indicate objects ending with this suffix in a bucket. For example,

    *.jpg

NOTE

Use commas (,) to separate one object (or object set) from another.

2.11.3.4 Actions

Actions are related to resources. When the resource is the current bucket, bucket-related actions should be configured in a bucket policy. When objects are specified as resources, object-related actions should be configured in a bucket policy.

Actions can be specified in either of the following ways:

- **Include**: The bucket policy applies to specified actions.
- **Exclude**: The bucket policy applies to actions except the specified ones.

Actions Related to Buckets

Table 2-21 Actions related to buckets

Type	Value	Description
General	*	The value supports a wildcard character (*) that indicates all operations can be performed.
	Get*	The value supports a wildcard character (*) that indicates all GET operations can be performed.
	Put*	The value supports a wildcard character (*) that indicates all PUT operations can be performed.
	List*	The value supports a wildcard character (*) that indicates all LIST operations can be performed.
Bucket	DeleteBucket	Deletes a bucket.
	ListBucket	Lists objects in a bucket, and obtains the bucket metadata.
	ListBucketVersions	Lists object versions in the bucket.
	ListBucketMultipar-tUploads	Lists multipart uploads.
	GetBucketAcl	Obtains the bucket ACL information.
	PutBucketAcl	Configures a bucket ACL.
	GetBucketCORS	Obtains the CORS configuration of the bucket.
	PutBucketCORS	Configures CORS for a bucket.
	GetBucketVersioning	Obtains the bucket versioning information.
	PutBucketVersioning	Configures versioning for a bucket.

Type	Value	Description
	GetBucketLocation	Obtains the bucket location.
	GetBucketLogging	Obtains the bucket logging information.
	PutBucketLogging	Configures logging for a bucket.
	GetBucketWebsite	Obtains the static website configuration of the bucket.
	PutBucketWebsite	Configures static website hosting for a bucket.
	DeleteBucketWebsite	Deletes the static website hosting configuration of the bucket.
	GetLifecycleConfiguration	Obtains the lifecycle rules of a bucket.
	PutLifecycleConfiguration	Configures a lifecycle rule for a bucket.

Actions Related to Objects

Table 2-22 Actions related to objects

Type	Value	Description
General	*	The value supports a wildcard character (*) that indicates all operations can be performed.
	Get*	The value supports a wildcard character (*) that indicates all GET operations can be performed.
	Put*	The value supports a wildcard character (*) that indicates all PUT operations can be performed.
	List*	The value supports a wildcard character (*) that indicates all LIST operations can be performed.
Object	GetObject	Obtains an object and its metadata.
	GetObjectVersion	Obtains the object of a specified version and its metadata.
	PutObject	Performs PUT upload, POST upload, multipart upload, initiation of uploaded parts, and assembling of parts.
	GetObjectAcl	Obtains the object ACL information.

Type	Value	Description
	GetObjectVersionAcl	Obtains the ACL information of a specified object version.
	PutObjectAcl	Configures an object ACL.
	PutObjectVersionAcl	Configures the ACL for a specified object version.
	DeleteObject	Deletes an object.
	DeleteObjectVersion	Deletes a specified object version.
	ListMultipartUpload-Parts	Lists uploaded parts.
	AbortMultipartUpload	Aborts a multipart upload.

2.11.3.5 Conditions

In addition to effect, principals, resources, and actions, you can specify conditions for a bucket policy. A bucket policy takes effect only when its condition expressions match values contained in the request. **Conditions** is an optional parameter. You can determine whether to use this parameter based on service requirements.

For example, if account **A** needs to be granted with full control permissions for an object uploaded by account **B** in bucket **example**, you can specify that the upload request must contain the **acl** key and set the policy effect to **Allow** for account **A**. The complete condition expression is as follows:

Condition Operator	Key	Value
StringEquals	acl	bucket-owner-full-control

A condition consists of three parts: condition operator, key, and value. Condition operators and keys are associated with each other. For example:

- If a string type condition operator is selected, such as **StringEquals**, the key can only be of the string type, such as **UserAgent**.
- If a date type key is selected, such as **CurrentTime**, the condition operator can only be of the date type, such as **DateEquals**.

Table 2-23 describes the predefined condition operators provided by OBS.

Table 2-23 Condition operators

Type	Key	Description
String	StringEquals	Strict matching. Short version: streq
	StringNotEquals	Strict negated matching. Short version: strneq
	StringEqualsIgnoreCase	Strict matching, ignoring case. Short version: streqi
	StringNotEqualsIgnore-Case	Strict negated matching, ignoring case. Short version: strneqi
	StringLike	Loose case-sensitive matching. The values can include a multi-character match wildcard (*) or a single-character match wildcard (?) anywhere in the string. Short version: strl
	StringNotLike	Negated loose case-sensitive matching. The values can include a multi-character match wildcard (*) or a single-character match wildcard (?) anywhere in the string. Short version: strnl
Numeric	NumericEquals	Strict matching. Short version: numeq
	NumericNotEquals	Strict negated matching. Short version: numneq
	NumericLessThan	"Less than" matching. Short version: numlt
	NumericLessThanEquals	"Less than or equals" matching. Short version: numlteq
	NumericGreaterThan	"Greater than" matching. Short version: numgt
	NumericGreaterThanEqu-als	"Greater than or equals" matching. Short version: numgteq
Date	DateEquals	Strict matching. Short version: dateeq
	DateNotEquals	Strict negated matching. Short version: dateneq
	DateLessThan	Indicates that the date is earlier than a specific date. Short version: datelt

Type	Key	Description
	DateLessThanEquals	Indicates that the date is earlier than or equal to a specific date. Short version: datelteq
	DateGreaterThan	Indicates that the date is later than a specific date. Short version: dategt
	DateGreaterThanEquals	Indicates that the date is later than or equal to a specific date. Short version: dategteq
Boolean	Bool	Strict Boolean matching
IP address	IpAddress	Takes effect only on a specified IP address or IP address range. Example: **x.x.x.x/24**
	NotIpAddress	Takes effect only on all except the specified IP address or IP address range. Example: **x.x.x.x/24**

A condition can contain any of the three types of keys: general keys, keys related to bucket actions, and keys related to object actions.

Table 2-24 General keys

Key	Type	Description
CurrentTime	Date	Indicates the date when the request is received by the server. The date format must comply with ISO 8601.
EpochTime	Numeric	Indicates the time when the request is received by the server, which is expressed as seconds since 1970.01.01 00:00:00 UTC, regardless of the leap seconds.
SecureTransport	Bool	Requests whether to use SSL.
SourceIp	IP address	Source IP address from which the request is sent
UserAgent	String	Requested client software agent
Referer	String	Indicates the link from which the request is sent.

Table 2-25 Keys related to bucket actions

Action	Optional Key	Description	Description
ListBucket	prefix	Type: String. Lists objects that begin with the specified prefix.	If **prefix**, **delimiter**, and **max-keys** are configured, the key-value pair meeting the conditions must be specified in the List operation for the bucket policy to take effect. For example, if a bucket policy (with the conditional operator set to **NumericEquals**, the key to **max-keys**, and the value to **100**) that allows anonymous users to read data is configured for a bucket, the anonymous users must add **?max-keys=100** to the end of the bucket domain name for listing objects. The listed objects are the first 100 objects in alphabetic order.
	max-keys	Type: Numeric. Sets the maximum number of objects. Returned objects are listed in alphabetic order.	
ListBucketVersions	prefix	Type: String. Lists multi-version objects whose name starts with the specified prefix.	
	max-keys	Type: Numeric. Sets the maximum number of objects. Returned objects are listed in alphabetic order.	

Action	Optional Key	Description	Description
PutBucketAcl	acl	Type: String. Configures the bucket ACL. The canned ACLs that can be included in the modified bucket ACL contain **private**, **public-read**, **public-read-write**, **authenticated-read**, **bucket-owner-read**, **bucket-owner-full-control**, and **log-delivery-write**.	None

Table 2-26 Keys related to object actions

Action	Optional Key	Description
PutObject	acl	Type: String. Configures the object ACL. When an object is uploaded, the canned ACLs that can be included in the object ACL contain **private**, **public-read**, **public-read-write**, **authenticated-read**, **bucket-owner-read**, **bucket-owner-full-control**, and **log-delivery-write**.
	copy-source	Type: String. Specifies names of the source bucket and the source object. Format: **/bucketname/keyname**
	metadata-directive	Type: String. Specifies whether to copy the metadata from the source object or replace with the metadata in the request. Values: **COPY| REPLACE**
PutObjectAcl	acl	Type: String. Configures the object ACL. When an object is uploaded, the canned ACLs that can be included in the object ACL contain **private**, **public-read**, **public-read-write**, **authenticated-read**, **bucket-owner-read**, **bucket-owner-full-control**, and **log-delivery-write**.
GetObjectVersion	versionId	Type: String. Obtains the object with the specified version ID.
GetObjectVersionAcl	versionId	Type: String. Obtains the ACL of the object with specified version ID.
PutObjectVersionAcl	versionId	Type: String. Specifies a version ID.

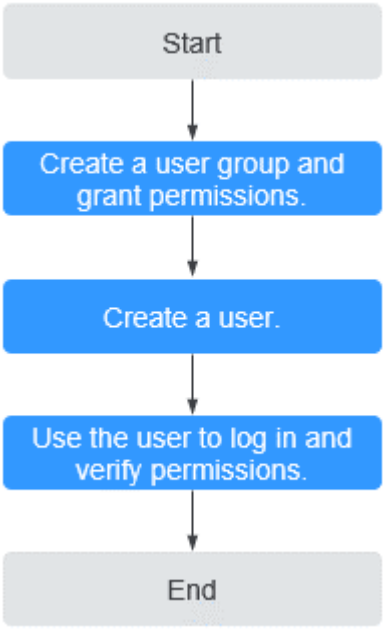
Action	Optional Key	Description
	acl	Type: String. Configures the ACL of the object with the specified version ID. When an object is uploaded, the canned ACLs that can be included in the object ACL contain **private**, **public-read**, **public-read-write**, **authenticated-read**, **bucket-owner-read**, **bucket-owner-full-control**, and **log-delivery-write**.
DeleteObjectVersion	versionId	Type: String. Deletes the object with the specified version ID.

2.11.4 Configuring IAM Policies

2.11.4.1 Creating an IAM User and Granting OBS Permissions

Process

Figure 2-9 Process of granting an IAM user the OBS permissions



Procedure

Step 1 Log in to the management console with your account.

Step 2 On the top menu bar, choose **Service List** > **Management & Deployment** > **Identity and Access Management**. The IAM console is displayed.

Step 3 Create a user group and assign OBS permissions to it.

A user group is a collection of users. By assigning permissions to a user group, you assign permissions to the users in this group. After you create an IAM user, add it to one or more user groups, so that it can inherit the permissions from the groups.

1. In the navigation pane, choose **User Groups**. The **User Groups** page is displayed.
2. Click **Create User Group**.
3. Enter a user group name and click **OK**.

   The user group is displayed in the user group list once the creation is complete.
4. Locate the user group you created and click **Modify** in the **Operation** column of the row.
5. In the **Group Permissions** area, locate the row that displays **Global service** > **OBS**, click **Attach Policy** in the **Operation** column, select the policy name, and click **OK**.

   NOTE

   In the **Policy Information** area, you can view the details about the policy.

   Due to data caching, an RBAC policy or a fine-grained policy involving OBS actions will take effect 10 to 15 minutes after it is attached to a user, an enterprise project, or a user group.

Step 4 Create an IAM user. For details, see section "Creating an IAM User" in the *Identity and Access Management User Guide*.

Step 5 Use the created IAM user to log in to OBS Console and verify the user permissions.

----End

2.11.4.2 Configuring Fine-Grained Policies

Custom policies can be created to supplement the system-defined policies of OBS. For the actions supported for custom policies, see **Bucket-Related Actions** and **Object-Related Actions**.

You can create custom policies in either of the following two ways:

- Visual editor: Select cloud services, actions, resources, and request conditions without the need to know policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**. The following provides examples of common OBS custom policies.

Example Custom Policies

- Example 1: Grant users all OBS permissions.

  This policy allows users to perform any operation on OBS.

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Allow",
```



```
            "Action": [
                "obs:*:*"
            ]
        }
    ]
}
```

- Example 2: Grant users all OBS Console permissions.

  This policy allows users to perform all operations on OBS Console.

  When a user logs in to OBS Console, the user may access resources of other services such as audit information in CTS. Therefore, in addition to the OBS permissions in example 1, you also need to configure the access permissions to other services. You need to configure the **Tenant Guest** permission for the global project and regional projects based on the services and regions that you use.

  ```
  {
      "Version": "1.1",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": [
                  "obs:*:*"
              ]
          }
      ]
  }
  ```

- Example 3: Grant users the read-only permission for all directories in a bucket.

  This policy allows users to list and download all objects in bucket **obs-example**.

  ```
  {
      "Version": "1.1",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": [
                  "obs:object:GetObject",
                  "obs:bucket:ListBucket"
              ],
              "Resource": [
                  "obs:*:*:object:obs-example/*",
                  "obs:*:*:bucket:obs-example"
              ]
          }
      ]
  }
  ```

- Example 4: Grant users the read-only permission for a specified directory in a bucket.

  This policy allows users to download objects in only the **my-project/** directory of bucket **obs-example**. Objects in other directories can be listed but cannot be downloaded.

  ```
  {
      "Version": "1.1",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": [
                  "obs:object:GetObject",
                  "obs:bucket:ListBucket"
              ],
              "Resource": [
                  "obs:*:*:object:obs-example/my-project/*",
                  "obs:*:*:bucket:obs-example"
              ]
  ```



```
        }
    ]
}
```

- Example 5: Grant users the read/write permissions for a specified directory in a bucket.

  This policy allows users to list, download, upload, and delete objects in the **my-project** directory of bucket **obs-example**.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "obs:object:GetObject",
                "obs:object:ListMultipartUploadParts",
                "obs:bucket:ListBucket",
                "obs:object:DeleteObject",
                "obs:object:PutObject"
            ],
            "Resource": [
                "obs:*:*:object:obs-example/my-project/*",
                "obs:*:*:bucket:obs-example"
            ]
        }
    ]
}
```

- Example 6: Grant users all permissions for a bucket.

  This policy allows users to perform any operation on bucket **obs-example**.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "obs:*:*"
            ],
            "Resource": [
                "obs:*:*:bucket:obs-example",
                "obs:*:*:object:obs-example/*"
            ]
        }
    ]
}
```

- Example 7: Grant users the permission to deny object upload.

  A deny policy must be used together with other policies. If the permissions assigned to a user contain both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

  If you grant the system policy OBS Operator to a user but do not want the user to have the object upload permission (which is also a permission allowed by OBS Operator), you can create a custom policy besides the OBS Operator policy, to deny the user's upload permission. According to the authorization principle, the policy with the deny statement takes precedence, so that the user can perform all operations allowed by OBS Operator, except uploading objects. The following is an example of a deny policy:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "obs:object:PutObject"
```



```
        ]
      }
    ]
}
```

2.11.4.3 OBS Resources

A resource is an object that exists within a service. OBS resources include buckets and objects. You can select these resources by specifying their paths.

Table 2-27 OBS resources and their paths

Resource Type	Resource Name	Path
Buckets	Bucket	[Format] **obs:*:*:bucket:***Bucket name***</br>[Notes] IAM automatically generates the prefix **obs:*:*:bucket:** for bucket resource paths. By adding *Bucket name* to the end of the generated prefix, you can define a specific path. An asterisk * is allowed to indicate any bucket. An example is given as follows: **obs:*:*:bucket:***
Objects	Object	[Format] **obs:*:*:object:***Bucket name*/*Object name***</br>[Notes] IAM automatically generates the prefix **obs:*:*:object:** for object resource paths. By adding *Bucket name/Object name* to the end of the generated prefix, you can define a specific path. An asterisk * is allowed to any object in the bucket. An example is given as follows: **obs:*:*:object:my-bucket/my-object/*** (indicating any object in the **my-object** directory of bucket **my-bucket**)

2.11.5 Configuring a Bucket Policy

2.11.5.1 Configuring a Standard Bucket Policy

For standard bucket policy, OBS offers three options, namely the Private, Public Read, and Public Read and Write policies. These policies are pre-defined and can be applied with a few clicks.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Permissions**.

Step 3 On the **Bucket Policies** tab page, select a policy from the **Standard Bucket Policies** area.

- **Private**: No access beyond the bucket ACL settings is granted.
- **Public Read**: Anyone can read objects in the bucket.
- **Public Read and Write**: Anyone can read, write, or delete objects in the bucket.

NOTE

For your data security, the **Public Read** and **Public Read and Write** policies are not recommended.

Step 4 In the dialog box that is displayed, click **Yes**.

----End

2.11.5.2 Configuring a Custom Bucket Policy

If you want to grant special permissions to specific users, you can configure custom bucket policies. If a standard bucket policy conflicts with a custom bucket policy, the authorization priority is given to the custom bucket policy and then the standard bucket policy.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Permissions**.

Step 3 On the **Bucket Policies** tab page, configure a custom bucket policy according to your needs.

Step 4 Click **Create Bucket Policy**. Select a proper policy mode as required. Valid values are as follows:

- **Read-only**: The authorized user will have the read permission on the bucket and objects. For subsequent operations, see **Step 5**.
- **Read and write**: The authorized user will have the read and write permissions on the bucket and objects. For subsequent operations, see **Step 5**.
- **Customized**: The authorized user will have the customized permissions on the bucket and objects. For detailed configuration, see **Step 6**.

NOTE

Only one bucket policy mode can be configured at a time.

Step 5 For the read-only and read and write modes, enter information about the authorized user in the following format and click **OK**.

