Object Storage Service

User Guide (Ankara Region)

Issue	01
Date	2024-04-15

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

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

Notice

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

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

Huawei Cloud Computing Technologies Co., Ltd.

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

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

Contents

1 Service Overview

1.1 About OBS

Object Storage Service (OBS) is a scalable service that provides secure, reliable cloud storage for massive amounts of data. On OBS, you can easily manage your OBS resources, such as creating, modifying, and deleting buckets, or uploading, downloading, and deleting objects.

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 Browser+ 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.

Cloud service infrastructures can be deployed in multiple regions, delivering high scalability and reliability. You can deploy OBS in specific regions for faster access.

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 1-1** compares OBS with on-premises storage servers.

Table 1-1 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. 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.
Reliability	The OBS five-level reliability architecture ensures 99.9999999999% of durability and 99.95% of continuity for multi-AZ storage, much higher than those of the conventional architecture.	Due to limited investment, on-premises storage servers cannot ensure reliability at all levels of media, servers, cabinets, data centers, and regions. Once there is a failure or disaster, it may cause irreversible data loss to enterprises.
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 hundreds of millions of users.
- **Multi-level protection and authorization management**: Measures, including versioning, URL validation, virtual private cloud (VPC)-based network isolation, 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 and data migration tools 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

- OBS is built for you to store and retrieve any amount of data anytime, anywhere. It is a good data storage choice for mobile, web, and application developers. OBS also reduces costs in nearline and offline storage, such as backup, big data storage, and archiving.
- OBS is linearly scalable, highly reliable, and secure (end-to-end security for access, transfer, and storage). Thanks to its scalability, as businesses continue to grow, developers can focus on business innovations, instead of underlying storage technologies. Most importantly, this greatly reduces IT costs.

OBS can be used for video surveillance, video on demand (VOD), backup and archive, high-performance computing (HPC), mobile Internet, enterprise cloud boxes (web disks), and many other scenarios.

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.

You can grant users permissions by using roles or policies.

- Roles: A type of coarse-grained authorization mechanism that provides only a limited number of service-level roles. When using roles to grant permissions, you also need to assign dependency roles. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization for secure access control. For example, you can grant OBS users only the permissions for managing a certain type of OBS resources. Most policies define permissions based on APIs.

NOTE

Due to data caching, a role and policy involving OBS actions will take effect 10 to 15 minutes after it is attached to a user and a user group.

Table 1-2 lists all system permissions of OBS.

Table 1-2 OBS system permissions

Role/Policy Name	Description	Type	Dependency
Tenant Administrator	Allows you to perform all operations on all services except IAM.	System-defined role	None
Tenant Guest	Allows you to perform read-only operations on all services except IAM.	System-defined role	None
OBS Administrator	Allows you to perform any operation on all OBS resources under the account.	System-defined policy	None
OBS ReadOnlyAccess	Allows you to list buckets, obtain basic bucket information and bucket metadata, and list objects (excluding versioned objects).	System-defined policy	None
OBS OperateAccess	Allows you to perform all operations defined in OBS ReadOnlyAccess and to perform basic object operations, such as uploading objects, downloading objects, deleting objects, and obtaining object ACLs.	System-defined policy	None

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

Table 1-3 Permissions and the allowed operations on OBS resources

Operation	Tenant Administrator	Tenant Guest	OBS Administrator	OBS ReadOnly Access	OBS OperateAccess
Listing buckets	Supported	Supported	Supported	Supported	Supported
Creating buckets	Supported	Not supported	Supported	Not supported	Not supported
Deleting buckets	Supported	Not supported	Supported	Not supported	Not supported
Obtaining basic bucket information	Supported	Supported	Supported	Supported	Supported
Controlling bucket access	Supported	Not supported	Supported	Not supported	Not supported
Managing bucket policies	Supported	Not supported	Supported	Not supported	Not supported
Listing objects	Supported	Supported	Supported	Supported	Supported
Listing objects with multiple versions	Supported	Supported	Supported	Not supported	Not supported
Uploading files	Supported	Not supported	Supported	Not supported	Supported
Creating folders	Supported	Not supported	Supported	Not supported	Supported
Deleting objects	Supported	Not supported	Supported	Not supported	Supported
Deleting folders	Supported	Not supported	Supported	Not supported	Supported
Downloading objects	Supported	Supported	Supported	Not supported	Supported
Deleting object versions	Supported	Not supported	Supported	Not supported	Supported

Operation	Tenant Administrator	Tenant Guest	OBS Administrator	OBS ReadOnlyAccess	OBS OperateAccess
Downloading object versions	Supported	Supported	Supported	Not supported	Supported
Undeleting objects	Supported	Not supported	Supported	Not supported	Supported
Deleting fragments	Supported	Not supported	Supported	Not supported	Supported
Controlling object access	Supported	Not supported	Supported	Not supported	Not supported
Configuring object metadata	Supported	Not supported	Supported	Not supported	Not supported
Obtaining object metadata	Supported	Supported	Supported	Not supported	Supported
Managing versioning	Supported	Not supported	Supported	Not supported	Not supported
Managing logging	Supported	Not supported	Supported	Not supported	Not supported
Managing event notifications	Supported	Not supported	Supported	Not supported	Not supported
Managing lifecycle rules	Supported	Not supported	Supported	Not supported	Not supported
Managing static website hosting	Supported	Not supported	Supported	Not supported	Not supported
Managing CORS rules	Supported	Not supported	Supported	Not supported	Not supported
Managing URL validation	Supported	Not supported	Supported	Not supported	Not supported
Managing domain names	Supported	Not supported	Supported	Not supported	Not supported

Operation	Tenant Administrator	Tenant Guest	OBS Administrator	OBS ReadOnlyAccess	OBS OperateAccess
Managing cross-region replication	Supported	Not supported	Supported	Not supported	Not supported
Appending data to objects	Supported	Not supported	Supported	Not supported	Supported
Configuring object ACL	Supported	Not supported	Supported	Not supported	Not supported
Configuring the ACL for an object version	Supported	Not supported	Supported	Not supported	Not supported
Obtaining object ACL information	Supported	Supported	Supported	Not supported	Supported
Obtaining the ACL of a specific object version	Supported	Supported	Supported	Not supported	Supported
Initiating a multipart upload	Supported	Not supported	Supported	Not supported	Supported
Listing uploaded parts	Supported	Supported	Supported	Not supported	Supported
Canceling multipart uploads	Supported	Not supported	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 1-4 OBS use restrictions

Restriction Item	Description
Capacity utilization for small files	A small file defined by OBS is an object smaller than 200 KB. A small file stored in OBS will occupy more physical storage space than its actual size. This results in low space utilization. A single OBS node supports only a limited number of objects. In massive storage scenarios, smaller files will result in a lower space utilization. To store a large number of small files, you are advised to plan the storage capacity and storage nodes separately to prepare for capacity expansion.
Access rule	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.

1.6 Using OBS

You can use the following tools to access and manage OBS resources:

Table 1-5 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 or Mac 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.

1.7 Related Services

Table 1-6 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** **Creating an Agency**
SMN sends OBS related alarms and event notifications, and triggers workflows.	Simple Message Notification (SMN)	**SMN-Enabled Event Notifications**

OBS can be used as the storage resource pool for other cloud services such as Image Management Service (IMS).

1.8 Basic Concepts

1.8.1 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 is uniquely identified by a key within a bucket.
- 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 for processing objects. 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.

Generally, objects are managed as files. However, OBS is an object-based storage service and there is no concept of files and folders. For easy data management, OBS provides a method to simulate folders. By adding a slash (/) 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**.

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

1.8.2 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, and the region. You can specify access permissions, and regions when creating buckets. You can also configure advanced attributes to meet storage requirements in different scenarios.

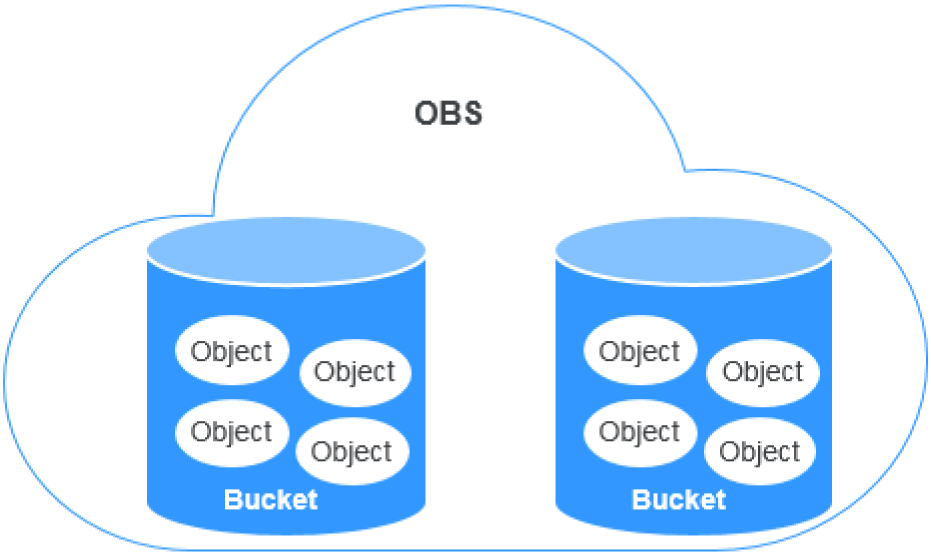
Each bucket name in OBS is globally unique and cannot be changed after the bucket is created. The region where a bucket resides cannot be changed once the bucket is created. When you create a bucket, OBS creates a default access control list (ACL) that grants users permissions (such as read and write 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 1-1 illustrates the relationship between buckets and objects in OBS.