Table 2-28 Parameters in bucket policies

Parameter	Value	Description
Principal	• **Include** or **Exclude** • **Current account** or **Other account**	Specifies users on whom this bucket policy takes effect. • **Include**: The policy applies to specified users. • **Exclude**: The policy applies to all users except the specified ones.
Principal	• **Include** or **Exclude** • **Current tenant** or **Other tenant**	The person the policy is applied to. • **Include**: The policy applies to specified users. • **Exclude**: The policy applies to all users except the specified ones.
Resources	• **Include** or **Exclude** • Input format: Object: *Object name* Object set: *Object name prefix**, **Object name suffix*, or *	Indicates the resource that a bucket policy applies to. With the read-only mode and read and write mode, the policy can only apply to objects. • **Include**: The policy takes effect on the specified OBS resources. • **Exclude**: The policy takes effect on all OBS resources except the specified ones.

Step 6 For the customized mode, set parameters based on the site requirements and click **OK**.

Table 2-29 describes each parameter.

Table 2-29 Parameters for configuring a custom bucket policy

Parameter	Value	Description
Effect	**Allow** or **Deny**	Effect of a bucket policy. • **Allow**: The policy allows the matched requests. • **Deny**: The policy denies the matched requests.
Principal	• **Include** or **Exclude** • **Current account** or **Other account**	Specifies users on whom this bucket policy takes effect. • **Include**: The policy applies to specified users. • **Exclude**: The policy applies to all users except the specified ones.

Parameter	Value	Description
Principal	• **Include** or **Exclude** • **Current tenant** or **Other tenant**	The person the policy is applied to. • **Include**: The policy applies to specified users. • **Exclude**: The policy applies to all users except the specified ones.
Resources	• **Include** or **Exclude** • **Specific resources**: Object: *Object name* Object set: *Object name prefix**, **Object name suffix*, or * • **Entire bucket**: The policy applies to the entire bucket.	Indicates the resource that a bucket policy applies to. • **Include**: The policy takes effect on the specified OBS resources. • **Exclude**: The policy takes effect on all OBS resources except the specified ones. Relationship between resource types and actions: • When a resource is an object or an object set, only the actions related to the object can be configured. • When the resource is a bucket, only the actions related to the bucket can be configured.
Actions	• **Include** or **Exclude** • For details, see **Actions**.	Operations stated in the bucket policy. • **Include**: The policy applies to specified actions. • **Exclude**: The policy takes effect on all actions except the specified ones.
Conditions	• **Conditional Operator**: See **Table 2-23**. • **Key**: See **Table 2-24**, **Table 2-25**, and **Table 2-26**. • **Value**: The entered value is associated with the key.	Conditions under which the bucket policy takes effect

----End

2.11.6 Configuring an Object Policy

Object policies are applied to the objects in a bucket. With an object policy, you can configure conditions and actions for objects in a bucket.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 On the right of the object to be operated, choose **More** > **Configure Object Policy**. The **Configure Object Policy** dialog box is displayed.

Step 4 Select a proper policy mode as required. Valid options are as follows:

- **Read-only**: The authorized user has the read permission on the object. For follow-up procedure, see **Step 5**.
- **Read and write**: The authorized user has the read and write permissions on the object. For follow-up procedure, see **Step 5**.
- **Customized**: The authorized user has the customized permissions on the object. For detailed configuration, see **Step 6**.

NOTE

You can configure only one object policy at a time.

Step 5 For read-only and read and write modes, enter information about the authorized user in the following format and click **OK**.

Table 2-30 Object policy parameters in read-only or read and write mode

Parameter	Value	Description
Principal	• **Include** or **Exclude** • **Current account** or **Other account**	Indicates the user that the object policy applies to. • **Include**: The policy applies to specified users. • **Exclude**: The policy applies to users except the specified ones.
Principal	• **Include** or **Exclude** • **Current tenant** or **Other tenant**	The person the object policy is applied to. • **Include**: The policy applies to specified users. • **Exclude**: The policy applies to users except the specified ones.
Resources	**Include** or **Exclude**	Resources on which the object policy takes effect. • **Include**: The bucket policy applies to specified OBS resources. • **Exclude**: The bucket policy applies to OBS resources except the specified ones.

Step 6 For the customized mode, set parameters based on the site requirements and click **OK**.

Table 2-31 Object policy parameters in the custom mode

Parameter	Value	Description
Effect	**Allow** or **Deny**	Effect of the object policy. • **Allow**: The policy allows the matched requests. • **Deny**: The policy denies the matched requests.
Principal	• **Include** or **Exclude** • **Current account** or **Other account**	Specifies users on whom this object policy takes effect. • **Include**: The policy applies to specified users. • **Exclude**: The policy applies to users except the specified ones.
Principal	• **Include** or **Exclude** • **Current tenant** or **Other tenant**	The person the object policy is applied to. • **Include**: The policy applies to specified users. • **Exclude**: The policy applies to users except the specified ones.
Resources	• **Include** or **Exclude**	Resources on which the object policy takes effect. • **Include**: The bucket policy applies to specified OBS resources. • **Exclude**: The bucket policy applies to OBS resources except the specified ones.
Actions	• **Include** or **Exclude** • For details about the actions, see **Actions Related to Objects**.	Operation stated in the object policy. • **Include**: The bucket policy applies to specified actions. • **Exclude**: The bucket policy applies to actions except the specified ones.
Conditions	• **Condition Operator**: See **Table 2-23**. • **Key**: See **Table 2-24** and **Table 2-26**. • **Value**: The entered value is associated with the key.	Condition for an object policy to take effect.

Step 7 Click **OK**.

After the object policy is configured successfully, it is displayed in the list under **Custom Bucket Policies**.

----End

2.11.7 Configuring a Bucket ACL

Prerequisites

You are the bucket owner or you have the permission to write the bucket ACL.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Permissions**.

Step 3 Under **Bucket ACLs**, click **Edit** to grant the owner, anonymous user, and log delivery user required permissions for the bucket.

Step 4 Click **Add** to apply specific ACL permissions to an account.

Enter an account ID or account name and specify ACL permissions for the account. You can obtain the account ID or account name from the **My Credentials** page.

Step 5 Click **Save**.

----End

2.11.8 Configuring an Object ACL

Prerequisites

You are the object owner or you have the permission to write the object ACL.

An object owner is the account that uploads the object, but may not be the owner of the bucket that stores the object. For example, account **B** is granted the permission to access a bucket of account **A**, and account **B** uploads a file to the bucket. In that case, account **B**, instead of the bucket owner account **A**, is the owner of the object. By default, account A is not allowed to access this object and cannot read or modify the object ACL.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 Click the object to be operated.

Step 4 On the **Object ACL** tab page, click **Edit** to grant the owner and anonymous user ACL permissions for the object.

NOTE

ACL permissions for encrypted objects cannot be granted to registered users or anonymous users.

Step 5 Click **Add** to apply specific ACL permissions to an account.

Enter an account ID or account name and specify ACL permissions for the account. You can obtain the account ID or account name from the **My Credentials** page.

Step 6 Click **Save**.

----End

2.11.9 Application Cases

2.11.9.1 Granting an IAM User Permissions to Operate a Specific Bucket

Create an IAM user under in an account. The IAM user has no permission to any resource before it is added to any user group. The bucket owner (root account) or other accounts and IAM users, who have the permission to set bucket policies, can configure bucket policies to grant the bucket operation permissions to IAM users.

The following is an example about how to grant an IAM user the bucket access and object upload permissions.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Permissions**.

Step 3 Choose **Bucket Policies** > **Custom Bucket Policies**.

Step 4 Click **Create Bucket Policy**.

Step 5 Configure parameters listed in the table below to grant IAM users the permissions to access the bucket (to list objects in the bucket). Retain the default values for the other parameters.

Table 2-32 Parameters for granting the object listing permission

Parameter	Value
Policy Mode	**Customized**
Effect	**Allow**
Principal	• **Include** • Select **Current account** and select the IAM user to be authorized.

Parameter	Value
Resources	• **Include** • Select **Entire bucket**.
Actions	• **Include** • ListBucket

Step 6 Click **OK**.

Step 7 Click **Create Bucket Policy**. The **Create Bucket Policy** dialog box is displayed.

Step 8 Configure parameters in the table below to grant an IAM user the permission to upload objects to a bucket.

NOTE

Before granting this permission to a user, ensure that the user has the permission to access the bucket.

Table 2-33 Parameters for granting the object upload permission

Parameter	Value
Policy Mode	**Customized**
Effect	**Allow**
Principal	• **Include** • Select **Current account** and select the IAM user to be authorized.
Resources	• **Include** • Select **Specific resources**. • Resource name: *****
Actions	• **Include** • PutObject **NOTE** In this example, only the permission to upload objects is granted. You can also select other object actions to grant corresponding permissions if needed. The asterisk (*) indicates all actions. For details about the supported actions, see **Actions**.

Step 9 Click **OK**.

----End

2.11.9.2 Granting Other Accounts Permissions to Operate a Specific Bucket

The bucket owner (root account) or other accounts and IAM users, who have the permission to set bucket policies, can configure bucket policies to grant the bucket operation permissions to other accounts or IAM users under other accounts.

The following is an example about how to grant other accounts bucket access and object upload permissions.

NOTE

To grant permissions to IAM users under other accounts, you need to configure both bucket policies and IAM policies.

1. Configure a bucket policy to allow IAM users to access the bucket.
2. Configure IAM policies for the account where authorized IAM users belong, to allow the IAM users to access the bucket.

Only permissions that are allowed by both the bucket policy and IAM policies can take effect.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Permissions**.

Step 3 Choose **Bucket Policies** > **Custom Bucket Policies**.

Step 4 Click **Create Bucket Policy**.

Step 5 Configure the parameters listed in the table below to grant other accounts the bucket access permission. Retain the default values for the other parameters.

Table 2-34 Parameters for granting the object listing permission

Parameter	Value
Policy Mode	**Customized**
Effect	**Allow**
Principal	• **Include** • Select **Other account**. Enter the account ID and user ID. **NOTE** The account ID and user ID can be obtained on the **My Credentials** page. If you grant the permission only to the account itself, IAM user IDs are not required. If you grant the permission to one or more IAM users under the account, configure both the account ID and IAM user IDs. Use commas (,) to separate multiple IAM user IDs.

Parameter	Value
Resources	• **Include** • Select **Entire bucket**.
Actions	• **Include** • ListBucket

Step 6 Click **OK**.

Step 7 Click **Create Bucket Policy**. The **Create Bucket Policy** dialog box is displayed.

Step 8 Configure the parameters listed in the table below to grant other accounts the object upload permission:

NOTE

Before granting this permission to a user, ensure that the user has the permission to access the bucket.

Table 2-35 Parameters for granting the object upload permission

Parameter	Value
Policy Mode	**Customized**
Effect	**Allow**
Principal	• **Include** • Select **Other account**. Enter the account ID and user ID. **NOTE** The account ID and user ID can be obtained on the **My Credentials** page. If you grant the permission only to the account itself, IAM user IDs are not required. If you grant the permission to one or more IAM users under the account, configure both the account ID and IAM user IDs. Use commas (,) to separate multiple IAM user IDs.
Resources	• **Include** • Select **Specific resources**. • Resource name: *****
Actions	• **Include** • PutObject

Step 9 Click **OK**.

----End

2.11.9.3 Restricting Access to a Bucket for Specific Addresses

You can configure a bucket policy to restrict access to a bucket for specific addresses. This example describes how to deny access from clients whose IP address is in the range of **114.115.1.0/24** to a bucket.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Permissions**.

Step 3 Choose **Bucket Policies** > **Custom Bucket Policies**.

Step 4 Click **Create Bucket Policy**.

Step 5 Configure parameters listed in the table below.

Table 2-36 Restricting access to a bucket for specific addresses

Parameter	Value
Policy Mode	**Customized**
Effect	Deny
Principal	• **Include** > **Other account** • If the account ID is set to *****, the policy setting takes effect on all anonymous users. • Leave the user ID blank.
Resources	• **Include** • Select **Entire bucket**.
Actions	• **Include** • Select the asterisk (*), indicating all actions are involved.
Conditions	• **Conditional Operator**: **IpAddress** • **Key**: **SourceIP** • **Value**: **114.115.1.0/24**

Step 6 Click **OK**.

----End

Verification

Initiate an access request from an IP address in the range of **114.115.1.0/24**. The access is denied. Initiate an access request from an IP address beyond the range of **114.115.1.0/24**. The access is allowed.

2.11.9.4 Limiting the Time When Objects in a Bucket Are Accessible

You can configure the bucket policy to limit the time when objects in a bucket are accessible. In the following example, the access time window is from 2019-03-26T12:00:00Z to 2019-03-26T15:00:00Z.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Permissions**.

Step 3 Choose **Bucket Policies** > **Custom Bucket Policies**.

Step 4 Click **Create Bucket Policy**. The **Create Bucket Policy** dialog box is displayed.

Step 5 Configure parameters listed in the table below.

Table 2-37 Parameters for granting permission to access a bucket

Parameter	Value
Policy Mode	**Customized**
Effect	**Allow**
Principal	• **Include** • Select **Other account**, and enter an asterisk (*) as the account ID, indicating all anonymous users.
Resources	• Choose **Include** > **Specific resources**. • Set the resource name to *****, indicating all resources in the bucket. **NOTE** This example only grants permissions for resources in the bucket. If you also want to grant permission for the bucket (for example, the permission to list objects in the bucket), create another custom bucket policy.
Actions	• **Include** • Select ***** as the action name, which indicates all action permissions. **NOTE** Selecting ***** may cause resources to be deleted. To avoid this risk, select **Get*** that indicates all read permissions.

Parameter	Value
Conditions	• **Condition Operator**: Select **DateGreaterThan**. • **Key**: Select **CurrentTime**. • **Value**: Enter **2019-03-26T12:00:00Z** (UTC).
Conditions	• **Condition Operator**: Select **DateLessThan**. • **Key**: Select **CurrentTime**. • **Value**: Enter **2019-03-26T15:00:00Z** (UTC).

NOTE

The preceding two conditions must be configured in the same bucket policy.

Step 6 Click **OK**.

----End

Verification

During the specified time period, any user can access the specified resources in the bucket. Outside the specified time period, only the bucket owner can access the bucket.

2.11.9.5 Granting Anonymous Users Permission to Access Objects

An enterprise stores a large volume of file data in OBS, and offers the data for public query. This enterprise sets a read permission for anonymous users, and provides the data URLs on the Internet. Then all users can read or download the data through the URLs.

Procedure

Step 1 Log in to OBS Console and click **Create Bucket** to create a bucket.

Step 2 In the bucket list, click the name of the newly created bucket. On the displayed object management page, upload the file data to the new bucket. The file data is stored as an object.

Step 3 Click the object name. The object details page is displayed.

Step 4 Under **Object ACL** > **Public Permissions**, click **Edit** to grant the object read permission to anonymous users.

Step 5 Click **Save** to save the permission setting.

----End

Verification

Step 1 Click the object. Its URL is displayed under **Link**. Share the URL over the Internet, so that all users can access or download the object through the Internet.

Step 2 An anonymous user can view the object by copying the URL of the object to the web browser.

----End

2.11.9.6 Granting Anonymous Users Permission to Access Folders

If all objects in a folder need to be accessible to anonymous users, you can configure a bucket policy or an object policy to grant anonymous users the permission to access the folder. In this example, a bucket policy is used. If you want to use an object policy to grant permission, select the target folder and configure an object policy. Parameters in both types of policies are the same.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Permissions**.

Step 3 Choose **Bucket Policies** > **Custom Bucket Policies**.

Step 4 Click **Create Bucket Policy**.

Step 5 Configure parameters according to the following table, so that you can grant anonymous users the permission to access the folder and objects in it. Retain the default values for the other parameters.

Table 2-38 Parameters for granting permission to access a bucket

Parameter	Value
Policy Mode	**Customized**
Effect	**Allow**
Principal	• **Include** • Select **Other account**, and enter an asterisk (*) as the account ID, indicating all anonymous users.
Resources	• **Include** • Select **Specific resources**. • Set this parameter to all objects in the selected folder. If the folder name is **folder-001**, enter the value **folder-001/***.
Actions	• **Include** • GetObject

Step 6 Click **OK**.

----End

Verification

Step 1 After the permission is successfully configured, select an object in the folder and click the object name to view its details. The object link (URL) is displayed on the details page. Share the URL over the Internet, so that all users can access or download the object through the Internet.

Step 2 Use the URL to access the object in a browser. An anonymous user can access the object.

----End

2.12 Versioning

2.12.1 Versioning Overview

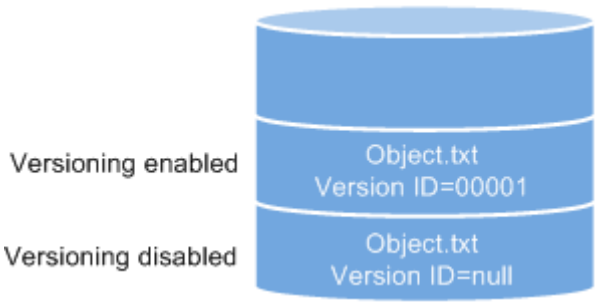
OBS can store multiple versions of an object. You can quickly search for and restore different versions or restore data in the event of accidental deletions or application faults.

By default, the versioning function is disabled for new buckets on OBS. Therefore, if you upload an object to a bucket where an object with the same name exists, the new object will overwrite the existing one.

Enabling Versioning

- Enabling versioning does not change the versions and contents of existing objects in the bucket. The version ID of an object is **null** before versioning is enabled. If a namesake object is uploaded after versioning is enabled, a version ID will be assigned to the object. For details, see **Figure 2-10**.

  Figure 2-10 Versioning (with existing objects)

- OBS automatically allocates a unique version ID to a newly uploaded object. Objects with the same name are stored in OBS with different version IDs.

Figure 2-11 Versioning (for new objects)

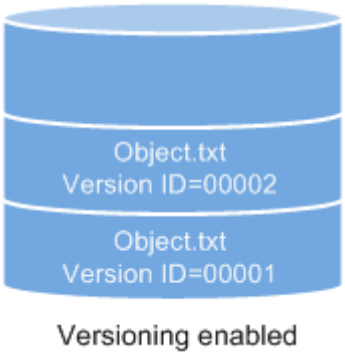
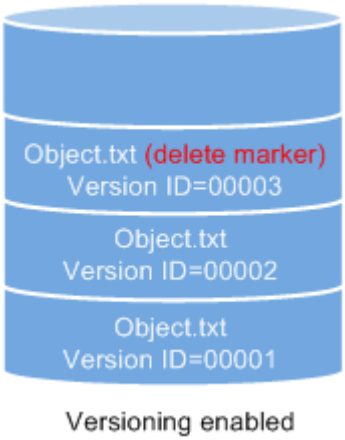


Table 2-39 Version description

Version	Description
Latest version	After versioning is enabled, each operation on an object will result in saving of the object with a new version ID. The version ID generated upon the latest operation is called the latest version.
Historical version	After versioning is enabled, each operation on an object will result in saving of the object with a new version ID. Version IDs generated upon operations other than the latest operation are called historical versions.

- The latest objects in a bucket are returned by default after a GET Object request.
- Objects can be downloaded by version IDs. By default, the latest object is downloaded if the version ID is not specified. For details, see **Related Operations** in **Configuring Versioning**.
- You can select an object and click **Delete** on the right to delete the object. After the object is deleted, OBS generates a **Delete Marker** with a unique version ID for the deleted object, and the deleted object is displayed in the **Deleted Objects** list. For details, see **Deleting an Object or Folder**. If attempts are then made to access this deleted object, error 404 will be returned.

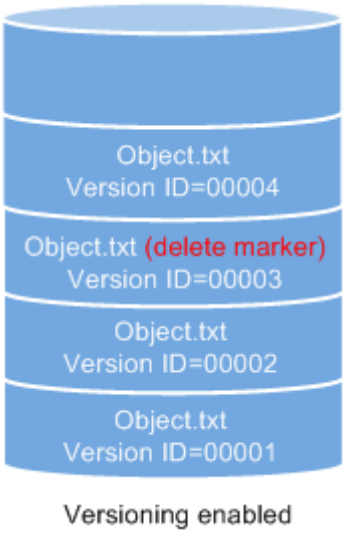
Figure 2-12 Object with a delete marker



- You can recover a deleted object by deleting the delete marker. For details, see **Related Operations** in **Undeleting an Object**.
- After an object is deleted, you can specify the version number in **Deleted Objects** to permanently delete the object of the specified version. For details, see **Related Operations** in **Deleting an Object or Folder**.
- An object appears in either the object list or the list of deleted objects. It will never appear in both lists at the same time.

  For example, after object **A** is deleted, it will appear in the **Deleted Objects** list. If you later upload another object with the same name **A**, the new object **A** will appear in the **Objects** list, but the previously deleted object **A** will disappear from the **Deleted Objects** list. For details, see **Figure 2-13**.

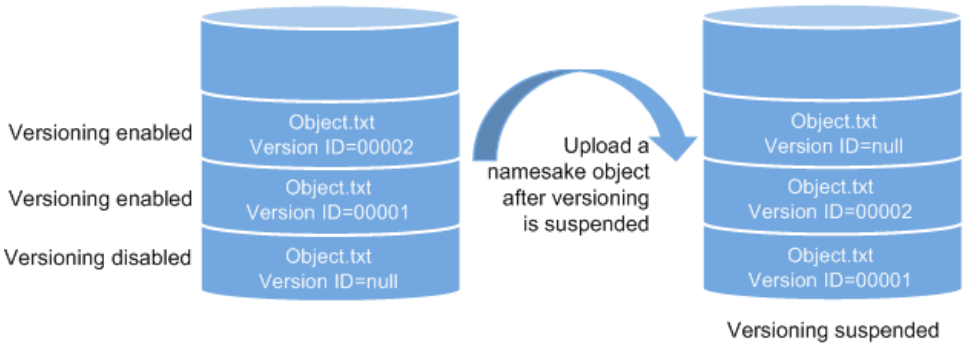
Figure 2-13 Uploading a namesake object after the original one is deleted



Suspending Versioning

Once versioning is enabled for a bucket, it cannot be disabled, but it can be suspended. When versioning is suspended, a null, not a specific version ID, will be allocated to a newly uploaded object. If the newly uploaded object has the same name as an existing object with a null version ID, the new object will overwrite the existing object.

Figure 2-14 Object versions in the scenario when versioning is suspended



If versioning is no longer needed, you can suspend it. After versioning is suspended:

- Existing object versions are still retained in OBS. If you no longer desire these versions, manually delete them.
- Objects can be downloaded by version IDs. By default, the latest object is downloaded if the version ID is not specified.

Differences Between Scenarios When Versioning Is Suspended and Disabled

If you delete an object after versioning is suspended for the bucket, a delete marker will be generated, no matter whether the object has historical versions. But, if versioning is disabled, the same operation will not generate a delete marker.

2.12.2 Configuring Versioning

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the **Basic Information** area, move the cursor over **Disabled**, **Suspended**, or **Enabled** next to **Versioning**. The **Edit** button is displayed next to the versioning status. Click **Edit**. The dialog box for editing the versioning status is displayed.

Step 3 Select **Enable**.

Step 4 Click **OK** to enable versioning for the bucket.

Step 5 Click an object to go to the object details page. On the **Versions** tab page, view all versions of the object.

----End

Related Operations

After versioning is configured for a bucket, you can go to the object details page, click the **Versions** tab, and then delete and download object versions, and extend the retention period of an object version.

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Objects**.

Step 3 In the object list, click the object you want to go to the object details page.

Step 4 On the **Versions** tab page, view all versions of the object.

Step 5 Perform the following operations on object versions:

1. Download a desired version of the object by clicking **Download** in the **Operation** column.

   NOTE

   If the version you want to download is in the Cold storage class, restore it first.

2. Permanently delete a version of the object by choosing **Delete** in the **Operation** column. The deleted object version cannot be recovered. If you delete the latest version, the most recent version will become the latest version.

   NOTE

   In a WORM-enabled bucket, if an object has no retention policy configured or its retention policy has expired, you can delete a desired object version on the object's **Versions** tab page. If an object version is within the retention period, it cannot be deleted.

3. Locate the object version for which you want to extend the retention period, choose **Extend Retention Period**, and select a date. A retention period can be extended, but cannot be shortened.

----End

2.13 Logging

2.13.1 Logging Overview

You can enable logging to facilitate analysis or audit. Access logs enable a bucket owner to analyze the property, type, or trend of requests to the bucket in depth. When the logging function of a bucket is enabled, OBS will log access requests for the bucket automatically, and write the generated log files to the specified bucket (target bucket).

After logging is enabled, the log delivery user group will be automatically granted the permission to read the bucket ACL and write the bucket where logs are saved. If you manually disable such permissions, bucket logging will fail.

OBS can log bucket access requests for further request analysis or log audit.

Logs occupy the OBS storage that incurs costs, so OBS does not collect bucket access logs by default.

OBS creates log files and uploads them to a specified bucket. To perform these operations, OBS must be granted required permissions. Therefore, before configuring logging for a bucket, you need to create an IAM agency for OBS and add this agency when configuring logging for the bucket. By default, when

configuring permissions for an agency, you only need to grant the agency the permission to upload log files (PutObject) to the bucket for storing log files. In the following example, **mybucketlogs** is the bucket. If the log storage bucket has default encryption enabled, the agency also requires the **KMS Administrator** permission for the region where the bucket is located.

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

After logging is configured, you can view operation logs in the bucket that stores the logs in approximately fifteen minutes.

The following shows an example access log of the target bucket:

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

The access log of each bucket contains the following information.

Table 2-40 Bucket log format

Parameter	Value Example	Description
BucketOwner	787f2f92b20943998a4fe2ab75eb09b8	Account ID of the bucket owner
Bucket	bucket	Name of the bucket
Time	[13/Aug/2015:01:43:42 +0000]	Timestamp of the request (UTC)
Remote IP	xx.xx.xx.xx	IP address from where the request is initiated
Requester	787f2f92b20943998a4fe2ab75eb09b8	Requester ID
RequestID	281599BACAD9376ECE141B842B94535B	Request ID
Operation	REST.GET.BUCKET.LOCATION	Name of the operation
Key	-	Object name

Parameter	Value Example	Description
Request-URI	GET /bucket?location HTTP/1.1	URI of the request
HTTPStatus	200	Return code
ErrorCode	-	Error code
BytesSent	211	Size of the HTTP response, expressed in bytes
ObjectSize	-	Object size (bytes)
TotalTime	6	Processing time on the server (ms)
Turn-AroundTime	6	Total time for processing the request (ms)
Referer	-	Header field **Referer** of the request
User-Agent	HttpClient	User-Agent header of the request
VersionID	-	Version ID carried in the request
STSLogUrn	-	Federated authentication and agency information
StorageClass	STANDARD_IA	Current storage class of the object
TargetStorageClass	GLACIER	Storage class that the object will be transited to

2.13.2 Configuring Access Logging for a Bucket

After logging is enabled for a bucket, OBS automatically converts bucket logs into objects following the naming rules and writes the objects into a target bucket.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the **Basic Configurations** area, click **Logging**. The **Logging** dialog box is displayed.

Step 3 Select **Enable**.

Step 4 Select an existing bucket where you want to store log files. Log delivery users of the selected bucket will be automatically granted the permissions to read the bucket ACL and write logs to the bucket.

Step 5 Enter a prefix for the **Log File Name Prefix**.

After logging is enabled, generated logs are named in the following format:

*<Log File Name Prefix>*YYYY-mm-DD-HH-MM-SS-*<UniqueString>*

- *<Log File Name Prefix>* is the shared prefix of log file names.
- **YYYY-mm-DD-HH-MM-SS** indicates when the log is generated.
- *<UniqueString>* indicates a character string generated by OBS.

On OBS Console, if the configured *<Log File Name Prefix>* ends with a slash (/), logs generated in the bucket are stored in the *<Log File Name Prefix>* folder in the bucket, facilitating the management of log files.

Example:

- If the bucket named **bucket** is used to save log files, and the log file name prefix is set to **bucket-log/**, all log files delivered to this bucket are saved in the **bucket-log** folder. A log file is named as follows: **2015-06-29-12-22-07-N7MXLAF1BDG7MPDV**.
- If the bucket named **bucket** is used to save log files, and the log file name prefix is set to **bucket-log**, all log files are saved in the root directory of the bucket. A log file is named as follows: **bucket-log2015-06-29-12-22-07-N7MXLAF1BDG7MPDV**.

Step 6 Select an IAM agency to grant OBS the permission to upload log files to the specified bucket.

By default, when configuring permissions for an agency, you only need to grant the agency the permission to upload log files (PutObject) to the bucket for storing log files. In the following example, **mybucketlogs** is the bucket. If the log storage bucket has default encryption enabled, the agency also requires the **KMS Administrator** permission for the region where the bucket is located.

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

You can choose an existing IAM agency from the drop-down list or click **Create Agency** to create one. For details about how to create an agency, see **Creating an Agency**.

Step 7 Click **OK**.

After logging is configured, you can view operation logs in the bucket that stores the logs in approximately fifteen minutes.

----End

Related Operations

If you do not need to record logs, in the **Logging** dialog box, select **Disable** and then click **OK**. After logging is disabled, logs are not recorded, but existing logs in the target bucket will be retained.

2.14 Tags

2.14.1 Tag Overview

Tags are used to identify and classify OBS buckets.

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

A tag is described using a key-value pair. A bucket can have a maximum of 10 tags. Each tag has only one key and one value.

The key and value can exist in either sequence in a tag. Each key is unique among all tags of a bucket, whereas values can be repetitive or blank.

2.14.2 Adding Tags to a Bucket

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

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the **Basic Configurations** area, click **Tags**.

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

Step 3 Click **Add Tag**. The **Add Tag** dialog box is displayed.

Step 4 Set the key and value based on **Table 2-41**.

Table 2-41 Parameter description

Parameter	Description
Tag key	Key of a tag. Tag keys for the same bucket must be unique. You can customize tags or select the ones predefined on TMS. A tag key: • Must contain 1 to 36 characters and be case sensitive. • Cannot start or end with a space or contain the following characters: =*<>\,|/
Tag value	Value of a tag. A tag value can be repetitive or left blank. A tag value: • Can contain 0 to 43 characters and must be case sensitive. • Cannot contain the following characters: =*<>\,|/

Step 5 Click **OK**.

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

----End

Related Operations

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

2.15 Event Notifications

2.15.1 SMN-Enabled Event Notifications

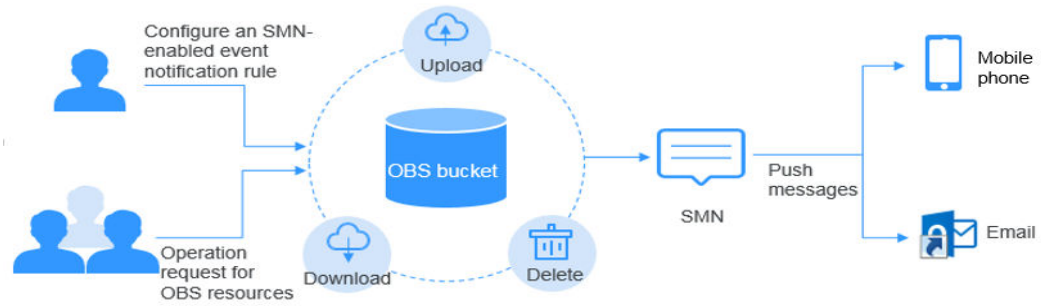
Simple Message Notification (SMN) is a reliable and extensible message notification service that can handle a huge number of messages. It significantly simplifies system coupling and can automatically push messages to endpoints via email.

OBS leverages SMN to provide event notifications. In OBS, you can use SMN to send event notifications to specified subscribers, so that you will be informed of any critical operations (such as upload and deletion) that occur on specified buckets in real time. For example, you can configure an event notification rule to send messages through SMN to the specified email address whenever an upload operation occurs on the specified bucket.

You can configure the event notification rule to filter objects by the object name prefix or suffix. For example, you can add an event notification rule to send notifications whenever an object with the **.jpg** suffix is uploaded to the specified bucket. You can also add an event notification rule to send notifications whenever an object with the **images/** prefix is uploaded to the specified bucket.

For details about events supported by SMN and how to configure an SMN-enabled event notification rule, see **Configuring SMN-Enabled Event Notification**.

Figure 2-15 SMN-enabled event notification



2.15.2 Configuring SMN-Enabled Event Notification

This section describes how to configure an SMN-enabled event notification rule on OBS Console.

Background Information

For details, see **SMN-Enabled Event Notifications**.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the **Basic Configurations** area, click **Event Notification**. The **Event Notification** page is displayed.

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

Step 3 Click **Create**. The **Create Event Notification** dialog box is displayed.

Step 4 Configure event notification parameters, as described in **Table 2-42**.

Table 2-42 Event notification parameters

Parameter	Description
Name	Name of the event. If the event name is left blank, the system will automatically assign a globally unique ID.

Parameter	Description
Events	Various types of events. Currently, OBS supports event notification for the following types of events: • **ObjectCreated**: all kinds of object creation operations, including PUT, POST, COPY, and part assembling – **Put**: Creates or overwrites an object using the PUT method. – **Post**: Creates or overwrites an object using the POST (browser-based upload) method. – **Copy**: Creates or overwrites an object using the COPY method. – **CompleteMultipartUpload**: Assembles parts of a multipart upload. • **ObjectRemoved**: Deletes an object. – **Delete**: Deletes an object with a specified version ID. – **DeleteMarkerCreated**: Deletes an object without specifying a version ID. Multiple event types can be applied to the same object. For example, if you have selected **Put**, **Copy**, and **Delete** in the same event notification rule, a notification will be sent to you when the specified object is uploaded to, copied to, or deleted from the bucket. **ObjectCreated** contains **Put**, **Post**, **Copy**, and **CompleteMultipartUpload**. If you select **ObjectCreated**, the events **ObjectCreated** contains are automatically selected. Similarly, if you select **ObjectRemoved**, **Delete** and **DeleteMarkerCreated** are automatically selected.
Prefix	Object name prefix for which notifications will be triggered. **NOTE** If neither the **Prefix** nor the **Suffix** is configured, the event notification rule applies to all objects in the bucket.
Suffix	Object name suffix for which notifications will be triggered. **NOTE** • A folder path ends with a slash (/). Therefore, if you want to configure event notification rules for operations on folders and you need to filter folders by suffix, the suffix must also end with a slash (/). • If neither the **Prefix** nor the **Suffix** is configured, the event notification rule applies to all objects in the bucket.

Parameter	Description
SMN Topic	Project: The project that contains the SMN topic you want to select. Projects are used to manage and classify cloud resources, including SMN topics. Each project contains different SMN topics. Select a project first and then a topic.
	Topic: specifies the SMN topic that authorizes OBS to publish messages. You can create such topics on the SMN management console. **NOTE** • Once SMN topics are selected for pushing OBS event notifications, do not delete them or cancel their authorizations to OBS. • If the topics are deleted or their authorizations to OBS are canceled, the following conditions may occur: a. The subscriber of the topic cannot receive messages. b. Event notifications associated with unavailable topics are automatically cleared. • For details about how to use SMN, see sections "Creating a Topic", "Adding a Subscription", and "Configuring Topic Policies" in the *Simple Message Notification User Guide*.

Step 5 Click **OK**.

----End

Related Operations

You can click **Edit** in the **Operation** column of an event notification rule, to edit the notification rule, or click **Delete** to delete the rule.

If you want to batch delete event notification rules, select the rules to delete and click **Delete** above the list.

2.15.3 Application Example: Configuring SMN-Enabled Event Notification

Background Information

An enterprise has a large number of files to archive but it does not want to cost much on storage resources. Therefore, the enterprise subscribes to OBS for storing daily files and expects that an employee can be informed of every operation performed on OBS via email.

Procedure

Step 1 Log in to OBS Console as an enterprise user.

Step 2 Create a bucket.

Click **Create Bucket** in the upper right corner of the page. On the page, select a region and storage class, and specify a bucket name and other parameters. Then, click **Create Now**.