Figure 1-1 Relationship between objects and buckets



1.8.3 Parallel File System

Parallel File System (PFS) is a high-performance semantic file system provided by OBS. It features access latency in milliseconds, TB/s-level bandwidth, and millions of IOPS, which makes it ideal for processing high-performance computing (HPC) workloads.

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

1.8.4 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

You can create a pair of permanent AK and SK on the **My Credentials** page.

- 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 1-7 OBS resource management tools

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

1.8.5 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. Obtain your required region and endpoint information from the administrator.

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

Table 1-8 OBS endpoints

Region Name	Region	Endpoint	Protocol
TR-Ankara-PUR	tr-central-201	obs.tr-central-201.hc.vodafone.com.tr	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 1-9 lists the bucket domain name and other domain names in OBS, including their structure and protocols.

Table 1-9 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 1-8**.	HTTPS HTTP

Type	Structure	Description	Protocol
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
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

1.8.6 Region and AZ

Concept

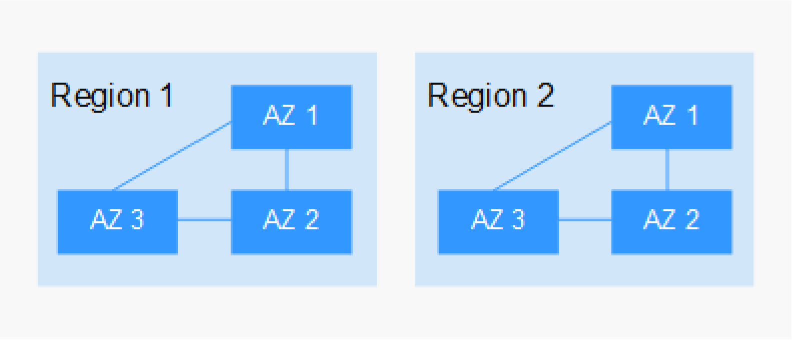
A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

- 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 1-2 shows the relationship between regions and AZs.

Figure 1-2 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. Obtain the regions and endpoints from the enterprise administrator.

2 Configuration Before Use

2.1 Configuring the Local hosts File

Configure DNS or the local **hosts** file before you can use the basic functions of OBS.

NOTE

No matter whether you are using OBS buckets or parallel file systems, configurations in this section are required.

Scenarios

End users usually access OBS in the scenarios listed in **Table 2-1**.

Table 2-1 OBS access scenarios

Access Method	Required Configuration	Operated By
Local network (IDC without a DNS server) outside the cloud	**Configuring the local hosts file**	End users

Configuration Description

All configurations below are examples under the following scenario assumption. Actual configurations are subject to the actual information.

Scenario Assumption

The global domain name of a cloud service is **huaweicloud.com**. OBS is deployed in two regions (with the region IDs of **region1** and **region2**).

- **region1** is the default region, and the IP address of its default cluster **lz01** is **192.168.0.1**.
- **region2** is a non-default region, and the IP address of its default cluster **lz01** is **192.168.0.2**.

Scenario: Configuring the Local hosts File

To access OBS (without a DNS server) over a local network, you need to add the following information to the local **hosts** file:

Note: Each time you create a bucket, you need to add two records for the bucket in the hosts file. One is for the domain name with the region information and the other is for the domain name without the region information. For details, see the example below.

```
192.168.0.1 obs.huaweicloud.com
192.168.0.1 obs.region1.huaweicloud.com
192.168.0.2 obs.region2.huaweicloud.com
192.168.0.1 obsbrowser.obs.region1.huaweicloud.com # This configuration is required before downloading OBS Browser+.
192.168.0.1 bucket1.obs.region1.huaweicloud.com
192.168.0.2 bucket2.obs.region2.huaweicloud.com
192.168.0.1 bucket1.obs.huaweicloud.com
192.168.0.2 bucket2.obs.huaweicloud.com
......
```

Configure the following items when static website hosting is required:

```
192.168.0.1 bucket1.obs-website.region1.huaweicloud.com
192.168.0.2 bucket2.obs-website.region2.huaweicloud.com
......
```

NOTE

1. In this example, **huaweicloud.com** (global domain name), **region1** and **region2** (region IDs), **bucket1** and **bucket2** (bucket names), as well as **192.168.0.1** and **192.168.0.2** (IP addresses) are all examples. Replace them with the actual information.
   - The first line is the OBS global domain name, in which the IP address is the OBS global IP address (the IP address of the default cluster **lz01** in the default region). To obtain this IP address, log in to CloudAutoDeploy-EDK in the default region, and choose **Project Management** > **Projects**. Click **Export Deliverable** on the right of the desired project. Decompress the exported package, and open the **obs_lld** file in the **obs** directory. In this file, find the value of **virtual_ip** in the **OM IP /27** column of the **lz01** region's LVS node.
   - The lines after the first line record the domain name of each region, with the IP address of the region's default cluster. To obtain this IP address, log in to CloudAutoDeploy-EDK in each region, and choose **Project Management** > **Projects**. Click **Export Deliverable** on the right of the desired project. Decompress the exported package, and open the **obs_lld** file in the **obs** directory. In this file, find the value of **virtual_ip** in the **OM IP /27** column of the **lz01** region's LVS node.
2. Path of the **hosts** file:
   - Windows: C:\Windows\System32\drivers\etc\hosts
   - Linux: /etc/hosts
3. In addition to the first three lines for global and regional domain configuration, you also need to add the access domain name of each bucket (like **bucket1** or **bucket2** in the example) to be accessed to the **hosts** file. Replace the bucket and region configuration (including the region ID and default cluster IP address) with the actual information. Two records should be configured for each bucket, one for the domain name with region information and the other for the domain name without region information.
4. The fourth line is the configuration required for downloading OBS Browser+. For a successful download, you need to add the domain name of the bucket that stores the OBS Browser+ software package in the **hosts** file. In the bucket domain names above, **obsbrowser** (bucket name) and **region1** (region ID) are only examples. Replace them with the actual information in the download link of OBS Browser+ on OBS Console.

3 OBS Console Operation Guide

3.1 Console Function Overview

Table 3-1 lists functions provided by OBS Console.

Table 3-1 OBS Console functions

Function	Description
Basic bucket operations	Allow you to create and delete buckets in specified regions (service areas).
Basic object operations	Allow you to manage objects, including uploading (multipart uploads included), downloading, and deleting objects.
Object metadata	Allows you to set properties for objects.
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.
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.

Function	Description
Cross-region replication	Implements object replication across regions under the same account. A cross-region replication rule enables OBS to automatically, asynchronously copy data from a source bucket in one region to a destination bucket in a different region. This provides disaster recovery across regions, catering to your needs for remote backup.
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.
Two-AZ DR	You can enable two-AZ DR for a bucket to store data in the bucket in two AZs for a higher data reliability. Once the two-AZ DR policy is configured, it cannot be modified later.

3.2 Restrictions

Table 3-2 lists the web browser versions compatible with OBS Console.

Table 3-2 Supported web browser versions

Web Browser	Version
Internet Explorer	● Internet Explorer 9 (IE9) ● Internet Explorer 10 (IE10) ● Internet Explorer 11 (IE11)

Web Browser	Version
Firefox	Firefox 55 and later
Chrome	Chrome 60 and later

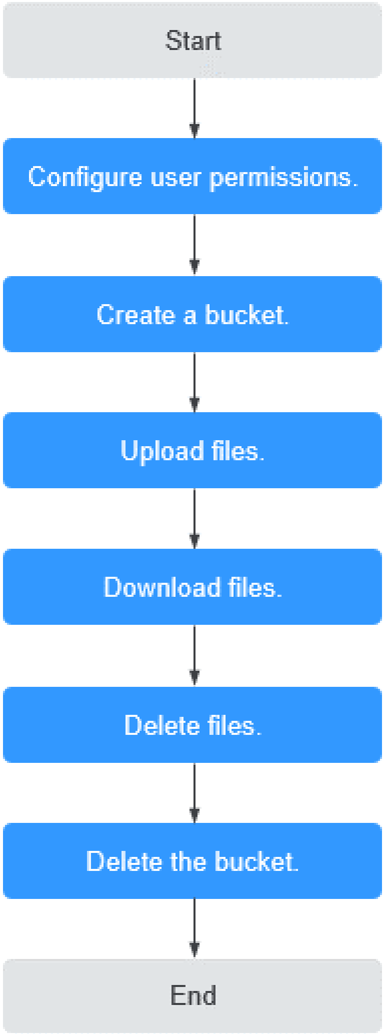
3.3 Getting Started

3.3.1 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 3-1**.