Step 3 Create a folder.

Click the name of the bucket created in **Step 2** to go to the **Overview** page. Then, choose **Objects** and click **Create Folder**. In the displayed dialog box, enter a folder name and click **OK**. In the following example, **SMN** is the folder name.

Step 4 In the upper left corner of the page, click ☰ and choose **Simple Message Notification**. On the displayed SMN page, create a topic.

In the following example, **TestTopic** is the SMN topic and the notifications are sent via email.

Use SMN to create a notification topic for OBS as follows:

1. Create an SMN topic.
2. Add a subscription.
3. Modify the topic policy. On the **Configure Topic Policy** page, select **OBS** under **Services that can publish messages to this topic**.

For details, see **Table 2-42**.

Step 5 Go back to OBS Console.

Step 6 Configure an event notification rule.

1. In the bucket list, click the bucket that you have created in **Step 2**.
2. In the navigation pane, choose **Basic Configurations** > **Event Notification**. The **Event Notification** page is displayed.
3. Click **Create**. The **Create Event Notification** dialog box is displayed.
4. Configure event notification parameters.

   After the notification is configured, an employee will be informed of all specified operations on the **SMN** folder in bucket **testbucket**.

Table 2-43 Event notification parameters

Parameter	Value
Name	test
Events	ObjectCreated, ObjectRemoved

Parameter	Value
Prefix	SMN/ **NOTE** – A folder path ends with a slash (/). Therefore, if you want to configure event notification rules for operations on folders and you need to filter folders by suffix, the suffix must also end with a slash (/). – If neither the **Prefix** nor the **Suffix** is configured, the event notification rule applies to all objects in the bucket.
Notification Method	SMN topic *Select the project to which the SMN topic belongs.* TestTopic

----End

Verification

Step 1 Log in to OBS Console as an enterprise user.

Step 2 Upload the **test.txt** file to the folder created in **Step 3**.

After the file is uploaded, an employee receives an email. Keyword **ObjectCreated:Post** in the email indicates that the object is successfully uploaded.

Step 3 Delete the **test.txt** file uploaded in **Step 2**.

After the file is successfully deleted, an employee will receive an email. Keyword **ObjectRemoved:Delete** in the email indicates that the object is successfully deleted.

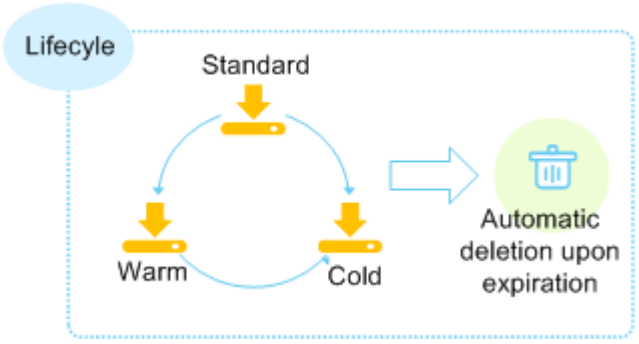
----End

2.16 Lifecycle Management

2.16.1 Lifecycle Management Overview

Lifecycle management means periodically deleting objects in a bucket or transitioning between object storage classes by configuring rules.

Figure 2-16 Lifecycle management



You may configure lifecycle rules to:

- Periodically delete logs that are only meant to be retained for a specific period of time (a week or a month).
- Transition documents that are seldom accessed to the Warm or Cold storage class or delete them.

You can define lifecycle rules for your scenarios similar to those mentioned above to better manage your objects.

You can configure lifecycle rules for objects that will no longer be frequently accessed to transition them to the Warm or Cold storage class as needed. This can help reduce costs on storage. In short, transition basically means that the object storage class is altered without copying the object. You can also manually change the storage class of an object on the Objects page. For details, see **Uploading an Object**.

Lifecycle rules have the following key elements:

- Policy

  You can specify an object name prefix to apply a lifecycle rule to a set of objects. You can also apply a lifecycle rule to the entire bucket (including the objects in it).
- Time

  You can specify the number of days after which objects that have been last updated and meet specified conditions are automatically transitioned to Warm or Cold, or are expired and then deleted.
  - Transition to Warm: This defines the number of days since the last object update after which objects meeting specified conditions are automatically transitioned to the Warm storage class.
  - Transition to Cold: This defines the number of days since the last object update after which objects meeting specified conditions are automatically transitioned to the Cold storage class.
  - Expiration time: This defines the number of days since the last object update after which objects meeting specified conditions are automatically expired and then deleted.

Objects can be transitioned to Warm at least 30 days after their last update. If you configure to transition objects first to Warm and then Cold, the objects must stay Warm at least 30 days before they can be transitioned to Cold. For example, if you configure to transition objects to Warm 33 days after their last update, the objects

can be transitioned to Cold at least 63 days after their last update. If transition to Cold is used, but transition to Warm is not, there is no limit on the number of days for transition. The number set for expiration time must be larger than that specified for any of the transition operations.

2.16.2 Configuring a Lifecycle Rule

You can configure a lifecycle rule for a bucket or a set of objects to:

- Transition objects from Standard to Warm or Cold.
- Transition objects from Warm to Cold.
- Expire objects and then delete them.

Lifecycle rules do not transition Cold objects to other storage classes.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the **Basic Configurations** area, click **Lifecycle Rules**. The **Lifecycle Rules** page is displayed.

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

Step 3 Click **Create**.

Step 4 Configure a lifecycle rule.

Basic Information:

- **Status**:

  Select **Enable** to enable the lifecycle rule.
- **Rule Name**:

  It identifies a lifecycle rule. A rule name can contain a maximum of 255 characters.
- **Applies To**: Can be set to **Object name prefix** or **Bucket**.
  - **Object name prefix**: Objects with this specified prefix will be managed by the lifecycle rule. The prefix cannot start with a slash (/) or contain two consecutive slashes (//), and cannot contain the following special characters: \ : * ? " < > |
  - **Bucket**: All objects in the bucket will be managed by the lifecycle rule.

NOTE

- If the specified prefix overlaps with the prefix of an existing lifecycle rule, OBS regards these two rules as one and forbids you to configure the one you are configuring. For example, if there is already a rule with prefix **abc** in OBS, you cannot configure another rule whose prefix starts with **abc**.
- If there is already a lifecycle rule based on an object prefix, you are not allowed to configure another rule that is applied to the entire bucket.
- If a lifecycle rule has been configured for the entire bucket, no more rules that apply to object name prefix can be added.

Current Version or **Historical Version**:

- If **Versioning** is not enabled for a bucket, only **Current Version** can be configured and **Historical Version** is not displayed by default.
- If **Versioning** was ever enabled for a bucket, both **Current Version** and **Historical Version** can be configured.

NOTE

- **Current Version** and **Historical Version** are two concepts for versioning. If versioning is enabled for a bucket, uploading objects with the same name to the bucket creates different object versions. The last uploaded object is called the current version, while those previously uploaded are called historical versions.
- You can configure either the **Current Version** or **Historical Version**, or both of them.

- **Transition to Warm**: After this number of days since the last update, objects meeting specified conditions will be transitioned to Warm. This number must be at least 30.
- **Transition to Cold**: After this number of days since the last update, objects meeting specified conditions will be transitioned to Cold. If you configure to transition objects first to Warm and then Cold, the objects must stay Warm at least 30 days before they can be transitioned to Cold. If transition to Cold is used, but transition to Warm is not, there is no limit on the number of days for transition.
- **Delete Objects After (Days)**: After this number of days since the last update, objects meeting certain conditions will be expired and then deleted. This number must be an integer larger than that specified for any of the transition operations.

For example, on January 7, 2015, you saved the following files in OBS:

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

On January 10, 2015, you saved another four files:

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

On January 10, 2015, you set the objects prefixed with **log** to expire one day later. You might encounter the following situations:

- Objects **log/test1.log** and **log/test2.log** uploaded on January 7, 2015 might be deleted after the last system scan. The deletion could happen on January 10, 2015 or January 11, 2015, depending on the time of the last system scan.
- Objects **log/clientlog.log** and **log/serverlog.log** uploaded on January 10, 2015 might be deleted on January 11, 2015 or January 12, 2015, depending on whether they have been stored for over one day (since their last update) when the system scan happened.

On the day of operation, you can set the objects with the name prefix **log** to be transitioned to **Warm** 30 days later, transitioned to **Cold** 60 days later, and deleted 100 days later, then OBS will transition **log/clientlog.log**, **log/serverlog.log**, **log/**

test1.log, and **log/test2.log** to **Warm** when their storage duration exceeds 30 days, transition them to **Cold** when their storage duration exceeds 60 days, and delete them when their storage duration exceeds 100 days, respectively.

NOTE

In theory, it takes 24 hours at most to execute a lifecycle rule. After an object is updated, OBS calculates its lifecycle from the next 00:00 (UTC time), so there may be a delay of up to 48 hours in transitioning objects between storage classes or deleting expired objects. If you make changes to an existing lifecycle rule, the rule will take effect again.

Step 5 Click **OK** to complete the lifecycle rule configuration.

----End

Follow-up Procedure

You can click **Edit**, **Delete**, or **Disable** (or **Enable**) in the **Operation** column of a lifecycle rule to edit, delete, disable (or enable) the rule.

You can also select multiple lifecycle rules at a time and click **Delete** or **Disable** (or **Enable**) above the list to batch delete or disable (or enable) them.

2.17 Configuring User-Defined Domain Names

2.17.1 Overview

Application Scenario

After you upload a file to a bucket, you can access this file using the bucket's access domain name by default. If you want to use a custom domain name to access the file, bind the custom domain name to the bucket.

Assume that you have a domain name **www.example.com** and you upload an image **image.png** to an OBS bucket. As long as you bind **www.example.com** to the bucket, you can use **http://www.example.com/image.png** to access **image.png**. The steps below describe the configurations:

1. Create a bucket on OBS and upload file **image.png** to the bucket.
2. On OBS Console, bind **www.example.com** to the created bucket.
3. On the DNS server, add a CNAME record and map **www.example.com** to the domain name of the bucket.
4. Send a request for image **image.png**. After the request for **http://www.example.com/image.png** reaches OBS, OBS finds the mapping between the **www.example.com** and the bucket's domain name, and redirects the request to the **image.png** file stored in the bucket. This way, a request for **http://www.example.com/image.png** actually accesses **http://***Bucket domain name***/image.png**.

Constraints

1. Only buckets whose version is 3.0 or later support the configuration of user-defined domain names. The version number of a bucket is displayed in the **Basic Information** area.
2. User-defined domain names currently allow requests over HTTP, instead of HTTPS.
3. A user-defined domain name can be bound to only one bucket.
4. The suffix of a user-defined domain name can contain 2 to 6 uppercase or lowercase letters.

2.17.2 Configuring a User-Defined Domain Name

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the navigation pane, choose **Domain Name Mgmt**.

Step 3 Click **Bind User Domain Name** and enter the domain name to be bound.

The suffix of a user-defined domain name can contain 2 to 6 uppercase or lowercase letters.

Step 4 Click **OK**.

Step 5 Configure a CNAME record on the DNS, and map the user-defined domain name (for example, **example.com**) to the domain name of the bucket.

The CNAME configuration varies depending on DNS providers. For details, contact your DNS provider.

----End

2.18 Static Website Hosting

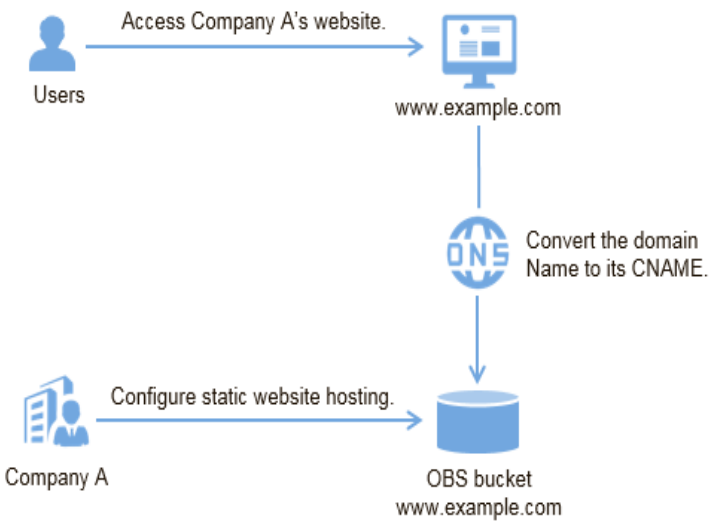
2.18.1 Static Website Hosting Overview

You can upload the content files of static websites to your bucket on OBS, authorize anonymous users the permission to read these files, and configure static website hosting for the bucket to host these files.

Static websites contain static web pages and some scripts that can run on clients, such as JavaScript and Flash. Different from static websites, dynamic websites rely on servers to process scripts, including PHP, JSP, and ASP.NET. OBS does not support scripts running on servers.

It can take up to two minutes for the configuration of static website hosting to take effect. After the static website hosting is effective in OBS, you can access the static website by using the URL provided by OBS.

Figure 2-17 Static website hosting



2.18.2 Redirection Overview

When using static website hosting, you can also configure redirection to redirect specific or all requests.

If the structure, address, or file name extension of a website is changed, users will fail to access the website using the old address (such as the address saved in the folder of favorites), and the 404 error message is returned. In this case, you can configure redirection for the website to redirect user access requests to the specified page instead of returning the 404 error page.

Typical configurations include:

- Redirecting all requests to another website.
- Redirecting specific requests based on redirection rules.

2.18.3 Configuring Static Website Hosting

You can configure static website hosting for a bucket and then use the bucket's domain name to access static websites hosted in the bucket.

It can take up to two minutes for the configuration of static website hosting to take effect.

Prerequisites

Web page files required for static website hosting have been uploaded to the specified bucket.

The static website files hosted in the bucket are accessible to anonymous users.

Static web page files in the Cold storage class have been restored. For more information, see **Restoring an Object from Cold Storage**.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 (**Optional**) If the static website files in the bucket are not accessible to anonymous users, perform this step. If they are already accessible to everyone, skip this step.

Grant the read permission for static website files to anonymous users. For details, see **Granting Anonymous Users Permission to Access Objects**.

If the bucket contains only static website files, configure the **Public Read** policy for the bucket so that all files in it are publicly accessible.

1. Choose **Permissions** > **Bucket Policies**.
2. In the **Standard Bucket Policies** area, select the **Public Read** policy for the bucket.
3. Click **Public Read**. In the confirmation dialog box that is displayed, click **Yes**.

Step 3 In the **Basic Configurations** area, click **Static Website Hosting**. The **Static Website Hosting** page is displayed.

Alternatively, you can choose **Basic Configurations** > **Static Website Hosting** from the navigation pane on the left.

Step 4 Click **Configure Static Website Hosting**. The **Configure Static Website Hosting** dialog box is displayed.

Step 5 Enable **Status**.

Step 6 Set the hosting type to the current bucket.

Step 7 Configure the homepage and 404 error page.

- **Home Page**: specifies the default homepage of the static website. When OBS Console is used to configure static website hosting, only HTML web pages are supported. When APIs are used to configure static website hosting, OBS does not have any restriction but the **Content-Type** of objects must be specified.

  OBS only allows files such as **index.html** in the root directory of a bucket to function as the default homepage. Do not set the default homepage with a multi-level directory structure (for example, **/page/index.html**).
- **404 Error Page**: specifies the error page returned when an error occurs during static website access. When OBS Console is used to configure static website hosting, only HTML, JPG, PNG, BMP, and WebP files under the root directory are supported. When APIs are used to configure static website hosting, OBS does not have any restriction but the **Content-Type** of objects must be specified.

Step 8 **Optional**: In **Redirection Rules**, configure redirection rules. Requests that comply with the redirection rules are redirected to the specific host or page.

A redirection rule is compiled in the JSON or XML format. Each rule contains a **Condition** and a **Redirect**. The parameters are described in **Table 2-44**.

Table 2-44 Parameter description

Container	Key	Description
Condition	KeyPrefixEquals	Object name prefix on which the redirection rule takes effect. When a request is sent for accessing an object, the redirection rule takes effect if the object name prefix matches the value specified for this parameter. For example, to redirect the request for object **ExamplePage.html**, set the **KeyPrefixEquals** to **ExamplePage.html**.
	HttpErrorCodeReturnedEquals	HTTP error codes upon which the redirection rule takes effect. The specified redirection is applied only when the error code returned equals the value specified for this parameter. For example, if you want to redirect requests to **NotFound.html** when HTTP error code 404 is returned, set **HttpErrorCodeReturnedEquals** to **404** in **Condition**, and set **ReplaceKeyWith** to **NotFound.html** in **Redirect**.
Redirect	Protocol	Protocol used for redirecting requests. The value can be **http** or **https**. If this parameter is not specified, the default value **http** is used.
	HostName	Host name to which the redirection is pointed. If this parameter is not specified, the request is redirected to the host from which the original request is initiated.
	ReplaceKeyPrefixWith	The object name prefix used in the redirection request. OBS replaces the value of **KeyPrefixEquals** with the value you specified here for **ReplaceKeyPrefixWith**. For example, to redirect requests for **docs** (objects in the **docs** directory) to **documents** (objects in the **documents** directory), set **KeyPrefixEquals** to **docs** under **Condition** and **ReplaceKeyPrefixWith** to **documents** under **Redirect**. This way, requests for object **docs/a.html** will be redirected to **documents/a.html**.

Container	Key	Description
	ReplaceKeyWith	The object name used in the redirection request. OBS replaces the entire object name in the request with the value you specified here for **ReplaceKeyWith**. For example, to redirect requests for all objects in the **docs** directory to **documents/error.html**, set **KeyPrefixEquals** to **docs** under **Condition** and **ReplaceKeyWith** to **documents/error.html** under **Redirect**. This way, requests for both objects **docs/a.html** and **docs/b.html** will be redirected to **documents/error.html**.
	HttpRedirectCode	HTTP status code returned to the redirection request. The default value is **301**, indicating that requests are permanently redirected to the location specified by **Redirect**. You can also set this parameter based on your service needs.

Example of setting a redirection rule

- Example 1: All requests for objects prefixed with **folder1/** are automatically redirected to pages prefixed with **target.html** on host **www.example.com** using HTTPS.

```
[
   {
   "Condition": {
      "KeyPrefixEquals": "folder1/"
      },
   "Redirect":{
      "Protocol": "https",
      "HostName": "www.example.com",
      "ReplaceKeyPrefixWith": "target.html"
      }
   }
]
```

- Example 2: All requests for objects prefixed with **folder2/** are automatically redirected to objects prefixed with **folder/** in the same bucket.

```
[
   {
   "Condition": {
      "KeyPrefixEquals": "folder2/"
      },
   "Redirect":{
      "ReplaceKeyPrefixWith": "folder/"
      }
   }
]
```

- Example 3: All requests for objects prefixed with **folder.html** are automatically redirected to the **folderdeleted.html** object in the same bucket.

```
[
   {
   "Condition": {
```



```
      "KeyPrefixEquals": "folder.html"
      },
   "Redirect":{
      "ReplaceKeyWith": "folderdeleted.html"
      }
   }
]
```

- Example 4: If the HTTP status code 404 is returned, the request is automatically redirected to the page prefixed with **report-404/** on host **www.example.com**.

  For example, if you request the page **ExamplePage.html** but the HTTP 404 error is returned, the request will be redirected to the **report-404/ExamplePage.html** page on the **www.example.com**. If the 404 redirection rule is not specified, the default 404 error page configured in the previous step is returned when the HTTP 404 error occurs.

```
[
   {
   "Condition": {
      "HttpErrorCodeReturnedEquals": "404"
      },
   "Redirect":{
      "HostName": "www.example.com",
      "ReplaceKeyPrefixWith": "report-404/"
      }
   }
]
```

Step 9 Click **OK**.

After the static website hosting is effective in OBS, you can access the static website by using the URL provided by OBS.

NOTE

In some conditions, you may need to clear the browser cache before the expected results are displayed.

----End

2.18.4 Configuring Redirection

You can redirect all requests for a bucket to another bucket or URL by configuring redirection rules.

Prerequisites

Web page files required for static website hosting have been uploaded to the specified bucket.

The static website files hosted in the bucket are accessible to anonymous users.

Static web page files in the Cold storage class have been restored. For more information, see **Restoring an Object from Cold Storage**.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the **Basic Configurations** area, click **Static Website Hosting**. The **Static Website Hosting** page is displayed.

Alternatively, you can choose **Basic Configurations** > **Static Website Hosting** from the navigation pane on the left.

Step 3 Click **Configure Static Website Hosting**. The **Configure Static Website Hosting** dialog box is displayed.

Step 4 Enable **Status**.

Step 5 Set **Hosting By** to **Redirection**, and enter the access domain name or URL of the bucket to which requests are redirected.

Step 6 Click **OK**.

Step 7 In the bucket list, click the bucket to which requests for the static website are redirected.

Step 8 (**Optional**) If the static website files in the bucket are not accessible to anonymous users, perform this step. If they are already accessible to everyone, skip this step.

Grant the read permission for static website files to anonymous users. For details, see **Granting Anonymous Users Permission to Access Objects**.

If the bucket contains only static website files, configure the **Public Read** policy for the bucket so that all files in it are publicly accessible.

1. Choose **Permissions** > **Bucket Policies**.
2. In the **Standard Bucket Policies** area, select the **Public Read** policy for the bucket.
3. Click **Public Read**. In the confirmation dialog box that is displayed, click **Yes**.

Step 9 **Verification**: Input the access domain name of the bucket in the web browser and press **Enter**. The bucket or URL to which requests are redirected will be displayed.

NOTE

In some conditions, you may need to clear the browser cache before the expected results are displayed.

----End

2.18.5 Using a User-Defined Domain Name to Configure Static Website Hosting

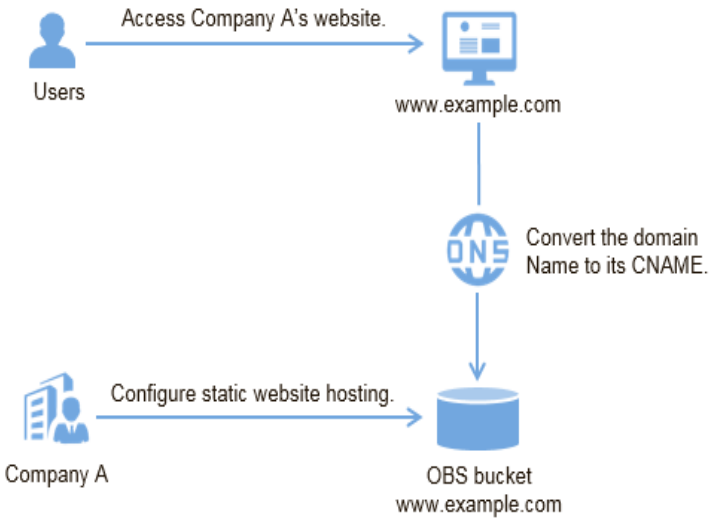
OBS allows you to access static websites hosted by OBS using user-defined domain names. This section uses a specific scenario as an example to describe how to use a user-defined domain name to configure static website hosting. For a basic understanding of the concepts and operations about the static website hosting on OBS, see **Configuring Static Website Hosting**.

Scenario

Company **A** has a large number of files to archive but it does not want to put the time and effort into its storage resources. Therefore, the company subscribes to

OBS for hosting static websites and expects that the usernames under the company account can access the static resources through a user-defined domain name. See **Figure 2-18**.

Figure 2-18 Using a user-defined domain name to access hosted static website



Operation Process

Create a bucket on OBS Console first, for storing static website resources, and enable static website hosting for this bucket. Then use DNS to create and configure domain name hosting. The procedure is as follows:

1. **Register a domain name.**
2. **Create a bucket.**
3. **Upload static website files.**
4. **Configure static website hosting on OBS.**
5. **Bind a user-defined domain name.**
6. **Create and configure domain name hosting.**
7. **Verify the configuration.**

Data Planning

Table 2-45 describes the data to be planned before this configuration.

Table 2-45 Data planning

Item	Description	Example
User-defined domain name	Indicates user's own domain name.	www.example.com

Item	Description	Example
Static website homepage	Indicates the index page that is returned when you access a static website, that is, the homepage.	index.html
404 error page	When an incorrect static website path is accessed, the 404 error page is returned.	error.html

- For example, the content of the **index.html** file is as follows:

```
<html>
 <head>
    <title>Hello OBS!</title>
    <meta charset="utf-8">
 </head>
 <body>
    <p>Welcome to use OBS static website hosting.</p>
    <p>This is the homepage.</p>
 </body>
</html>
```

- For example, the content of the **error.html** file is as follows:

```
<html>
 <head>
    <title>Hello OBS!</title>
    <meta charset="utf-8">
 </head>
 <body>
    <p>Welcome to use OBS static website hosting.</p>
    <p>This is the 404 error page.</p>
 </body>
</html>
```

Procedure

Step 1 Register a domain name.

If you have a registered domain name, skip this step.

If you do not have a registered domain name, register one with a registrar of your choice. In this scenario, the example domain name **www.example.com** is used. In practice, you need to replace the domain name with the one you actually planned.

Step 2 Create a bucket.

There are no special requirements on bucket names. Create a bucket for storing static website files as prompted. The following example describes how to create a bucket named **example**:

1. Log in to OBS Console.
2. Click **Create Bucket** in the upper right corner of the page.
3. Configure the following parameters in the dialog box that is displayed:
   - **Region**: Select a region closest to you.
   - **Bucket Name**: Enter **example**.

- **Storage Class**: It is recommended that you select **Standard**.

  NOTE

  You can also select the Warm, or Cold storage class based on the website requirements for access frequency and speed. For details about storage classes, see **Storage Classes**.

- **Bucket Policy**: Select **Public Read** to allow any user to access objects in the bucket.
- **Default Encryption**: Choose **Disable**.

4. Click **Create Now** to complete the creation.

Step 3 Upload static website files to the bucket.

Prepare the static website files to be uploaded and perform the following steps to upload all static website files to bucket **example**.

1. Click the bucket name **example** to go to the **Objects** page.
2. Click **Upload Object**.
3. Drag the prepared static website files to the **Upload Object** area.

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

   NOTE

   - The static website files cannot be encrypted for upload.
   - The website home page file (**index.html**) and 404 error page (**error.html**) must be stored in the root directory of the bucket.
   - It is recommended that you select **Standard** for the storage class. If the storage class of a static website file is Cold, you need to restore the static website file before you can access it. For details, see **Restoring an Object from Cold Storage**.

4. Click **Upload** to complete the upload.

Step 4 Configure static website hosting.

After uploading the static website files, you need to configure the static website hosting function for the bucket.

NOTE

You can also redirect the entire static website to another bucket or domain name. For details, see **Configuring Redirection**.

1. Click the bucket name **example** to go to the **Objects** page.
2. In the navigation pane, choose **Basic Configurations** > **Static Website Hosting**. The **Static Website Hosting** page is displayed.
3. Click **Configure Static Website Hosting**. The **Configure Static Website Hosting** dialog box is displayed.
4. Enable **Status**.
5. Set **Hosting Type** to **Host a static website**.

   NOTE

   You can also configure redirection rules based on service requirements to implement website content redirection. For details, see **Configuring Static Website Hosting**.

6. Set the **Home Page** to **index.html** as planned, and the **404 Error Page** to **error.html**.

7. Click **OK**.

Step 5 Bind a user-defined domain name.

To bind a user-defined domain name to a bucket, perform the following steps:

1. Click the bucket name **example** to go to the **Objects** page. In the navigation pane, choose **Domain Name Mgmt**.
2. Click **Bind User Domain Name** and set **User Domain Name** to **www.example.com**.
3. Click **OK**. The user-defined domain name is bound to the bucket.

Step 6 Create and configure domain name hosting.

To facilitate unified management of your user-defined domain names and static websites and implement cloud-based services, directly manage your user-defined domain names on DNS. After the hosting is configured, you can perform subsequent management of the domain name on DNS, including managing record sets and PTR records, as well as creating wildcard DNS records.

Alternatively, you can add a CNAME record to the DNS at the DNS registrar, mapping to the static website domain name hosted by the bucket.

To create and configure domain name hosting on DNS, perform the following steps:

1. Add a public zone.

   Use the root domain name **example.com** created in **Step 1** as the name of the public zone to be created. For details about how to create a public zone, see "Step 1. Create a Public Zone" in section "Routing Internet Traffic to a Website" of the *Domain Name Service User Guide*.

2. Add a CNAME record.

   In DNS, add a record set for the sub-domain name **www.example.com** of the hosted domain name, to map the CNAME of the sub-domain name to the static website domain name hosted by OBS. Configure the parameters as follows:

   - **Name**: Enter **www**.
   - **Type**: Select **CNAME-Canonical name**.
   - **Line**: Select **Default**.
   - **TTL (s)**: Retain the default value.
   - **Value**: Domain name to map, that is, the static website domain name hosted by bucket **example**.

   For details, see section "Adding a CNAME Record Set" in the *Domain Name Service User Guide*.

3. Change the DNS server address at your domain name registrar.

   At your domain name registrar, change the DNS server address in the NS record of the root domain name to the cloud DNS server address. The specific address is the NS value of the public zone in DNS.

   For details about how to change the addresses of the DNS servers, see "Step 4. Change DNS Servers of the Domain Name" in section "Routing Internet Traffic to a Website" of the *Domain Name Service User Guide*.

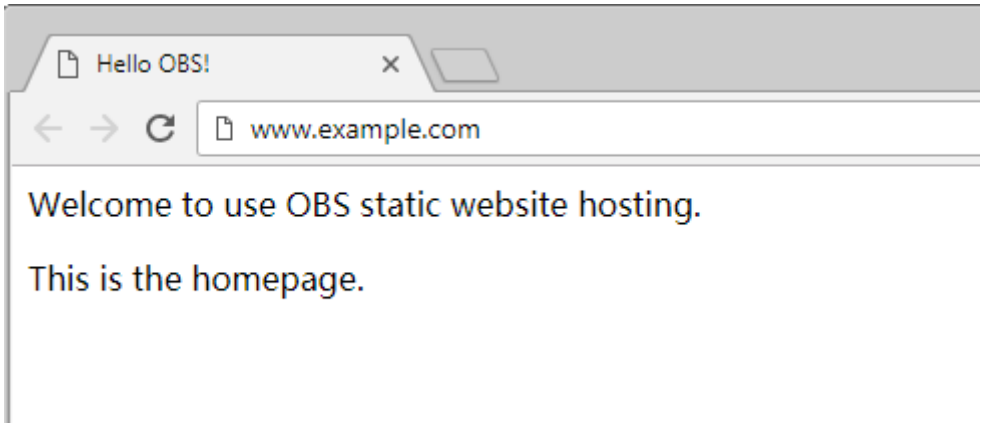
NOTE

The address change will be effective within 48 hours. The actual time taken varies depending on the domain name registrar.

Step 7 Verify that the configuration is successful.

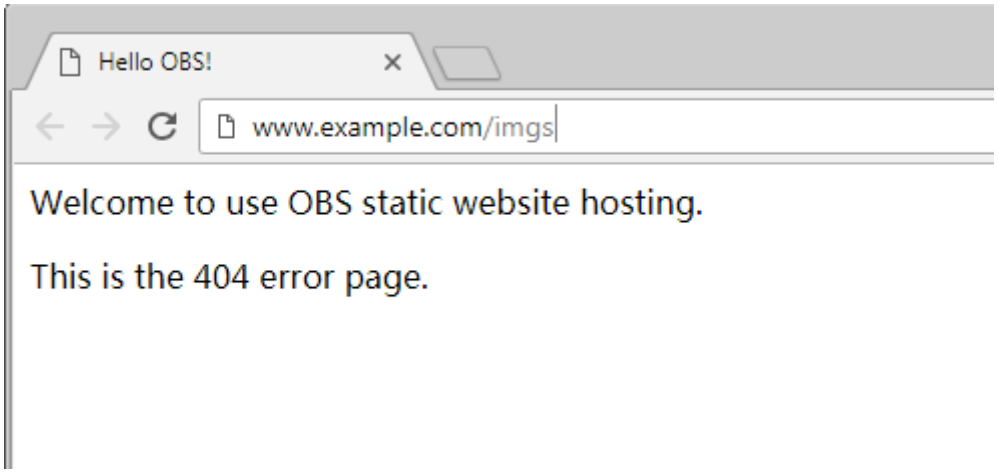
- Open **www.example.com** using a browser to verify that the default homepage can be accessed. See **Figure 2-19**.

  Figure 2-19 Default homepage

- Visit a static file that does not exist in the bucket, for example, opening **www.example.com/imgs** in a browser, to verify that the 404 error page can be returned. See **Figure 2-20**.

  Figure 2-20 404 error page

NOTE

In some conditions, you may need to clear the browser cache before the expected results are displayed.

----End

Website Update

If you need to update a static file, such as a picture, a piece of music, an HTML file, or a CSS file, you can re-upload the static file.

By default, if two files in a path share one name, the newly uploaded file overwrites the original one. To prevent files from being overwritten, you can enable the versioning function. Versioning allows you to keep multiple versions of a static file, so that you can retrieve and restore history versions conveniently. With versioning enabled, data can be restored rapidly when accidental operations or application faults occur. For detailed information about versioning, see chapter **Versioning Overview**.

2.19 Cross-Origin Resource Sharing

2.19.1 CORS Overview

CORS is a browser-standard mechanism provided by the World Wide Web Consortium (W3C). It defines the interaction methods between client-side web applications in one origin and resources in another origin. For general web page requests, website scripts and contents in one origin cannot interact with those in another origin because of Same Origin Policies (SOPs).

The CORS specification is supported to allow cross-origin requests to access OBS resources.

OBS supports static website hosting. Static websites stored in OBS can respond to website requests from another origin only when CORS is configured for the bucket.

Typical application scenarios of CORS are as follows:

- Enables JavaScript and HTML5 to be used for establishing web applications that can directly access resources in OBS. No proxy servers are required for transfer.
- Enables the dragging function of HTML5 to be used to upload files to OBS (with the upload progress displayed) or update OBS contents using web applications.
- Hosts external web pages, style sheets, and HTML5 applications in different origins. Web fonts or pictures in OBS can be shared by multiple websites.

The configuration of CORS takes effect within two minutes.

2.19.2 Configuring CORS

This section describes how to use CORS in HTML5 to implement cross-origin access.

Prerequisites

Static website hosting has been configured. For details, see **Configuring Static Website Hosting**.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the **Basic Configurations** area, click **CORS Rules**. The **CORS Rules** page is displayed.

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

Step 3 Click **Create**. The **Create CORS Rule** dialog box is displayed.

NOTE

A bucket can have a maximum of 100 CORS rules configured.

Step 4 In the **CORS Rule** dialog box, configure **Allowed Origin**, **Allowed Method**, **Allowed Header**, **Exposed Header**, and **Cache Duration (s)**.

Table 2-46 Parameters in CORS rules

Parameter	Description
Allowed Origin	Mandatory Specifies the origins from which requests can access the bucket. Multiple matching rules are allowed. One rule occupies one line, and allows one wildcard character (*) at most. An example is given as follows: http://rds.example.com https://*.vbs.example.com
Allowed Method	Mandatory Specifies the allowed request methods for buckets and objects. The methods include Get, Post, Put, Delete, and Head.
Allowed Header	Optional Specifies the allowed headers in cross-origin requests. Only CORS requests matching the allowed headers are valid. You can enter multiple allowed headers (one per line) and each line can contain one wildcard character (*) at most. Spaces and special characters including **&:<** are not allowed.
Exposed Header	Optional Specifies the exposed headers in CORS responses, providing additional information for clients. By default, a browser can access only headers **Content-Length** and **Content-Type**. If the browser wants to access other headers, you need to configure them in this parameter. You can enter multiple exposed headers (one per line). Spaces and special characters including ***&:<** are not allowed.