Figure 3-1 OBS Console quick start



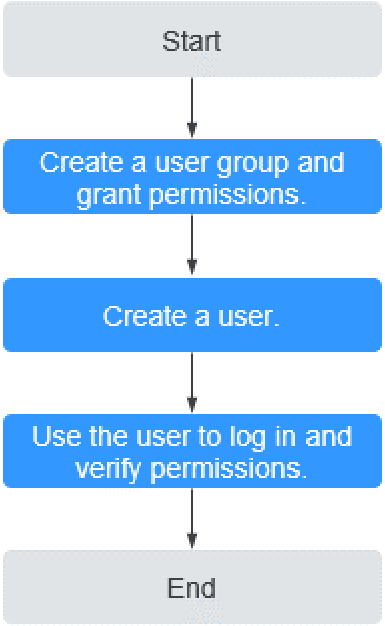
3.3.2 Configuring User Permissions

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 3-2 Process of granting an IAM user the OBS permissions


Start
Create a user group and grant permissions.
Create a user.
Use the user to log in and verify permissions.
End

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 **Authorize** in the **Operation** column of the row.

5. Under **Select Policy/Role**, filter policies based on policy types in the upper right corner, required policy names, and click **Next**.
6. Under **Select Scope**, select **Global services** and click **OK**.

NOTE

In the policy content area, you can view the authorization details.

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 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

3.3.3 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 3-3 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.
AZ	Isolated physical location where a bucket resides. If a region has multiple AZs, you can create buckets in clusters in different AZs and implement cross-AZ disaster recovery using cross-cluster replication. For details, see **Cross-Cluster Replication**.
Cluster Group	Cluster where a bucket resides. OBS provides cross-cluster replication to implement cross-AZ disaster recovery. For details, see **Cross-Cluster Replication**.

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.
Cluster Type	● **Public cluster**: If you select this option, the bucket you are creating will be in a public cluster shared with others. ● **Dedicated cluster**: If you select this option, the bucket you are creating will be in the dedicated cluster you purchased. **NOTE** The cluster type feature is only available to users who have purchased dedicated clusters. For the other users, the bucket they create is in a public cluster by default.
Disaster Recovery	● **Disable**: Data is stored only in one single AZ at a lower cost. ● **Enable**: Data is redundantly stored in two AZs in the same region. This delivers a higher reliability but costs more. Once enabled, the two-AZ disaster recovery cannot be disabled.
Data Redundancy Policy	● **Multi-AZ storage**: Data is stored in multiple AZs to achieve higher reliability. ● **Single-AZ storage**: Data is stored in a single AZ, with lower costs. Once a bucket is created, the data redundancy policy cannot be changed, so choose the policy that can meet your needs.
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.

Step 3 Click **Create Now**.

----End

3.3.4 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 file size exceeds 5 GB, use OBS tools (like OBS Browser+) or the multipart upload of OBS APIs for upload. To download OBS Browser+, you can click the OBS Browser+ package link on the homepage of OBS Console.

Before downloading and using OBS Browser+, you must complete required configurations by following **Configuring the Local hosts File**.

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

- You have completed required configurations by following **Configuring the Local hosts File**.
- 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 to go to the **Objects** page.

Step 2 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.

Step 3 In the **Upload Object** area, drag and drop the files or folders you want to upload.

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

Step 4 (Optional) To configure metadata, click **Next: (Optional) Configure Advanced Settings**.

Add metadata ContentDisposition, ContentLanguage, WebsiteRedirectLocation, ContentEncoding, or ContentType as needed. For more information, see **Object**

Metadata. Metadata is a set of name-value pairs. The metadata value cannot be left blank. You can add two or more metadata entries by clicking **Add**.

Step 5 Click **Upload**.

----End

3.3.5 Downloading an Object

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

Prerequisites

You have completed required configurations by following **Configuring the Local hosts File**.

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

Step 2 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

3.3.6 Deleting an Object

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

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

Step 2 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 3 Click **OK** 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.

3.3.7 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 **OK** to confirm the deletion.

----End

3.4 Managing Buckets

3.4.1 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 3-4 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.
AZ	Isolated physical location where a bucket resides. If a region has multiple AZs, you can create buckets in clusters in different AZs and implement cross-AZ disaster recovery using cross-cluster replication. For details, see **Cross-Cluster Replication**.
Cluster Group	Cluster where a bucket resides. OBS provides cross-cluster replication to implement cross-AZ disaster recovery. For details, see **Cross-Cluster Replication**.
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.
Cluster Type	● **Public cluster**: If you select this option, the bucket you are creating will be in a public cluster shared with others. ● **Dedicated cluster**: If you select this option, the bucket you are creating will be in the dedicated cluster you purchased. **NOTE** The cluster type feature is only available to users who have purchased dedicated clusters. For the other users, the bucket they create is in a public cluster by default.
Disaster Recovery	● **Disable**: Data is stored only in one single AZ at a lower cost. ● **Enable**: Data is redundantly stored in two AZs in the same region. This delivers a higher reliability but costs more. Once enabled, the two-AZ disaster recovery cannot be disabled.

Parameter	Description
Data Redundancy Policy	● **Multi-AZ storage**: Data is stored in multiple AZs to achieve higher reliability. ● **Single-AZ storage**: Data is stored in a single AZ, with lower costs. Once a bucket is created, the data redundancy policy cannot be changed, so choose the policy that can meet your needs.
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.

Step 3 Click **Create Now**.

----End

3.4.2 Viewing Basic Information of a Bucket

On OBS Console, you can view a bucket's details, including basic bucket information, process flows for common scenarios, domain name details, basic configurations, and others.

Viewing Bucket Details

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

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

Step 3 On the top of the page, view the bucket information, including the bucket name, region, and creation time.

Table 3-5 Bucket information

Item	Description
Bucket name	Name of the bucket
Region	Region where the bucket is located
Created	Creation time of the bucket

Step 4 In the **Basic Information** area, view the number of objects, storage usage, bucket version, versioning status, and account ID.

Table 3-6 Parameters in the Basic Information area

Parameter	Description
Objects	The total number of stored folders and objects of all versions in a bucket
Used Capacity	Total storage space occupied by objects of all versions in the bucket **NOTE** If usage statistics is available for the bucket, storage usage data is not displayed here.
Bucket Version	Version number of the bucket.
Versioning	Versioning status
Account ID	Unique identity of the bucket owner. It is the same as **Account ID** on the **My Credentials** page.

Step 5 In the **Process Flow** area, view the process flows for common scenarios. You can click **Change Scenario** in the upper right corner to choose a desired scenario.

In each flow, you can click a node to view relevant details, or click a card to navigate to the operation guide or console page.

Step 6 In the **Domain Information Details** area, view information about the endpoint, access domain name, and static website hosting domain name. You can also perform related operations by clicking buttons in the **Operation** column.

Step 7 In the **Basic Configurations** area, view the bucket's basic configurations, including lifecycle rules, static website hosting, and CORS rules. You can click a card to make required configurations.

----End

Exporting a Bucket List

Step 1 Go to the bucket list.

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

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

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

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

----End

3.4.3 Searching for a Bucket

On OBS Console, you can search for buckets by bucket name, and region.

NOTE

The keywords used for search are case-insensitive.

Procedure

Step 1 Click the search box above the bucket list, select **Bucket Name**, or **Region** from the level-1 drop-down list, and then the option you need from the corresponding level-2 drop-down list. Alternatively, after selecting an option from the level-1 drop-down list, you can enter a keyword in the search box and then select what you want from the level-2 drop-down list.

The found buckets are displayed in the bucket list.

For example, if you want to search for bucket **test**, click the search box, select **Bucket Name** and then **test**. Alternatively, after selecting **Bucket Name**, enter **test** in the search box, and all buckets whose names contain **test** are displayed in the level-2 drop-down list. Then, select **test** and click **OK**.

NOTE

- You can search for buckets based on combinations of different filter criteria.
  - If the filter criteria are of different types, they are in intersection logic.
  - If the filter criteria are of the same type, they are in union logic.
- After a keyword is entered in the search box, all buckets whose name, or region contains the specified keyword are displayed in the drop-down list. Click the option you want. Then, all the buckets meeting the search criteria are displayed in the bucket list.

Step 2 Enter a keyword in the search box and click 🔍 or press **Enter**.

All buckets whose name, data redundancy policy, or region contains the searched keyword will be displayed in the bucket list.

For example, if you enter **test** in the search box and click 🔍 or press **Enter**, all buckets whose name, data redundancy policy, or region contains keyword **test** are displayed in the bucket list.

----End

Related Operations

In the bucket list, click ↕ next to the bucket name, region, data redundancy policy, used capacity, number of objects, or creation time to sort buckets.

3.4.4 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 **OK** to confirm the deletion.

----End

3.5 Managing Objects

3.5.1 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 to go to the **Objects** page.

Step 2 Click **Create Folder**, or click a folder in the object list to open it and click **Create Folder**.

Step 3 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 4 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.

3.5.2 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 file size exceeds 5 GB, use OBS tools (like OBS Browser+) or the multipart upload of OBS APIs for upload. To download OBS Browser+, you can click the OBS Browser+ package link on the homepage of OBS Console.

  NOTICE

  Before downloading and using OBS Browser+, you must complete required configurations by following **Configuring the Local hosts File**.

- 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

- You have completed required configurations by following **Configuring the Local hosts File**.
- 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 to go to the **Objects** page.

Step 2 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.

Step 3 In the **Upload Object** area, drag and drop the files or folders you want to upload.

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

Step 4 (Optional) To configure metadata, click **Next: (Optional) Configure Advanced Settings**.

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

Step 5 Click **Upload**.

----End

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.

3.5.3 Downloading an Object

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

Prerequisites

You have completed required configurations by following **Configuring the Local hosts File**.

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

Step 2 Select the file you want to download. Then, click **Download** or **More** > **Download As** on the right.

You can also select multiple files and choose **Download** above the file list.

NOTE

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

----End

3.5.4 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 to go to the **Objects** page.

Step 2 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 3 Click 🔍. The search results are displayed in the object list.

----End

Related Operations

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

3.5.5 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**.

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

Step 2 Click the object to be shared. The object information is displayed on the top part of the page. You can find the link for accessing the object 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.

----End

3.5.6 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 to go to the **Objects** page.

Step 2 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 3 Click **OK** 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, see the following steps.

Step 4 If versioning is enabled for the bucket, delete the files or folders again from the **Deleted Objects** list to permanently delete them.

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.

3.5.7 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 to go to the **Objects** page.

Step 2 Click **Deleted Objects**.

Step 3 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 to go to the **Objects** page.

Step 2 Click **Deleted Objects**.

Step 3 Click the deleted file that you want to recover. The file information is displayed.

Step 4 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

3.5.8 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.

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

Step 2 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 3 Click **OK** to confirm the deletion.

----End

3.6 Object Metadata

3.6.1 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 3-7 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.
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.

Name	Description
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)
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**.

3.6.2 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 3-8 Common Content-Type values

File Name Extension	Content-Type	File Name Extension	Content-Type
* (binary stream, which does not know the type of the file to be downloaded)	application/octet-stream	.tif	image/tiff
.001	application/x-001	.301	application/x-301
.323	text/h323	.906	application/x-906
.907	drawing/907	.a11	application/x-a11
.acp	audio/x-mei-aac	.ai	application/postscript
.aif	audio/aiff	.aifc	audio/aiff
.aiff	audio/aiff	.anv	application/x-anv
.asa	text/asa	.asf	video/x-ms-asf
.asp	text/asp	.asx	video/x-ms-asf
.au	audio/basic	.avi	video/avi
.awf	application/vnd.adobe.workflow	.biz	text/xml
.bmp	application/x-bmp	.bot	application/x-bot
.c4t	application/x-c4t	.c90	application/x-c90
.cal	application/x-cals	.cat	application/vnd.ms-pki.seccat
.cdf	application/x-netcdf	.cdr	application/x-cdr

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

File Name Extension	Content-Type	File Name Extension	Content-Type
.frm	application/x-frm	.g4	application/x-g4
.gbr	application/x-gbr	.	application/x-
.gif	image/gif	.gl2	application/x-gl2
.gp4	application/x-gp4	.hgl	application/x-hgl
.hmr	application/x-hmr	.hpg	application/x-hpgl
.hpl	application/x-hpl	.hqx	application/mac-binhex40
.hrf	application/x-hrf	.hta	application/hta
.htc	text/x-component	.htm	text/html
.html	text/html	.htt	text/webviewhtml
.htx	text/html	.icb	application/x-icb
.ico	image/x-icon	.ico	application/x-ico
.iff	application/x-iff	.ig4	application/x-g4
.igs	application/x-igs	.iii	application/x-iphone
.img	application/x-img	.ins	application/x-internet-signup
.isp	application/x-internet-signup	.IVF	video/x-ivf
.java	java/*	.jfif	image/jpeg
.jpe	image/jpeg	.jpe	application/x-jpe
.jpeg	image/jpeg	.jpg	image/jpeg
.jpg	application/x-jpg	.js	application/x-javascript
.jsp	text/html	.la1	audio/x-liquid-file
.lar	application/x-laplayer-reg	.latex	application/x-latex
.lavs	audio/x-liquid-secure	.lbm	application/x-lbm
.lmsff	audio/x-la-lms	.ls	application/x-javascript
.ltr	application/x-ltr	.m1v	video/x-mpeg
.m2v	video/x-mpeg	.m3u	audio/mpegurl

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

File Name Extension	Content-Type	File Name Extension	Content-Type
.p12	application/x-pkcs12	.p7b	application/x-pkcs7-certificates
.p7c	application/pkcs7-mime	.p7m	application/pkcs7-mime
.p7r	application/x-pkcs7-certreqresp	.p7s	application/pkcs7-signature
.pc5	application/x-pc5	.pci	application/x-pci
.pcl	application/x-pcl	.pcx	application/x-pcx
.pdf	application/pdf	.pdf	application/pdf
.pdx	application/vnd.adobe.pdx	.pfx	application/x-pkcs12
.pgl	application/x-pgl	.pic	application/x-pic
.pko	application/vnd.ms-pki.pko	.pl	application/x-perl
.plg	text/html	.pls	audio/scpls
.plt	application/x-plt	.png	image/png
.png	application/x-png	.pot	application/vnd.ms-powerpoint
.ppa	application/vnd.ms-powerpoint	.ppm	application/x-ppm
.pps	application/vnd.ms-powerpoint	.ppt	application/vnd.ms-powerpoint
.ppt	application/x-ppt	.pr	application/x-pr
.prf	application/pics-rules	.prn	application/x-prn
.prt	application/x-prt	.ps	application/x-ps
.ps	application/postscript	.ptn	application/x-ptn
.pwz	application/vnd.ms-powerpoint	.r3t	text/vnd.rn-realtext3d
.ra	audio/vnd.rn-realaudio	.ram	audio/x-pn-realaudio

File Name Extension	Content-Type	File Name Extension	Content-Type
.ras	application/x-ras	.rat	application/rat-file
.rdf	text/xml	.rec	application/vnd.rn-recording
.red	application/x-red	.rgb	application/x-rgb
.rjs	application/vnd.rn-realsystem-rjs	.rjt	application/vnd.rn-realsystem-rjt
.rlc	application/x-rlc	.rle	application/x-rle
.rm	application/vnd.rn-realmedia	.rmf	application/vnd.adobe.rmf
.rmi	audio/mid	.rmj	application/vnd.rn-realsystem-rmj
.rmm	audio/x-pn-realaudio	.rmp	application/vnd.rn-rn_music_package
.rms	application/vnd.rn-realmedia-secure	.rmvb	application/vnd.rn-realmedia-vbr
.rmx	application/vnd.rn-realsystem-rmx	.rnx	application/vnd.rn-realplayer
.rp	image/vnd.rn-realpix	.rpm	audio/x-pn-realaudio-plugin
.rsml	application/vnd.rn-rsml	.rt	text/vnd.rn-realtext
.rtf	application/msword	.rtf	application/x-rtf
.rv	video/vnd.rn-realvideo	.sam	application/x-sam
.sat	application/x-sat	.sdp	application/sdp
.sdw	application/x-sdw	.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

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

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

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

3.6.3 Configuring Object Metadata

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

Step 2 Click the object to be operated, and then click the **Metadata** tab.

Step 3 Click **Add** and specify the metadata information.

Step 4 Click **OK**.

----End

3.7 Bucket Inventories

3.7.1 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, ETag, multipart upload, and replication status.

Constraints

- A bucket can have a maximum of 10 inventory rules.
- The source bucket (for which a bucket inventory rule is configured) and the target bucket (where the generated inventory files are stored) must belong to the same account.