Parameter	Description
Cache Duration (s)	Mandatory Specifies the duration that your browser can cache CORS responses, expressed in seconds. The default value is **100**.

Step 5 Click **OK**.

Message "The CORS rule created successfully." is displayed. The CORS configuration will take effect within two minutes.

Then, only the addresses specified in **Allowed Origin** can access the OBS bucket over the methods specified in **Allowed Method**. Suppose you are configuring a CORS rule for bucket **testbucket** by setting **Allowed Origin** to **https://www.example.com**, **Allowed Method** to **GET**, **Allowed Header** to *, **Exposed Header** to **ETag**, and **Cache Duration (s)** to **100**. Then, only GET requests from **https://www.example.com** are allowed to access bucket **testbucket**. In addition, there are no limits on headers in a request, the ETag value can be returned in the response, and the client which the requests are from can cache the CORS response for 100 seconds.

----End

2.20 URL Validation

2.20.1 URL Validation Overview

Some rogue websites may steal links from other websites to enrich their content without any costs. Link stealing hurts the interests of the original websites and it is also a strain on their servers. URL validation is designed to address this issue.

In HTTP, the **Referer** field allows websites and web servers to identify where people are visiting them from. URL validation of OBS utilizes this **Referer** field. The idea is that once you find that a request to your resource is not originated from an authorized source, you can have the request blocked or redirected to a specific web page. This way, OBS prevents unauthorized access to data stored in buckets.

Such authorization is controlled using both whitelists and blacklists.

2.20.2 Configuring URL Validation

OBS blocks access requests from blacklisted URLs and allows those from whitelisted URLs.

Prerequisites

Static website hosting has been enabled.

Procedure

Step 1 In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.

Step 2 In the **Basic Configurations** area, click **URL Validation**. The **URL Validation** page is displayed.

Step 3 Click  next to the text box of **Whitelisted Referers** or **Blacklisted Referers**, and enter the referers.

Principles for setting **Referers**:

- The length of a whitelist or blacklist cannot exceed 1024 characters.
- Referer format:
  - You can enter multiple referers, each in a line.
  - The referer parameter supports asterisks (*) and question marks (?). An asterisk works as a wildcard that can replace zero or multiple characters, and a question mark (?) can replace a single character.
  - If the referer header field contains **http** or **https** during download, the referer must contain **http** or **https**.
- If **Whitelisted Referers** is left blank but **Blacklisted Referers** is not, all websites except those specified in the blacklist are allowed to access data in the target bucket.
- If **Whitelisted Referers** is not left blank, only the websites specified in the whitelist are allowed to access the target bucket no matter whether **Blacklisted Referers** is left blank or not.

NOTE

If **Whitelisted Referers** is configured the same as **Blacklisted Referers**, the blacklist takes effect. For example, if both **Whitelisted Referers** and **Blacklisted Referers** are set to **https://www.example.com**, access requests from this address will be blocked.

- If **Whitelisted Referers** and **Blacklisted Referers** are both left blank, all websites are allowed to access data in the target bucket by default.
- Before determining whether a user has the four types of permissions (read, write, ACL read, and ACL write) for a bucket or objects in the bucket, check whether this user complies with the URL validation principles of the **Referer** field.

Step 4 Click  to save the settings.

----End

2.21 Monitoring

2.21.1 Monitoring OBS

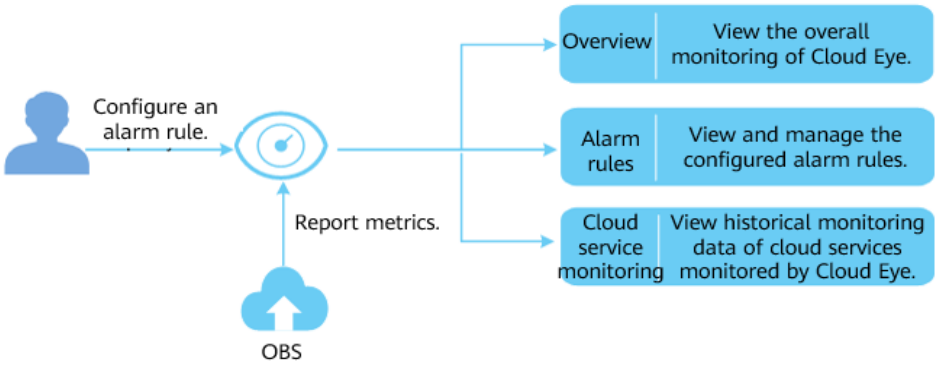
Scenarios

In the use of OBS, you may send PUT and GET requests that generate upload and download traffic, or receive error responses from the server. To learn the requests,

traffic, and error responses in a timely manner, you can use Cloud Eye to perform automatic and real-time monitoring over your buckets.

You do not need to separately subscribe to Cloud Eye. It starts automatically once you create a resource (a bucket, for example) in OBS. For more information about Cloud Eye, see *Cloud Eye User Guide*.

Figure 2-21 Cloud Eye monitoring



Setting Alarm Rules

In addition to automatic and real-time monitoring, you can configure alarm rules in Cloud Eye to receive alarm notifications when there are exceptions.

For details, see section "Creating an Alarm Rule" in *Cloud Eye User Guide*.

Viewing OBS Monitoring Metrics

Cloud Eye monitors **OBS monitoring metrics** in real time. You can view detailed monitoring statistics of each metric on the console of Cloud Eye.

For details, see section "Querying Cloud Service Monitoring Metrics" in *Cloud Eye User Guide*.

2.21.2 OBS Monitoring Metrics

Functions

This section defines the namespace, list, and dimensions of monitoring metrics reported by OBS to Cloud Eye. You can use the management console or APIs provided by Cloud Eye to search for monitoring metrics and alarms generated by OBS.

Namespace

SYS.OBS

Monitoring Metrics

Table 2-47 OBS metrics (for requests)

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
get_request_count	GET Requests	Number of GET requests made to all buckets and objects in the buckets of a region. Unit: count	≥ 0 counts	Bucket	1 minute
put_request_count	PUT Requests	Number of PUT requests made to all buckets and objects in the buckets of a region. Unit: count	≥ 0 counts	Bucket	1 minute
first_byte_latency	First Byte Download Delay	Average time from receiving a GET request to the time that the system starts to respond in a measurement period. Unit: ms	≥ 0 ms	Bucket	1 minute
request_count_4xx	4XX Status Codes	Number of requests whose status code returned by the server is 4*xx*. Unit: count	≥ 0 counts	User Bucket API	1 minute
request_count_5xx	5XX Status Codes	Number of requests whose status code returned by the server is 5*xx*. Unit: count	≥ 0 counts	User Bucket API	1 minute
total_request_latency	Average Request Latency	Average time from receiving a request to the time that the system response ends in a measurement period. Unit: ms	≥ 0 ms	User Bucket API	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
request_count_per_second	Total TPS	Average number of requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_get_per_second	GET Request TPS	Average number of GET requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_put_per_second	PUT Request TPS	Average number of PUT requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_delete_per_second	DELETE Request TPS	Average number of DELETE requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_success_rate	Request Success Rate	Used to measure the availability of the storage service system. It refers to the percentage of non-server error requests (with status code 5*xx* returned) in the total request count. It is calculated as follows: (1 – 5*xx* requests/Total requests) x 100% Unit: %	≥ 0, ≤ 100	User Bucket API Domain name	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
effective_request_rate	Valid request rate	Validity of client requests. Percentage of the valid requests in the total requests. It is calculated as follows: (Number of client requests whose returned status code is 2*xx* or 3*xx*/Total number of requests) x 100% Unit: %	≥ 0, ≤ 100	User Bucket API	1 minute
request_break_rate	Request interruption rate	Percentage of the number of requests interrupted by a client to the total number of requests. It is calculated as follows: (Number of requests interrupted by a client/ Total number of requests) x 100% Unit: %	≥ 0, ≤ 100	User Bucket API	1 minute
request_code_count	HTTP status code count	Measures the number of requests with status codes returned by the server. For details about the response status codes, see **Table 2-51**. Unit: count	≥ 0 counts	Bucket API HTTP status code	1 minute
api_request_count_per_second	API request TPS	Average number of specific API requests sent to all buckets and objects of a tenant per second within a statistical period. For details about the supported APIs, see **Table 2-50**.	≥ 0 counts	Bucket API	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
request_count_monitor_2XX	2xx Status Codes	Count of server responses to requests whose status codes are 2*xx*. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_monitor_3XX	3xx Status Codes	Count of server responses to requests whose status codes are 3*xx*. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
download_bytes	Total Download Bandwidth	Total size of objects downloaded per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
download_bytes_extranet	Download Bandwidth (Internet)	Total size of objects downloaded over the Internet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
download_bytes_intranet	Download Bandwidth (Intranet)	Total size of objects downloaded over the Intranet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
upload_bytes	Total Upload Bandwidth	Total size of objects uploaded per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
upload_bytes_extranet	Upload Bandwidth (Internet)	Total size of objects uploaded over the Internet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
upload_bytes_intranet	Upload Bandwidth (Intranet)	Total size of objects uploaded over the Intranet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
download_traffic	Total Download Traffic	Total size of objects downloaded in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
download_traffic_extranet	Download Traffic (Internet)	Total size of objects downloaded over the Internet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
download_traffic_intranet	Download Traffic (Intranet)	Total size of objects downloaded over the Intranet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
upload_traffic	Total Upload Traffic	Total size of objects uploaded in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
upload_traffic_extranet	Upload Traffic (Internet)	Total size of objects uploaded over the Internet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
upload_traffic_intranet	Upload Traffic (Intranet)	Total size of objects uploaded over the Intranet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute

Table 2-48 OBS metrics (for storage)

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
capacity_total	Total Used Storage Space	Measures the storage space occupied by all data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_standard	Used Space - Standard Storage	Measures the storage space occupied by Standard data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_infrequent_access	Used Space - Warm Storage	Measures the storage space occupied by Warm data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_archive	Used Space - Cold Storage	Measures the storage space occupied by Cold data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
object_num_all	Total Number of Objects	Measures the total number of objects (including folders and all file versions) stored in all storage classes. Unit: count	≥ 0	User Bucket	30 minutes
object_num_standard_total	Number of Objects - Standard Storage	Measures the total number of objects (including folders and all file versions) stored in the Standard storage class. Unit: count	≥ 0	User Bucket	30 minutes
object_num_infrequent_access_total	Number of Objects - Warm Storage	Measures the total number of objects (including folders and all file versions) stored in the Warm storage class. Unit: count	≥ 0	User Bucket	30 minutes

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
object_num_archive_total	Number of Objects - Cold Storage	Measures the total number of objects (including folders and all file versions) stored in the Cold storage class. Unit: count	≥ 0	User Bucket	30 minutes

Dimensions

Table 2-49 Dimensions

Key	Value
User	User dimension. The value is a domain ID.
Bucket_Name	Bucket dimension. The value is the bucket name.
Bucket_Name - API Name	API dimension. For details about the value, see **Table 2-50**.
Bucket_Name - Domain	Domain name dimension. The value is the domain name of the bucket to be accessed.
• User - HTTP Status Code • Bucket_Name - HTTP Status Code • Bucket_Name - API Name - HTTP Status Code	HTTP return code dimension. For details about the values, see **Table 2-51**.

Request APIs

Table 2-49 lists the APIs supported by the **Bucket_Name - API Name** dimension:

Table 2-50 Request APIs

ID	Name
LIST.BUCKETS	Listing Buckets

ID	Name
PUT.BUCKET	Creating a Bucket
LIST.BUCKET.OBJECTS	Listing Objects in a Bucket
LIST.BUCKET.OBJECTVERSIONS	Listing Objects in a Bucket (Versioning)
HEAD.BUCKET	Obtaining Bucket Metadata
GET.BUCKET.LOCATION	Obtaining Bucket Location
LIST.BUCKET.UPLOADS	Listing Multipart Uploads
POST.OBJECT.MULTIDELETE	Batch Deleting Objects
LIST.BUCKET.OBJECTS	Listing Objects
POST.OBJECT	Uploading Objects - POST
PUT.PART	Uploading Parts
PUT.PART.COPY	Copying Parts
DELETE.UPLOAD	Canceling a Part
LIST.OBJECT.UPLOAD	Listing Uploaded Parts of an Object
POST.UPLOAD.COMPLETE	Completing a Multipart Upload
POST.UPLOAD.INIT	Initiating a Multipart Upload
PUT.OBJECT	Uploading Objects
APPEND.OBJECT	Appending an Object
PUT.OBJECT.COPY	Copying an Object
DELETE.OBJECT	Deleting an Object
GET.OBJECT	Downloading an Object
HEAD.OBJECT	Heading objects
LIST.BUCKET.OBJECTVERSIONS	Listing Object Versions
POST.OBJECT.RESTORE	Restoring an Object
PUT.OBJECT.METADATA	Modifying Object Metadata

HTTP Status Codes

Table 2-49 lists the HTTP status codes supported by the **User - HTTP Status Code**, **Bucket_Name - HTTP Status Code**, **Bucket_Name - API Name - HTTP Status Code** dimensions:

Table 2-51 HTTP status codes

HTTP Status Code	Description
400	Incorrect request packet format.
401	Failed to authenticate and authorize.
403	Insufficient permission, access denied, limited MimeType, file type not allowed, or others
404	The requested resource does not exist.
405	The specified method is not allowed against the requested resource.
406	CRC32 check failed for the uploaded data.
413	Incorrect size of the uploaded object.
579	The object is successfully uploaded, but the callback fails.
599	The server fails to operate.
612	The specified resource does not exist or has been deleted.
614	The target resources already exist.
701	The block expires, the segments are discontinuous, the total block size does not match the object size, or others.

2.22 Related Operations

2.22.1 Creating an Agency

To use some OBS features, you need to use IAM agencies to grant required permissions to OBS for processing your data.

Creating an Agency for Uploading Logs

Step 1 In the **Logging** dialog box, click **Create Agency** to jump to the **Agencies** page on the **Identity and Access Management** console.

Step 2 In the navigation pane, choose **Policy Management** > **Agency Policies**.

Step 3 Click **Create Agency**.

Step 4 Enter an agency name.

Step 5 Select **Cloud service** for the **Agency Type**.

Step 6 Select **Object Storage Service (OBS)** as the cloud service.

Step 7 Set a validity period.

Step 8 In the **Permissions** area, find **Global service** > **OBS** and click **Attach Policy** on the right.

1. Search for and select the custom policy that has the permission to upload logs to the bucket, and click **OK**.

   If you have not created any custom policy, click **Policies** in the navigation pane on the left to create one.

   When creating a custom policy, select **Global services** for **Scope** and select **JSON** for **Policy View**. The policy content is as follows:

   NOTE

   When coding the policy content in an actual scenario, replace **mybucketlogs** with the actual bucket name:

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

2. (Optional) If the log storage bucket has default encryption enabled, the agency also requires the **KMS Administrator** permission for the region where the bucket is located.

   In the region project of the log storing bucket, click **Attach Policy**. In the displayed dialog box, search for the **KMS Administrator** policy, select the policy, and then click **OK**.

Step 9 Click **OK** to complete the agency creation.

----End

2.23 Troubleshooting

2.23.1 An Object Fails to Be Downloaded Using Internet Explorer 11

Symptom

A user logs in to OBS Console using Internet Explorer 11 and uploads an object. When the user attempts to download the object to the original path to replace the original object without closing the browser, a message is displayed indicating a download failure. Why does this happen?

For example, a user uploads object **abc** from the root directory of local drive C to a bucket in OBS Console. When the user attempts to download the object to the

root directory of local drive C to replace the original object without closing the browser, a message is displayed indicating a download failure.

Answer

This problem is caused by browser incompatibility. It can be solved by using a different web browser.

If this problem occurs, close the browser and try again.

2.23.2 OBS Console Couldn't Be Opened in Internet Explorer 9

Question

Why OBS Console cannot be opened in Internet Explorer 9, even if the address of OBS Console can be pinged?

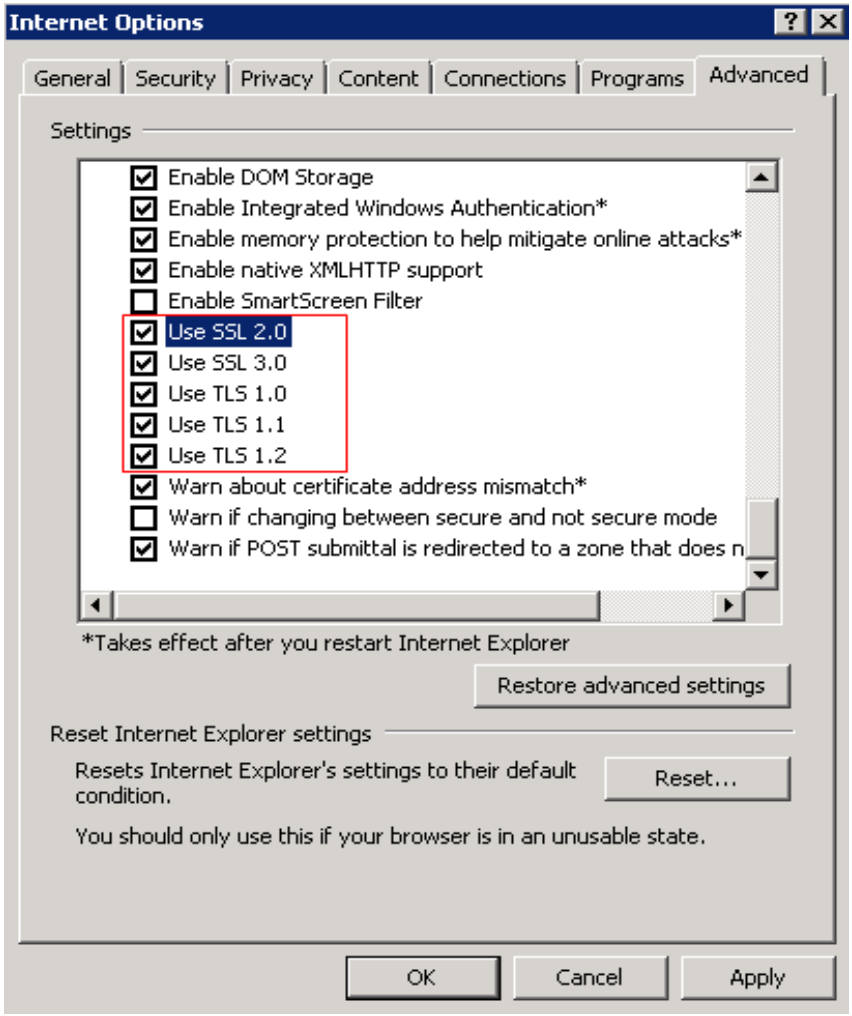
Answer

Confirm whether **Use SSL** and **Use TLS** are selected in **Internet Options**. If not, do as follows and try again:

Step 1 Open Internet Explorer 9.

Step 2 Click **Tools** in the upper right corner and choose **Internet Options** > **Advanced**. Then select **Use SSL 2.0**, **Use SSL 3.0**, **Use TLS 1.0**, **Use TLS 1.1**, and **Use TLS 1.2**, as shown in **Figure 2-22**.

Figure 2-22 Internet Options



Step 3 Click **OK**.

----End

2.23.3 The Object Name Changes After an Object with a Long Name Is Downloaded to a Local Computer

Symptom

After an object with a relatively long name is downloaded to a local path, the object name changes.

Cause

For Windows, a file name, including the file name extension, can contain a maximum of 255 characters.

When an object with a name containing more than 255 characters is downloaded to a local computer, the system keeps only the first 255 characters automatically.

Solution

Change the object name to a string of no more than 255 characters.

2.23.4 Failed to Configure Event Notifications

Symptom

During the configuration of event notifications on OBS, message "OBS is not authorized to use this topic. Go to SMN to authorize OBS to use this topic." is displayed.

Solution

Go to the SMN console. On the **Configure Topic Policy** page, select **OBS** under **Services that can publish messages to this topic**.

For details about how to use the SMN service, see "Topic Policy" in the *SMN User Guide*.

2.23.5 Time Difference Is Longer Than 15 Minutes Between the Client and Server

Symptom

Error message "Time difference is longer than 15 minutes between the client and server" or "The difference between the request time and the current time is too large" is displayed during the use of OBS.

Cause

For security purposes, OBS verifies the time offset between the client and server. If the offset is longer than 15 minutes, the OBS server will reject your requests and this error message is reported.

Solution

To resolve this problem, adjust your local time (UTC) and try again.

2.24 Error Code List

If a request fails to be processed due to errors, an error response is returned. An error response contains an error code and error details. **Table 2-52** lists some common error codes in OBS error responses.

Table 2-52 OBS error codes

Error Code	Description
Obs.0000	Invalid parameter.
Obs.0001	All access requests to this object are invalid.
Obs.0002	The absolute path of a file cannot exceed 1023 characters. Please retry.
Obs.0003	The connection timed out.

Error Code	Description
Obs.0004	Time difference is longer than 15 minutes between the client and server. Correctly set the local time. For security purposes, OBS verifies the time offset between the client and server. If the offset is longer than 15 minutes, the OBS server will reject your requests and this error message is reported. To resolve this problem, adjust your local time (UTC) and try again.
Obs.0005	The server load is too heavy. Try again later.
Obs.0006	The number of buckets has reached the upper limit. An account (including all IAM users under this account) can create a maximum of 100 buckets and parallel file systems. You can use the fine-grained access control of OBS to properly plan and use buckets.
Obs.0007	The target bucket does not exist or is not in the same region with the current bucket.
Obs.0008	The account has not been registered with the system. Only a registered account can be used.
Obs.0009	A conflicting operation is being performed on this resource. Please retry. This is because that there is a bucket with the same name as the bucket you are creating in OBS and the existing bucket has been released in the recent period due to arrears. In such case, try another bucket name.
Obs.0010	Deletion failed. Check whether objects or objects of historical versions exist in the bucket.
Obs.0011	The bucket policy is invalid. Configure it again.
Obs.0012	The requested bucket name already exists. Bucket namespace is shared by all users in the system. Enter a different name and try again.
Obs.0013	The requested folder name already exists. Enter a different name and try again.
Obs.0014	The file size has exceeded 50 MB. Use OBS Browser+ to upload it.
Obs.0015	The absolute path in the search criteria cannot exceed 1023 characters. Please retry.
Obs.0016	Upload failed. Possible causes: 1. The network is abnormal. 2. You have incorrect or no permissions to write the bucket.

Error Code	Description
Obs.0017	The end time of the new validity period must be later than that of the old validity period.
Obs.0018	The validity period cannot be shorter than the remaining period.
Obs.0019	Cannot determine whether the bucket has objects or fragments. Check whether you have the read permission for this bucket.
Obs.0020	TMS system error. Try again later.
Obs.0021	You do not have permissions to access TMS. Configure the required permissions in IAM.
Obs.0022	The TMS system is busy. Try again later.

3 FAQ

3.1 OBS Basics

3.1.1 How Can I Get Started with OBS?

Create an account, add a payment method, and you can start using OBS.

If you use an IAM user, ensure that the user has been added to a user group that has the permissions required to use OBS.

3.1.2 How Do I Obtain an OBS Endpoint?

You can access OBS through domain names. When you are using the API, third-party tools, or other methods to access OBS, you can use domain names to conveniently locate resources in OBS.

Before using OBS, ensure that the DNS server address has been correctly configured on the client.

Endpoints vary depending on services and regions. The following table lists OBS endpoints.

Table 3-1 OBS endpoints

Region Name	Region	Endpoint	Protocol
AP-Kuala Lumpur-OP6	my-kualalumpur-1	obs.my-kualalumpur-1.alphaedge.tmone.com.my	HTTPS/HTTP

3.1.3 What Are the Advantages of Object Storage over SAN and NAS Storage?

- SAN storage provides LUNs or volumes for applications. LUNs and volumes are forms of disk storage. Upper-layer applications use Fibre Channel or iSCSI protocols to access SAN storage. SAN storage focuses on disk management. For other purposes, SAN storage must rely on upper-layer applications.
- NAS storage provides file systems or folders for applications. Upper-layer applications use NFS or CIFS protocols to access NAS storage. Directory trees of file systems must be maintained.
- Object storage is suitable for web applications. A massive bucket storage space is provided based on a URL address to store a wide range of file objects. Object storage adopts a flat architecture. Users do not need to maintain complex file directories. There is no need to worry about running out of storage because the storage a bucket can provide is practically unlimited.

3.1.4 Which Types of Data Can Be Stored in OBS?

OBS can store all types of data.

3.1.5 How Much Data Can I Store in OBS?

There are no restrictions on the total capacity or number of objects or files that can be stored by the OBS system or in any single bucket. However, there are limitations on what you can upload to your bucket at a time.

- OBS Console allows you to upload files in a batch. Up to 100 files can be uploaded at a time, with the total size of no more than 5 GB. If you upload only one file in a batch upload, it cannot exceed 5 GB in size.
- If you use OBS Browser+ or an API, you can upload a single object of up to 48.8 TB.

3.1.6 Does OBS Support Traffic Monitoring?

Yes.

On Cloud Eye, you can monitor the OBS metrics described in the following table.

Table 3-2 OBS metrics (for requests)

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
get_request_count	GET Requests	Number of GET requests made to all buckets and objects in the buckets of a region. Unit: count	≥ 0 counts	Bucket	1 minute
put_request_count	PUT Requests	Number of PUT requests made to all buckets and objects in the buckets of a region. Unit: count	≥ 0 counts	Bucket	1 minute
first_byte_latency	First Byte Download Delay	Average time from receiving a GET request to the time that the system starts to respond in a measurement period. Unit: ms	≥ 0 ms	Bucket	1 minute
request_count_4xx	4XX Status Codes	Number of requests whose status code returned by the server is 4*xx*. Unit: count	≥ 0 counts	User Bucket API	1 minute
request_count_5xx	5XX Status Codes	Number of requests whose status code returned by the server is 5*xx*. Unit: count	≥ 0 counts	User Bucket API	1 minute
total_request_latency	Average Request Latency	Average time from receiving a request to the time that the system response ends in a measurement period. Unit: ms	≥ 0 ms	User Bucket API	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
request_count_per_second	Total TPS	Average number of requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_get_per_second	GET Request TPS	Average number of GET requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_put_per_second	PUT Request TPS	Average number of PUT requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_delete_per_second	DELETE Request TPS	Average number of DELETE requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_success_rate	Request Success Rate	Used to measure the availability of the storage service system. It refers to the percentage of non-server error requests (with status code 5*xx* returned) in the total request count. It is calculated as follows: (1 – 5*xx* requests/Total requests) x 100% Unit: %	≥ 0, ≤ 100	User Bucket API Domain name	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
effective_request_rate	Valid request rate	Validity of client requests. Percentage of the valid requests in the total requests. It is calculated as follows: (Number of client requests whose returned status code is 2*xx* or 3*xx*/Total number of requests) x 100% Unit: %	≥ 0, ≤ 100	User Bucket API	1 minute
request_break_rate	Request interruption rate	Percentage of the number of requests interrupted by a client to the total number of requests. It is calculated as follows: (Number of requests interrupted by a client/ Total number of requests) x 100% Unit: %	≥ 0, ≤ 100	User Bucket API	1 minute
request_code_count	HTTP status code count	Measures the number of requests with status codes returned by the server. For details about the response status codes, see **Table 2-51**. Unit: count	≥ 0 counts	Bucket API HTTP status code	1 minute
api_request_count_per_second	API request TPS	Average number of specific API requests sent to all buckets and objects of a tenant per second within a statistical period. For details about the supported APIs, see **Table 2-50**.	≥ 0 counts	Bucket API	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
request_count_monitor_2XX	2xx Status Codes	Count of server responses to requests whose status codes are 2*xx*. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_monitor_3XX	3xx Status Codes	Count of server responses to requests whose status codes are 3*xx*. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
download_bytes	Total Download Bandwidth	Total size of objects downloaded per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
download_bytes_extranet	Download Bandwidth (Internet)	Total size of objects downloaded over the Internet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
download_bytes_intranet	Download Bandwidth (Intranet)	Total size of objects downloaded over the Intranet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
upload_bytes	Total Upload Bandwidth	Total size of objects uploaded per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
upload_bytes_extranet	Upload Bandwidth (Internet)	Total size of objects uploaded over the Internet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
upload_bytes_intranet	Upload Bandwidth (Intranet)	Total size of objects uploaded over the Intranet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
download_traffic	Total Download Traffic	Total size of objects downloaded in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
download_traffic_extranet	Download Traffic (Internet)	Total size of objects downloaded over the Internet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
download_traffic_intranet	Download Traffic (Intranet)	Total size of objects downloaded over the Intranet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
upload_traffic	Total Upload Traffic	Total size of objects uploaded in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
upload_traffic_extranet	Upload Traffic (Internet)	Total size of objects uploaded over the Internet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
upload_traffic_intranet	Upload Traffic (Intranet)	Total size of objects uploaded over the Intranet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute

Table 3-3 OBS metrics (for storage)

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
capacity_total	Total Used Storage Space	Measures the storage space occupied by all data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_standard	Used Space - Standard Storage	Measures the storage space occupied by Standard data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_infrequent_access	Used Space - Warm Storage	Measures the storage space occupied by Warm data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_archive	Used Space - Cold Storage	Measures the storage space occupied by Cold data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
object_num_all	Total Number of Objects	Measures the total number of objects (including folders and all file versions) stored in all storage classes. Unit: count	≥ 0	User Bucket	30 minutes
object_num_standard_total	Number of Objects - Standard Storage	Measures the total number of objects (including folders and all file versions) stored in the Standard storage class. Unit: count	≥ 0	User Bucket	30 minutes
object_num_infrequent_access_total	Number of Objects - Warm Storage	Measures the total number of objects (including folders and all file versions) stored in the Warm storage class. Unit: count	≥ 0	User Bucket	30 minutes

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
object_num_archive_total	Number of Objects - Cold Storage	Measures the total number of objects (including folders and all file versions) stored in the Cold storage class. Unit: count	≥ 0	User Bucket	30 minutes

3.1.7 Can Folders in OBS Be Used the Same Way as in a File System?

No.

OBS does not involve files or folders like in a file system. For your convenience, OBS provides a way to simulate folders. On OBS Console, you can simulate a folder by adding a slash (/) to the name of an object, which is then displayed as a folder.

3.1.8 Where Is Data Stored in OBS?

When creating a bucket on OBS, you can specify a region for the bucket. Then your data on OBS is stored on multiple storage devices in this region.

3.1.9 Does OBS Support Access over HTTPS?

Yes, OBS can be accessed over HTTPS.

- When accessing OBS using the allocated domain name, just replace **http** in the URL of the bucket or object with **https** in the browser.

3.1.10 Can Other Users Access My Data Stored in OBS?

Yes.

- Bucket ACLs and bucket policies can be used to grant other users read access to your buckets.
- You can grant other users read permissions for objects in your bucket by configuring object ACLs, object policies, or bucket policies.

3.1.11 Does OBS Support Resumable Transfer?

Resumable transfer is supported for all transfer methods except API.

Table 3-4 Support for resumable transfer by different OBS tools

OBS Tool	Resumable Data Transfer
OBS Console	Not supported
OBS Browser+	Supported
obsutil	Supported
APIs	Not supported

3.1.12 Does OBS Support Batch Upload?

The following table lists the batch upload support for different OBS tools.

Table 3-5 Support for batch upload by different OBS tools

Tool	Batch Upload
OBS Console	OBS Console allows you to upload files in a batch. Up to 100 files can be uploaded at a time, with the total size of no more than 5 GB. For details, see **Uploading an Object**.
OBS Browser+	Supports batch upload of files and folders. A maximum of 500 files or folders can be uploaded at a time.
obsutil	Supports upload of a single folder with the maximum size of 48.8 TB.
APIs	Not supported

3.1.13 Does OBS Support Batch Download?

The following table lists the batch download support for different OBS tools.

Table 3-6 Support for batch download by different OBS tools

Tool	Batch Download
OBS Console	Not supportedSupported
OBS Browser+	Supported
obsutil	Supported
APIs	Not supported

3.1.14 Does OBS Support Batch Deletion of Objects?

The following table lists the batch deletion support for different OBS tools.

Table 3-7 Support for batch deletion by different OBS tools

Tool	Batch Deletion
OBS Console	Supported. A maximum of 100 objects can be deleted at a time. If you are deleting a folder, only one folder can be deleted at a time.
OBS Browser+	Supported. Files and folders can be deleted in a batch, and the number of files and folders to be deleted is not limited.
obsutil	You can delete objects in batches by prefix.
APIs	Supported. A maximum of 1,000 objects can be deleted at a time.

NOTE

The batch deletion performance is negatively correlated with the number of objects in a single request. When it comes to QPS, deleting *N* objects is counted as *N* operations. If a large number of objects named with prefixes in lexicographic order are deleted, lots of requests may be concentrated in a specific partition, which results in hot access. This limits the request rate in the hot partition and increases access latency.

To address this problem, you can reduce the number of objects in a single batch deletion request, initiate more concurrent requests, and name objects with random prefixes.

3.1.15 What Are the Factors That Affect Upload and Download Speeds of OBS?

The OBS upload and download speeds may be affected by:

- The default upper limit of the OBS read/write bandwidth allowed for a single account: 16 Gbit/s (which means the total GET and PUT bandwidths over both public and private networks)

  If the actual bandwidth reaches this upper limit, flow control will be triggered.
- Bandwidth of the purchased VM NIC

  If the NIC bandwidth is lower than 16 Gbit/s, the node bandwidth will be limited by the VM bandwidth. You need to purchase multiple VMs to run concurrently to reach 16 Gbit/s.
- Disk I/O and resources consumed by other processes

3.1.16 Why Did Some of My Data Stored on OBS Get Lost?

- Check whether there is a lifecycle rule configured to automatically delete objects after a certain date.
- Check whether the write permission to the bucket has been granted to other users. If it was, those other users can delete objects from the bucket. If you have enabled logging, you can check the logs to find out who deleted the objects.

3.1.17 Can Deleted Data Be Recovered?

- If versioning is enabled for a bucket, deleted objects are saved to the **Deleted Objects** list. You can recover objects from the **Deleted Objects** list. For details, see **Undeleting an Object**.
- If versioning is not enabled, deleted objects cannot be recovered.

3.1.18 Will There Be Data Left Over in OBS After I Delete an Object?

After you select the objects that you want to delete, OBS will delete the data completely, with nothing remaining. This protects against data leaks.

3.1.19 Will My Bucket Performance Be Affected by Other Users' Services?

No. OBS isolates the access from different accounts, so there is no performance interference or impact between different accounts.

3.2 Access Control

3.2.1 How Can I Control Access to OBS?

You can use the following mechanisms to control access to OBS.

- IAM policies

  IAM policies define the actions that can be performed on your cloud resources, specifying what actions are allowed or denied.

  IAM policies can be used to grant access to various IAM users under the same parent account.

  The process is as follows:

  a. Create a user group and select an IAM permission set for it.

  b. Create an IAM user and add it to the user group, and it will inherit the permissions of the user group you added it to.

- Bucket policies

  A bucket policy applies to the configured OBS bucket and all the objects in the bucket. An OBS bucket owner can use a bucket policy to grant permissions on buckets and objects in the buckets to IAM users or other accounts.

- Access Control List (ACL)

  ACLs control read and write permissions for accounts. ACL control is not as fine-grained as bucket policies and IAM policies, so IAM policies and bucket policies are recommended instead.

3.2.2 What Are the Differences Between Using an IAM Policy and a Bucket Policy in Access Control?

IAM policies apply to cloud resources. With the OBS permissions, an IAM policy can be applied to all buckets and objects, or it can be applied only to specified buckets and objects. IAM policies are recommended if you assign permissions to IAM users of the same account.