- The source bucket and the target bucket 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 **ab**. Or, you can delete the existing inventory rule and create a new one that filters objects according to your needs.

3.7.2 Configuring a Bucket Inventory

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

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

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

Step 4 Configure required parameters.

Table 3-9 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.

Parameter	Description
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 3-10 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**, **ETag**, **Multipart upload**, 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 **OK**.

----End

Related Operations

Export bucket inventories on the bucket inventory page.

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

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

Step 3 Click ↗ in the upper right corner.

Step 4 View the bucket inventory Excel file automatically downloaded by the browser. The file contains the inventory name, inventory status, filter criteria, inventory storage bucket, inventory file prefix, generation frequency, and last generation time.

----End

3.8 Permissions Control

3.8.1 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.

  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.

3.8.2 Permission Control Mechanisms

3.8.2.1 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

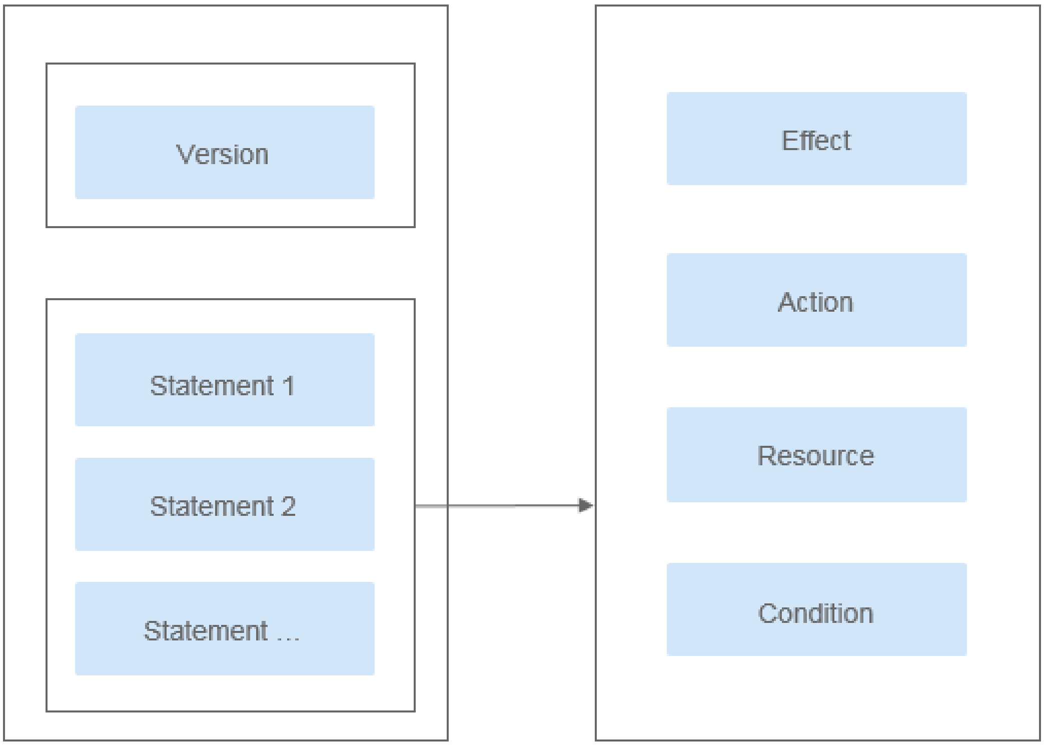
IAM policies are used to authorize IAM users under an account.

- 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 3-3 Policy structure



```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "obs:bucket:ListAllMybuckets",
                "obs:bucket:HeadBucket",
                "obs:bucket:ListBucket",
                "obs:bucket:GetBucketLocation"
            ]
        }
    ]
}
```

Table 3-11 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.

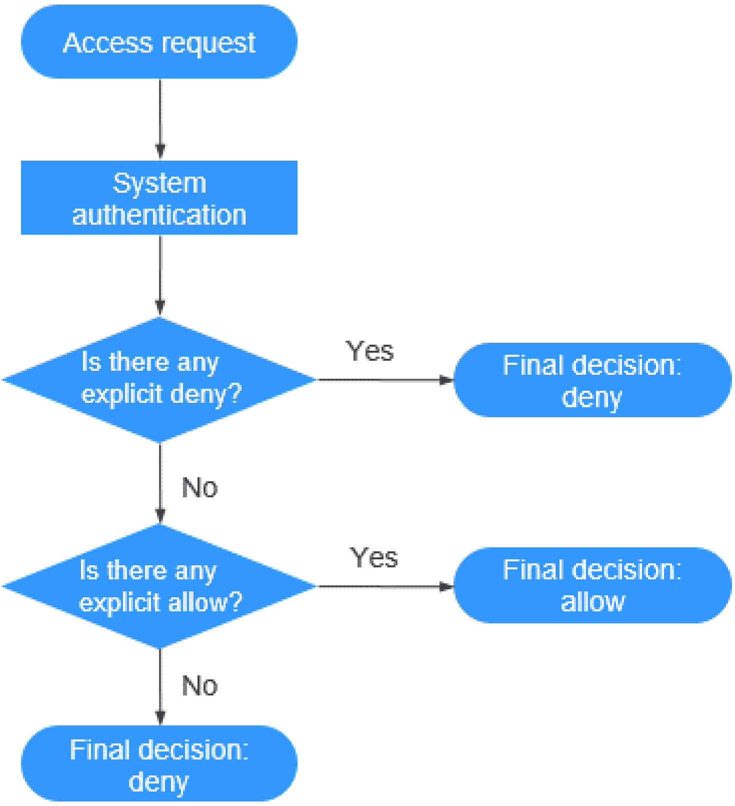
Parameter	Description
Statement	Permissions defined by a policy, including **Effect**, **Action**, **Resource**, and **Condition**. **Condition** is optional. ● **Effect** The valid values for **Effect** are **Allow** and **Deny**. System policies contain only **Allow** statements. For custom policies 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-Related Actions" and "Object-Related Actions" in the "IAM Permissions Policies and Supported Actions" section of the *OBS 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 3-4 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.

3.8.2.2 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.

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.

Bucket Policy Templates

OBS Console provides bucket policy templates for eight typical scenarios. You can use these templates to quickly create bucket policies.

Some templates may require a configuration of principals or resources. You can also modify the existing template settings, including principals, resources, actions, and conditions.

Table 3-12 Bucket policy templates

Principal	Resource	Template	Actions Allowed	Advanced Settings
All accounts	Entire bucket (including the objects in it)	Public Read	**Allows anonymous users to perform the following actions on a bucket and the objects in it:** HeadBucket (to check whether the bucket exists and obtain the bucket metadata) GetBucketLocation (to get the bucket location) GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version)	Excluding the specified actions is not allowed.

Principal	Resource	Template	Actions Allowed	Advanced Settings
		Public Read/ Write	**Allows anonymous users to perform the following actions on a bucket and the objects in it:** ListBucket (to list objects in the bucket and obtain the bucket metadata) ListBucketVersions (to list object versions in the bucket) HeadBucket (to check whether the bucket exists and obtain the bucket metadata) GetBucketLocation (to get the bucket location) PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts) GetObject (to obtain object content and metadata) ModifyObjectMetaData (to modify object metadata) ListBucketMultipartUploads (to list multipart uploads) ListMultipartUploadParts (to list uploaded parts) AbortMultipartUpload (to abort multipart uploads) GetObjectVersion (to obtain the content and metadata of a specified object version) PutObjectAcl (to configure the object ACL) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL)	Excluding the specified actions is not allowed.

Principal	Resource	Template	Actions Allowed	Advanced Settings
Current account/Other accounts/Delegated accounts	Entire bucket (including the objects in it)	Bucket Read-Only	**Allows specified accounts to perform the following actions on a bucket and the objects in it:** Get* (all GET actions) List* (all LIST actions) HeadBucket (to check whether the bucket exists and obtain the bucket metadata)	Excluding the specified actions is not allowed.
		Bucket Read/Write	**Allows specified accounts to perform all actions excluding the following ones on a bucket and the objects in it:** DeleteBucket (to delete the bucket) PutBucketPolicy (to configure a bucket policy) PutBucketAcl (to configure the bucket ACL)	The specified actions are excluded.

Principal	Resource	Template	Actions Allowed	Advanced Settings
All accounts/ Current account/ Other accounts/ Delegated accounts	Current bucket + Specified objects	Directory Read-Only	**Allows anonymous users or specified accounts to perform the following actions on the current bucket and the specified resources in it:** GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL) ListBucket (to list objects in the bucket and obtain the bucket metadata) ListBucketVersions (to list object versions in the bucket) HeadBucket (to check whether the bucket exists and obtain the bucket metadata) GetBucketLocation (to get the bucket location) **NOTE** If you apply the policy to **All accounts**, **ListBucket** and **ListBucketVersions** are not included in the template.	Excluding the specified actions is not allowed.