A bucket policy only applies to the bucket the policy was configured for.

3.2.3 What Is the Relationship Between a Bucket Policy and an Object Policy?

An object policy takes effect on only one object in a bucket. A bucket policy can be applied to multiple or all objects in a bucket.

3.3 Buckets and Objects

3.3.1 Why Am I Unable to Create a Bucket?

- If the number of created buckets has reached 100 (the maximum number allowed), delete some unneeded buckets and try again.
- If the new bucket name already exists, use another one and try again. Each OBS bucket name must be globally unique. Specifically, it must be different from that of buckets created by its owner or by any other users.
- The name of a deleted bucket cannot be reused immediately after the deletion. It can be reused for a bucket or a parallel file system at least 30 minutes later after the deletion.
- Check whether the account has required permissions. If not, grant the account permissions as needed.
- Check whether the account is in arrears or the account balance is insufficient. If this is the case, pay off the outstanding balance or top up the account.
- Check whether the network connectivity between the local computer and OBS is normal. If the network is faulty, restore the network connection.
- If the failure is not caused by any of the described reasons, you can check the returned error code to try to find the reason.

3.3.2 Why Am I Unable to Upload an Object?

- Check whether the network connectivity between the local computer and OBS is normal. If the network is faulty, restore the network connection.
- If a message indicating "service unavailable" is displayed when objects are being uploaded, try again later.
- Check whether the account is in arrears or the account balance is insufficient. If this is the case, pay off the outstanding balance or top up the account.
- Check whether the account has the permissions required to upload objects. This check should cover the IAM policies, bucket policies, and bucket ACLs. If

the account does not have the required permissions, grant the permissions first.

- If the fault persists, contact customer service.

3.3.3 Why Am I Unable to Download an Object?

- Check whether the network connectivity between the local computer and OBS is normal. If the network is faulty, restore the network connection.
- Check whether the account is in arrears or the account balance is insufficient. If this is the case, pay off the outstanding balance or top up the account.
- Check whether the account has the permissions needed to download objects from the bucket. This check should cover IAM policies, bucket policies, object policies, bucket ACLs, and object ACLs. If the account does not have the required permissions, grant the permissions first.
- Check whether the object is in the Cold storage class. If it is and the status is **Unrestored**, restore the object first.
- If the fault persists, contact customer service.

3.3.4 Why Can't I Delete a Bucket?

- Check whether the network connectivity between the local computer and OBS is normal. If the network is faulty, restore the network connection.
- Check whether all objects in the bucket have been deleted. If no, delete all objects from the bucket.
- Check whether all fragments in the bucket have been deleted. If no, delete all fragments from the bucket.
- If versioning is enabled, check whether there are deleted objects remaining in the bucket. If yes, permanently delete all deleted objects from the bucket.
- Check whether the account that deletes the bucket is the owner of the bucket.
- If the fault persists, contact customer service.

3.3.5 What Is the Relationship Between Bucket Storage Classes and Object Storage Classes?

You can specify the storage class for a bucket when creating the bucket. You can also change the storage class of a bucket after the bucket is created.

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

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

3.3.6 Can I Modify the Region of a Bucket?

No. After a bucket is created, the region cannot be changed.

3.3.7 How Do I Obtain the Access Path to an Object?

Object access paths use the following format: **https://***{bucket name}*.*{domain name}*/*{object name}*.

You can combine a path manually or use the tools in the following table to obtain it.

Table 3-8 How to obtain an object URL

Tool	Object URL
OBS Console	Click the object and copy the URL for the detailed information of the object.
OBS Browser+	Click the **Attribute** button of the object and then you can copy the URL displayed in the detailed information about the object.
obsutil	Not supported
APIs	Not supported

NOTE

If the object access path is user-assembled, you need to escape the object name by referring to the URL encoding rules.

3.3.8 Why Can't I Search for Certain Objects in My Bucket?

On OBS Console and OBS Browser+, you can search for objects by object name prefix. For example, if you search for **test**, you will find all objects whose name starts with **test**. However, if the keyword entered is in the middle or at the end of the object name, the search will not return those results. For example, you want to search for **testabc** and you enter **abc** in the search box, **testabc** will not be found. Only objects whose name starts with the prefix **abc** will be found.

3.3.9 What Do I Do If I Encounter an Error While Trying to Access an Object URL with Full-Width Characters Using Internet Explorer?

Problem Description

HTTP 400 error is returned when using the Internet Explorer to access an object URL that contains Chinese characters?

Handling Method

By default, the Internet Explorer does not use the UTF-8 to send query strings. To solve this problem, change the default configuration of the Internet Explorer.

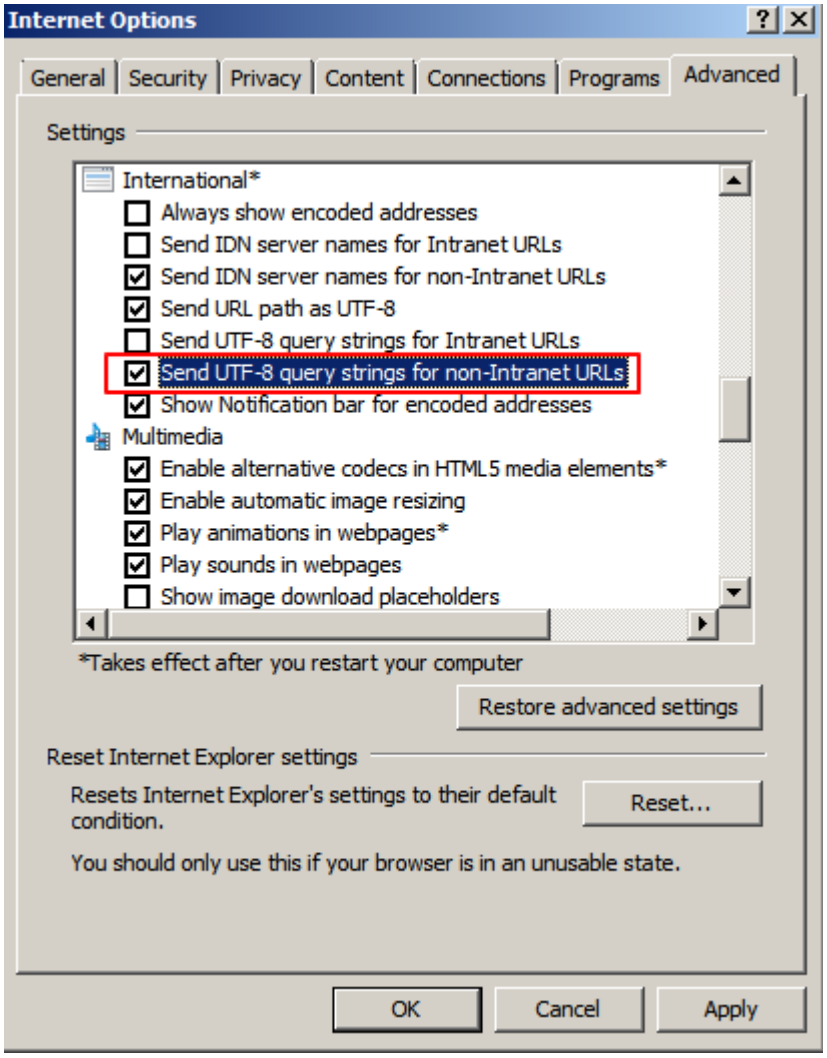
Procedure

Step 1 Open Internet Explorer, for example, IE 11.

Step 2 Click **Settings** in the upper right corner of the browser and choose **Internet Options** > **Advanced**.

Step 3 Select **Send UTF-8 query string for non-Internet URLs**, as shown in the following figure.

Figure 3-1 Changing IE default settings



Step 4 Click **Apply**, and then click **OK**.

Step 5 Restart Internet Explorer.

Then, you can properly access the object URL.

----End

3.4 Tools

3.4.1 When Downloading a Folder Using obsutil, the Download Speed Slows After the Folder Download Progress Reaches 90%

This problem may occur in the following scenarios:

- Scenario 1: The folder contains a few large objects among a large number of small objects. Large objects are downloaded at fast speed. But the download speed of small objects in large quantity is closely related to the TPS performance. Therefore, if the remaining 10% are mostly small objects, the download speed may decrease.
- Scenario 2: The folder contains same-size objects. It is possible that all objects have been downloaded but are queuing to be written to disks, which may be reflected as a slowdown in the download progressing. In this case, check the writing speed of your clients.

3.4.2 With obsutil, Downloading a File Fails After the Download Progress Reaches 99%

Possible causes:

1. Network fluctuation
2. Failure in caching the file to the target folder due to disk I/O freezes.

Solution:

1. Run the download command again.

   The resumable download function is enabled by default for obsutil download tasks. You only need to run the same download command again, the failed file download will be resumed and the file will be downloaded to your local path.
2. If the problem persists, upgrade obsutil to the latest version and try again.

3.4.3 How Do I Use the obsutil cp Command to Enable Incremental Upload, Download, or Replication?

When running the **obsutil cp** command to upload or download data, you can add the **-u** parameter to enable the incremental upload/download function.

This parameter indicates that the system will compare the source path with the target path when uploading, downloading, or replicating an object. The system uploads, downloads, or replicates an object only when the target object does not exist, the object size is inconsistent, or the last modification time of the target object is earlier than that of the source object.

3.5 APIs and SDKs

3.5.1 What Are the Differences Between PUT and POST Upload Methods?

Parameters are passed through the request header if the PUT method is used to upload objects; if the POST method is used to upload objects, parameters are passed through the form field in the message body.

With the PUT method, you need to specify the object name in the URL, but object name is not required with the POST method, which uses the bucket domain name as the URL. Request lines of these two methods are given as follows:

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

Either PUT or POST method allows the object size of [0, 5 GB] for each upload. If you need to upload an object greater than 5 GB, use the multipart upload method.

For details about PUT and POST APIs, see the *Object Storage Service API Reference*.

3.5.2 Failure with OBS SDK in Uploading a File Greater than 5 GB

OBS server has a restriction on the object upload API, which only allows a maximum of 5 GB for an upload. If you want to upload a file greater than 5 GB, use the multipart upload API. Operations are detailed in the following procedure:

1. Call the OBS API for initializing a multipart upload task to generate a multipart upload ID (Upload ID).
2. Call the OBS API for uploading parts one by one or in parallel. The size of each part can be up to 5 GB.
3. After parts are uploaded, call the OBS API to merge parts to generate the complete object.

3.5.3 Why Don't the Signatures Match?

Symptom

The following error is reported during an OBS API call.

Status code: 403 Forbidden

Error code: SignatureDoesNotMatch

Error message: The request signature we calculated does not match the signature you provided. Check your key and signing method.

Possible Causes

The provided signature does not match the signature calculated by the system.

Solution

Step 1 Check the AK and SK.

Ensure that the AK and SK you entered are correct, so they can match those used in the request.

Step 2 Check **HTTP-Verb**.

Ensure that the **HTTP-Verb** in the signature is the same as that in the request.

Step 3 Check **Date** and **Expires**.

- Signature in a header: Check whether the **Date** in the signature is the same as that in the request header.
- Signature in a URL: Check whether the **Expires** in the signature is the same as that in the request URL.

Step 4 Check headers.

Check **Content-MD5**, **Content-Type**, and **Canonicalized Headers**. If any of them are contained during signature calculation, they must be also contained in the request.

NOTE

If a URL with a signature contained is used to access OBS resources through a browser, the header parameters above cannot be contained during signature calculation.

Step 5 Check **Canonicalized Resource**.

Canonicalized Resource indicates the OBS resources that are requested. Configure this parameter based on the requirements in the API reference. For details, see section "Authentication of Signature in a Header" or "Authentication of Signature in a URL" in the *Object Storage Service API Reference*.

Step 6 Check **StringToSign**.

Check whether **StringToSign** is constructed based on the following rules:

- Signature in a header:
  ```
  HTTP-Verb + "\n" + Content-MD5 + "\n" + Content-Type + "\n" + Date + "\n" + CanonicalizedHeaders + CanonicalizedResource
  ```
- Signature in a URL:
  ```
  HTTP-Verb + "\n" + Content-MD5 + "\n" + Content-Type + "\n" + Expires + "\n" + CanonicalizedHeaders + CanonicalizedResource
  ```

NOTE

If a parameter is left blank, put it in a new line.

Step 7 Check the signature calculation.

Check whether the signature is calculated as follows:

1. Construct the request string **StringToSign**.
2. Perform UTF-8 encoding on the result in the **1**.
3. Use the SK to perform the HMAC-SHA1 signature calculation on the result in **2**.

4. Perform Base64 encoding on the result in **3**. If the signature is contained in a header, this step generates the final signature and no further actions are required.
5. If the signature is contained in a URL, perform the URL encoding on the result in **4** to obtain the final signature.

----End

3.6 Security

3.6.1 How Is Data Security Ensured in OBS?

OBS is secure. It provides end-to-end security services. For example, if a bucket or an object is undisclosed when you access the bucket or object, only the owner of the bucket or object can access it. Further, the access to the bucket or object requires access keys (AK/SK). You can also use various access control mechanisms (such as bucket policies and ACLs) to select users and user groups and grant them permissions. OBS supports data transfer over the HTTPS/SSL protocol. Data encryption prior to upload is available to meet your higher security requirements.

3.6.2 Does OBS Scan My Data for Other Purposes?

OBS only determines whether data blocks exist or are damaged (repairs data if damaged) by scanning for the data. It does not read specific data.

3.6.3 Can Engineers Export My Data from the Background of OBS?

No. Background engineers cannot export your data. For example, if a bucket or an object is undisclosed when you access the bucket or object, only the owner of the bucket or object can access it. Further, the access to the bucket or object requires access keys (AK/SK).

3.6.4 How Does OBS Protect My Data from Being Stolen?

Only the owner of a bucket or an object can access it. Accessing a bucket or object requires access keys (AK/SK). In addition, multiple access control mechanisms such as the ACLs, bucket policies, and URL validation are used to ensure data access security.

3.6.5 Can a Pair of AK and SK Be Replaced When It Is Being Used to Access OBS?

Yes. The pair of AK and SK can be replaced at any time.

3.6.6 Can Multiple Users Share One Pair of AK and SK to Access OBS?

Yes. Different users can use the same pair of AK and SK to access the same resources in OBS.

3.7 Fragment Management

3.7.1 Why Are Fragments Generated?

Fragments are incomplete data in buckets generated due to data upload failures.

Data can be uploaded to OBS using multipart uploads. There will be fragments generated, if a multipart upload fails because of the following causes (included but not limited to):

- The network is in poor conditions, and the connection to the OBS server is interrupted frequently.
- The upload task is manually suspended.
- The device is faulty.
- The device is powered off suddenly.

3.7.2 How Do I Manage Fragments?

Generated fragments take up storage space that is billable.

You can clear the fragments in a bucket on OBS Console or OBS Browser+.

If fragments are generated due to interruptions to multipart uploads on OBS Browser+, the fragments will disappear once those multipart uploads are continued and finished.

3.8 Versioning

3.8.1 Can I Upload Objects with the Same Name to the Same Folder?

When versioning is enabled for a bucket, OBS automatically assigns a unique version ID to each object uploaded to the bucket. Objects with the same name are stored in OBS with different version IDs.

If versioning is not enabled for a bucket, a newly uploaded object in the folder will overwrite the previously uploaded object with the same name.

3.8.2 Can I Recover a Deleted Object?

When versioning is enabled, if you delete an object without specifying a version ID, the object is tagged with a delete marker and displayed in the list of **Deleted Objects**. You can recover the object from that list.

If you delete an object with a version ID specified when versioning is enabled or you delete an object when versioning is not enabled, OBS permanently deletes the object, and you cannot recover it.

For details, see **Versioning Overview**.

3.9 Event Notifications

3.9.1 Which Events Can Trigger Event Notifications?

OBS supports notification for the following event types:

- **ObjectCreated**: all kinds of object creation operations, including PUT, POST, COPY, and part assembling
  - **Put**: Creates or overwrites an object using the PUT method.
  - **Post**: Creates or overwrites an object using the POST (browser-based upload) method.
  - **Copy**: Creates or overwrites an object using the COPY method.
  - **CompleteMultipartUpload**: Assembles parts of a multipart upload.
- **ObjectRemoved**: Deletes an object.
  - **Delete**: Deletes an object with a specified version ID.
  - **DeleteMarkerCreated**: Deletes an object without specifying a version ID.

3.10 How Do I Use Lifecycle Management?

3.10.1 What Are the Application Scenarios of Lifecycle Management?

You may configure lifecycle rules to:

- Periodically delete logs that are only meant to be retained for a specific period of time (a week or a month).
- Transition documents that are seldom accessed to the Warm or Cold storage class or delete them.

If you want to delete a large number of objects from a bucket, you can configure a lifecycle rule to automatically delete the expired objects. **Table 3-9** lists the parameters for configuring such a lifecycle rule on OBS Console.

Table 3-9 Parameters for deletion upon expiration

Parameter	Value
Status	Enable
Rule Name	Example: **rule-delete**
Applies To	You can apply the deletion rule to the entire bucket or to objects that share the same name prefix in the bucket.

Parameter	Value
Current Version	Expiration Time Days: 1
Historical Version	Expiration Time Days: 1

One day later, objects in the bucket are successfully deleted based on the rule. If you no longer need this lifecycle rule, you can disable it or delete it.

3.11 How Do I Use Static Website Hosting?

3.11.1 Can I Host My Static Websites on OBS?

OBS supports static website hosting. You can configure the static website hosting function for your buckets on OBS Console. When a client accesses objects from the website address of a bucket, the browser can directly resolve the web resources and present them to end users.

3.11.2 Which Types of Websites Can I Use OBS to Host?

Static websites contain static web pages and some scripts that can run on clients, such as JavaScript and Flash.

3.11.3 How Do I Obtain the Static Website Hosting Address of a Bucket?

You can obtain the static website hosting address of the bucket on OBS Console.

You can also get the address according to the following rule and format. Address format: https://*Bucket name.Domain name of the hosted static website*