Principal	Resource	Template	Actions Allowed	Advanced Settings
		Directory Read/ Write	**Allows anonymous users or specified accounts to perform the following actions on the current bucket and the specified resources in it:** PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts) GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) ModifyObjectMetaData (to modify object metadata) ListBucketMultipartUploads (to list multipart uploads) ListMultipartUploadParts (to list uploaded parts) AbortMultipartUpload (to abort multipart uploads) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL) PutObjectAcl (to configure the object ACL) ListBucket (to list objects in the bucket and obtain the bucket metadata) ListBucketVersions (to list object versions in the bucket) HeadBucket (to check whether the bucket exists and obtain the bucket metadata) GetBucketLocation (to get the bucket location)	Excluding the specified actions is not allowed.

Principal	Resource	Template	Actions Allowed	Advanced Settings
All accounts/ Current account/ Other accounts/ Delegated accounts	Specified objects	Object Read-Only	**Allows anonymous users or specified accounts to perform the following actions on specified resources in the bucket:** GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL)	Excluding the specified actions is not allowed.
		Object Read/ Write	**Allows anonymous users or specified accounts to perform the following actions on specified resources in the bucket:** PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts) GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) ModifyObjectMetaData (to modify object metadata) ListMultipartUploadParts (to list uploaded parts) AbortMultipartUpload (to abort multipart uploads) GetObjectVersionAcl (to obtain the ACL of an object version) GetObjectAcl (to obtain the object ACL) PutObjectAcl (to configure the object ACL)	Excluding the specified actions is not allowed.

Custom Bucket Policies

You can also customize bucket policies based on your needs. A custom bucket policy consists of five basic elements: effect, principals, resources, actions, and conditions. For details, see **Bucket Policy Parameters**.

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.

Object Policy Templates

OBS Console provides object policy templates for two typical scenarios. You can use these templates to quickly create object policies.

Some templates may require a configuration of principals. You can also modify the existing template settings, including principals, actions, and conditions. The resource in an object policy is the object that the policy is applied to, which is automatically specified by the system and does not need to be modified.

Table 3-13 Object policy templates

Principal	Resource	Template	Actions Allowed	Advanced Settings
All accounts/Current account/Other accounts/Delegated accounts	Specified objects	Object Read-Only	**Allows anonymous users or specified accounts to perform the following actions on specified resources in the bucket:** GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL)	Excluding the specified actions is not allowed.

Principal	Resource	Template	Actions Allowed	Advanced Settings
		Object Read/ Write	**Allows anonymous users or specified accounts to perform the following actions on specified resources in the bucket:** PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts) GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) ModifyObjectMetaData (to modify object metadata) ListMultipartUploadParts (to list uploaded parts) AbortMultipartUpload (to abort multipart uploads) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL) PutObjectAcl (to configure the object ACL)	Excluding the specified actions is not allowed.

Custom Object Policies

You can also customize object policies based on your needs. A custom object policy consists of five basic elements: effect, principals, resources, actions, and conditions, similar to a bucket policy. For details, see **Bucket Policy Parameters**.

3.8.2.3 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 3-14**.

Table 3-14 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**.
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 3-15 lists the permissions you can grant using a bucket ACL.

Table 3-15 Access permissions controlled by a bucket ACL

Permission	Option	Description
Access to Bucket	READ	Used to obtain the list of objects or object versions in a bucket and to obtain the multipart uploads, metadata, and versioning settings of a bucket.
	WRITE	Used to upload, overwrite, and delete any object in a bucket.
Access to Object	Object READ	Used to obtain the object content and metadata.
Access to ACL	READ_ACP	Used to list the ACLs of a bucket and of objects in the bucket. The bucket owner has this permission permanently by default.
	WRITE_ACP	Used to update the ACL of a bucket. The bucket owner has this permission permanently by default.

Table 3-16 lists the permissions you can grant using an object ACL.

Table 3-16 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 3-17** lists the predefined permissions.

Table 3-17 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.
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 or the bucket can be added as an external bucket.

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.

3.8.2.4 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 3-18** shows the mapping between bucket ACL access permissions and bucket policy actions.

Table 3-18 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

ACL Permission	Option	Mapped Action in a Custom Bucket Policy
Access to object	Object read	● GetObject
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 3-19 describes the mapping relationship between object ACL access permissions and bucket policy actions.

Table 3-19 Mapping relationship between object ACLs and bucket policies

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

3.8.2.5 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 3-5 Authorization process

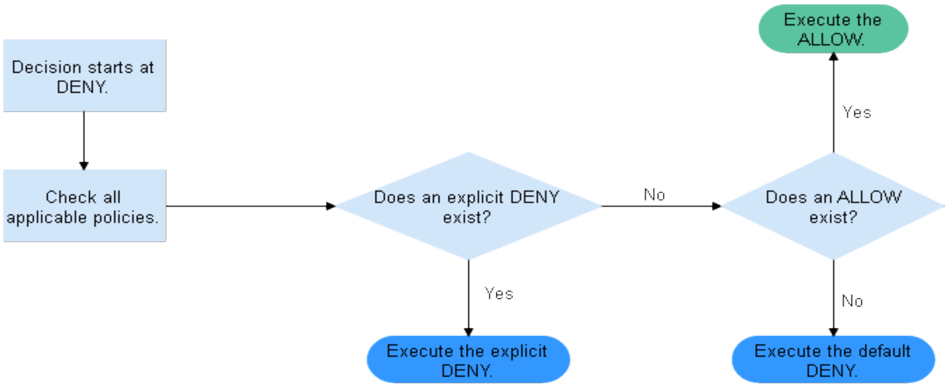
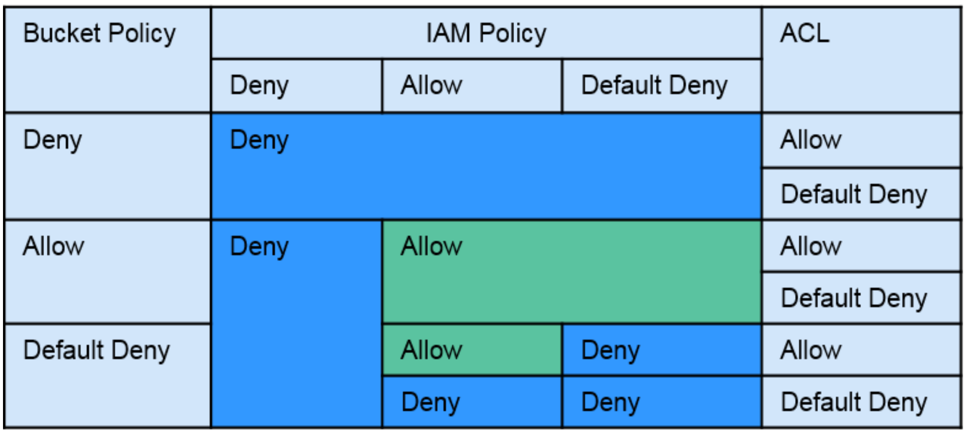

Decision starts at DENY.
Check all applicable policies.
Does an explicit DENY exist?
No
Does an ALLOW exist?
Yes
Execute the ALLOW.
Yes
Execute the explicit DENY.
No
Execute the default DENY.

Figure 3-6 is a matrix of the IAM policies, bucket policies, and ACLs (allow and deny effects).

Figure 3-6 Matrix of the IAM policies, bucket policies, and ACLs (allow and deny effects)



Bucket Policy	IAM Policy			ACL
	Deny	Allow	Default Deny	
Deny	Deny			Allow
				Default Deny
Allow	Deny	Allow		Allow
				Default Deny
Default Deny		Allow	Deny	Allow
		Deny	Deny	Default Deny

3.8.3 Bucket Policy Parameters

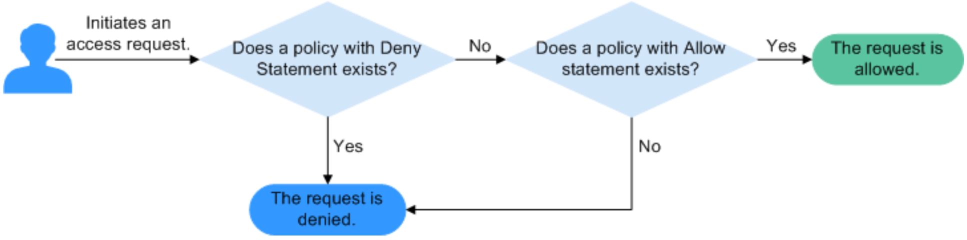
3.8.3.1 Effect

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 3-7 Determining a bucket policy when the allow and deny statements conflict


Initiates an access request.
Does a policy with Deny Statement exists?
No
Does a policy with Allow statement exists?
Yes
The request is allowed.
Yes
No
The request is denied.

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.

3.8.3.2 Principals

The principals indicate the users bucket policies apply to. These users can be accounts and IAM users. The **Exclude** setting can be used to determine whether the bucket policy applies to the specified principals.

Specified principals: By selecting this option (optional), the bucket policy applies to users except the specified ones.

NOTE

- **Exclude** not selected: The bucket policy applies to the specified users.
- **Exclude** selected: The bucket policy applies to users except the specified ones.

3.8.3.3 Resources

You can apply a bucket policy to the following resources: an entire bucket (including the objects in it), the current bucket, and specified objects in a bucket.

The **Exclude** setting can be used to determine whether the bucket policy applies to the specified resources.

Selecting **Specified resources** for **Exclude** will let the bucket policy apply to the resources except the specified ones.

NOTE

If you do not select **Specified resources** for **Exclude**, the bucket policy applies to the specified resources.

Applying a Bucket Policy to the Entire Bucket (Including the Objects in It)

If you apply the bucket policy to the entire bucket (including the objects in it), actions related to the bucket and objects must be configured in the policy.

Applying a Bucket Policy to a Bucket

To specify the current bucket as the resource, select **Current 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. Specifically, select **Specified objects** for **Resources**. 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

3.8.3.4 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.

The **Exclude** setting can be used to determine whether the bucket policy applies to the specified actions.

Selecting **Specified actions** for **Exclude** will let the bucket policy apply to the actions except the specified ones.

NOTE

- If you do not select **Specified actions** for **Exclude**, the bucket policy applies to the specified actions.
- By default, **Specified actions** is selected for **Exclude** in the bucket read/write template only. The action exclusion setting in bucket policy templates cannot be modified.

Actions Related to Buckets

Table 3-20 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 versioned objects in a bucket.
	ListBucketMultipartUploads	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.
	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 the static website hosting for the bucket.

Type	Value	Description
	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 3-21 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, initialization of uploaded parts, and merging of parts.
	GetObjectAcl	Obtains the object ACL information.
	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.

Type	Value	Description
	ListMultipartUploadParts	Lists uploaded parts.
	AbortMultipartUpload	Aborts a multipart upload.

3.8.3.5 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 3-22 describes the predefined condition operators provided by OBS.

Table 3-22 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
	StringNotEqualsIgnoreCase	Strict negated matching, ignoring case. Short version: strneqi

Type	Key	Description
	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
	NumericGreaterThanEquals	"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
	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

Type	Key	Description
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 3-23 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 3-24 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 3-25 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**.
	copysource	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.

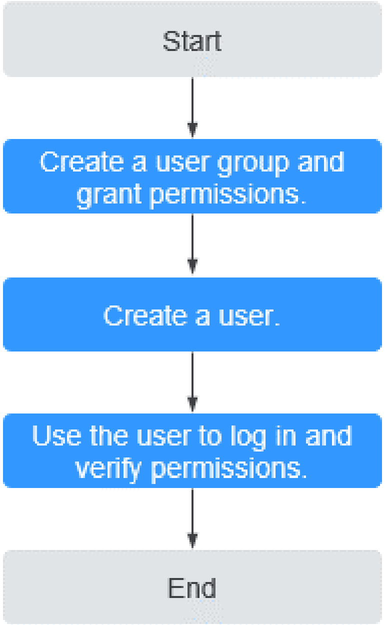
Action	Optional Key	Description
	acl	Type: String. Configures the ACL of the object with the specified version ID. When an object is uploaded, the canned ACLs that can be included in the object ACL contain **private**, **public-read**, **public-read-write**, **authenticated-read**, **bucket-owner-read**,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

3.8.4 Configuring IAM Policies

3.8.4.1 Creating an IAM User and Granting OBS Permissions

Process

Figure 3-8 Process of granting an IAM user the OBS permissions


Start
Create a user group and grant permissions.
Create a user.
Use the user to log in and verify permissions.
End

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 **Authorize** in the **Operation** column of the row.
5. Under **Select Policy/Role**, filter policies based on policy types in the upper right corner, required policy names, and click **Next**.
6. Under **Select Scope**, select **Global services** and click **OK**.

> **NOTE**
>
> In the policy content area, you can view the authorization details.
>
> 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 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

3.8.4.2 Configuring Fine-Grained Policies

Custom policies can be created to supplement the system-defined policies of OBS.

Procedure

Step 1 Log in to the IAM management console.

Step 2 In the navigation pane on the left, click **Policies**.

Step 3 On the page that is displayed, click **Create Custom Policy**.

Step 4 On the **Create Custom Policy** page, set the following parameters:

Table 3-26 Custom policy parameters

Parameter	Description
Policy Name	Only letters, digits, and the following special characters are allowed: -_.,
Scope	Select the scope based on the service properties. OBS is a global-level service.

Parameter	Description
Description	(Optional) Brief description about the policy
Policy Information	After you select a template, you can customize the content. For details, see **Policy Structure and Syntax**.

NOTE

A custom policy can contain multiple authorization items. In addition to the authorization items related to OBS, it can also contain authorization items related other services. But the other services must be at the same project level as OBS.

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

Step 6 Grant fine-grained permissions to a user group and add a user to the user group, so that the user can have the granted permissions. For details, see **Creating an IAM User and Granting OBS Permissions**.

----End

3.8.5 Configuring a Bucket Policy

3.8.5.1 Creating a Bucket Policy with a Template

OBS Console provides bucket policy templates for eight typical scenarios. You can use these templates to quickly configure bucket policies.

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

Step 2 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 3 Click **Create**.

Step 4 Choose a policy template. For details about the parameters, see **Bucket Policies and Object Policies**.

Table 3-27 Bucket policy templates

Principal	Resource	Template	Actions Allowed	Advanced Settings
All accounts	Entire bucket (including the objects in it)	Public Read	**Allows anonymous users to perform the following actions on a bucket and the objects in it:** GetBucketLocation (to get the bucket location) GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version)	Excluding the specified actions is not allowed.

Principal	Resource	Template	Actions Allowed	Advanced Settings
		Public Read/ Write	**Allows anonymous users to perform the following actions on a bucket and the objects in it:** ListBucket (to list objects in the bucket and obtain the bucket metadata) ListBucketVersions (to list object versions in the bucket) GetBucketLocation (to get the bucket location) PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts) GetObject (to obtain object content and metadata) ModifyObjectMetaData (to modify object metadata) ListBucketMultipartUploads (to list multipart uploads) ListMultipartUploadParts (to list uploaded parts) AbortMultipartUpload (to abort multipart uploads) GetObjectVersion (to obtain the content and metadata of a specified object version) PutObjectAcl (to configure the object ACL) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL)	Excluding the specified actions is not allowed.

Principal	Resource	Template	Actions Allowed	Advanced Settings
Current account/ Other accounts/ Delegated accounts	Entire bucket (including the objects in it)	Bucket Read-Only	**Allows specified accounts to perform the following actions on a bucket and the objects in it:** Get* (all GET actions) List* (all LIST actions)	Excluding the specified actions is not allowed.
		Bucket Read/ Write	**Allows specified accounts to perform all actions excluding the following ones on a bucket and the objects in it:** DeleteBucket (to delete the bucket) PutBucketPolicy (to configure a bucket policy) PutBucketAcl (to configure the bucket ACL)	The specified actions are excluded.
All accounts/ Current account/ Other accounts/ Delegated accounts	Current bucket + Specified objects	Directory Read-Only	**Allows all accounts or specified accounts to perform the following actions on the current bucket and the specified resources in it:** GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL) ListBucket (to list objects in the bucket and obtain the bucket metadata) ListBucketVersions (to list object versions in the bucket) GetBucketLocation (to get the bucket location) **NOTE** If you apply the policy to **All accounts**, **ListBucket** and **ListBucketVersions** are not included in the template.	Excluding the specified actions is not allowed.

Principal	Resource	Template	Actions Allowed	Advanced Settings
		Directory Read/ Write	**Allows all accounts or specified accounts to perform the following actions on the current bucket and the specified resources in it:** PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts) GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) ModifyObjectMetaData (to modify object metadata) ListBucketMultipartUploads (to list multipart uploads) ListMultipartUploadParts (to list uploaded parts) AbortMultipartUpload (to abort multipart uploads) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL) PutObjectAcl (to configure the object ACL) ListBucket (to list objects in the bucket and obtain the bucket metadata) ListBucketVersions (to list object versions in the bucket) GetBucketLocation (to get the bucket location)	Excluding the specified actions is not allowed.

Principal	Resource	Template	Actions Allowed	Advanced Settings
All accounts/ Current account/ Other accounts/ Delegated accounts	Specified objects	Object Read-Only	**Allows all accounts or specified accounts to perform the following actions on specified resources in the bucket:** GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL)	Excluding the specified actions is not allowed.
		Object Read/ Write	**Allows all accounts or specified accounts to perform the following actions on specified resources in the bucket:** PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts) GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) ModifyObjectMetaData (to modify object metadata) ListMultipartUploadParts (to list uploaded parts) AbortMultipartUpload (to abort multipart uploads) GetObjectVersionAcl (to obtain the ACL of an object version) GetObjectAcl (to obtain the object ACL) PutObjectAcl (to configure the object ACL)	Excluding the specified actions is not allowed.

Step 5 Complete the bucket policy configuration.

Some bucket policy templates require a configuration of principals or resources. You can also change the existing settings of a template, including the policy name, principals, resources, actions, and conditions. For details, see **Bucket Policy Parameters**.

Step 6 Click **Create** in the lower right corner.

----End

3.8.5.2 Creating a Custom Bucket Policy (Visual Editor)

You can customize bucket policies based on your needs. A custom bucket policy consists of five basic elements: effect, principals, resources, actions, and conditions.

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

Step 2 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 3 Click **Create**.

Step 4 Configure a bucket policy.

Table 3-28 Parameters for configuring a custom bucket policy

Parameter		Description
Method		Visual editor or JSON. The visual editor is used here. For details about configurations in the JSON view, see **Creating a Custom Bucket Policy (JSON View)**.
Policy Name		Enter a bucket policy name.
Policy content	Effect	● **Allow**: The policy allows the matched requests. ● **Deny**: The policy denies the matched requests.

Parameter		Description
	Principals	- **All accounts**: The bucket policy applies to anonymous users. - **Current account**: Specify one or more IAM users under the current account. - **Other accounts**: Specify one or more accounts. **NOTE** The account ID and IAM user ID can be obtained from the **My Credentials** page. Accounts should be configured in the *Domain ID*/*IAM user ID* format, with each one on a separate line. *Account ID*/* indicates that permission is granted to all IAM users under the account. - **Delegated accounts**: Delegated accounts can be added only after **Other accounts** is selected. **NOTE** Delegated accounts should be configured in the *ID of a delegating account*/*Agency name* format. Multiple delegated accounts are allowed, with each one on a separate line.
	Resources	- **Entire bucket (including the objects in it)**: The policy applies to the bucket and the objects in it. You can configure bucket and object actions in this policy. - **Current bucket**: The policy applies to the current bucket. You can configure bucket actions in this policy. - **Specified objects**: The policy applies to specified objects in the bucket. You can configure object actions in this policy. **NOTE** 1. Multiple resource paths can be specified. 2. A resource path should be configured in the *Folder name*/*Object name* format, for example, **testdir/a.txt**. To specify the **testdir** folder and all objects in it, enter **testdir/***. 3. You can specify a specific object, an object set, or a directory. * indicates all objects in the bucket. To specify a specific object, enter the object name. To specify a set of objects, enter *Object name prefix**, **Object name suffix*, or *. For example, **testdir/*** indicates objects in the **testdir** folder, and **testprefix*** indicates objects whose prefix is **testprefix**.

Parameter		Description
	Actions	● **Actions**: Choose **Customize**. ● **Select Actions**: See **Actions**. NOTE 1. If you select **Entire bucket (including the objects in it)** for **Resources**, common actions, bucket actions, and object actions will be available for you to choose from. 2. If you select **Current bucket** for **Resources**, common actions and bucket actions will be available for you to choose from. 3. If you select **Specified objects** for **Resources**, common actions and object actions will be available for you to choose from. 4. If you select both **Current bucket** and **Specified objects** for **Resources**, common actions, bucket actions, and object actions will be available for you to choose from.
	Conditions (Optional)	● **Key**: See **Conditions**. ● **Conditional Operator**: See **Conditions**. ● **Value**: The entered value is associated with the key.
	Advanced Settings > Exclude (Optional)	● **Specified principals**: By selecting this option, the bucket policy applies to users except the specified ones. NOTE If you do not select this option, the bucket policy applies to the specified users. ● **Specified resources**: By selecting this option, the bucket policy applies to resources except the specified ones. NOTE If you do not select this option, the bucket policy applies to the specified resources. ● **Specified actions**: By selecting this option, the bucket policy applies to actions except the specified ones. NOTE 1. If you do not select this option, the bucket policy applies to the specified actions. 2. By default, **Specified actions** is selected for **Exclude** in the bucket read/write template only. The action exclusion setting in bucket policy templates cannot be modified.

Step 5 Click **Create** in the lower right corner.

----End

3.8.5.3 Creating a Custom Bucket Policy (JSON View)

If you are familiar with the JSON syntax and OBS bucket policies, you can code a bucket policy in the JSON view. There is no limit on the number of bucket policies (statements) for a bucket, but the JSON descriptions of all bucket policies in a bucket cannot exceed 20 KB in total.

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

Step 2 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 3 Click **Create** and click the **JSON** tab.

Step 4 Edit the bucket policy. Below gives a bucket policy example in JSON:

```
{
    "Statement":[
        {
            "Action":[
                "CreateBucket",
                "DeleteBucket"
            ],
            "Effect":"Allow",
            "Principal":{
                "ID":[
                    "domain/account ID",
                    "domain/account ID:user/User ID"
                ]
            },
            "Condition":{
                "NumericNotEquals":{
                    "Referer":"sdf"
                },
                "StringNotLike":{
                    "Delimiter":"ouio"
                }
            },
            "Resource":"000-02/key01"
        }
    ]
}
```

Table 3-29 Parameters for creating a bucket policy in JSON

Parameter	Description
Action	Actions the bucket policy applies to. For details, see **Actions**.
Effect	Effect of the bucket policy. For details, see **Effect**.

Parameter	Description
Principal	Users the bucket policy is applied to. You can obtain the user ID on the **My Credentials** page by logging in to the console as the user to be authorized. Principals should be configured as follows: ● **domain/***Account ID* (indicating that the principal is an account) ● **domain/***Account ID***:user/***User ID* (indicating that the principal is a user under an account)
Condition	Conditions under which the bucket policy takes effect. For details, see **Conditions**.
Resource	Resources the bucket policy is applied to. For details, see **Resources**.

Step 5 Click **Create**.

----End

3.8.6 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 to go to the **Objects** page.

Step 2 In the row containing the object for which you want to configure a policy, choose **More** > **Configure Object Policy** in the **Operation** column. The **Configure Object Policy** page is displayed.

You can customize a policy or use a preset template to configure one as needed.

- **Using a preset template**: The system presets object policy templates for two typical scenarios. You can use the templates to quickly configure object policies.
- **Customizing a policy**: You can also customize an object policy based on your needs. A custom object policy consists of five basic elements: effect, principals, resources, actions, and conditions, similar to a bucket policy. For details, see **Bucket Policy Parameters**. The resource is the selected object and is automatically configured by the system. For details about how to customize an object policy, see **Creating a Custom Bucket Policy (Visual Editor)**. Different from customizing a bucket policy, to customize an object policy, you:

  a. Do not need to specify the resource.

b. Can configure only object-related actions.

----End

3.8.7 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 to go to the **Objects** page.

Step 2 In the navigation pane, choose **Permissions** > **Bucket ACLs**.

Step 3 On the **Bucket ACLs** page, choose a permission from **Private**, **Public Read**, and **Public Read/Write** to grant bucket ACL permission for anonymous users.

NOTE

1. After you change **Public Read** or **Public Read/Write** to **Private**, only the bucket owner or object owner has the access.
2. After you change **Private** to **Public Read**, anyone can read objects in the bucket. No identity authentication is required.
3. After you change **Private** to **Public Read/Write**, anyone can read, write, and delete objects in the bucket. No identity authentication is required.

Step 4 In the **Operation** column, click **Edit** to grant the owner, anonymous user, or log delivery user required ACL permissions for the bucket.

Step 5 In the middle of the page, click **Export** to get the bucket ACL configuration. The file includes the user type, account, bucket access, and ACL access.

Step 6 In the middle of the page, click **Add** to apply specific ACL permissions to an account.

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

Click **OK**.

NOTE

To select **Object read** for **Object Permission**, you must select **Read** for **Access to Bucket**.

----End

3.8.8 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 to go to the **Objects** page.

Step 2 Click a desired object.

Step 3 On the **Object ACL** page, choose a permission from **Private** and **Public Read** to grant object ACL permission for anonymous users.

NOTE

1. After you change **Public Read** to **Private**, only the bucket owner or object owner has the access.
2. After you change **Private** to **Public Read**, anyone can read the object content and metadata. No identity authentication is required.

Step 4 Click **Edit** to grant the owner, anonymous user, or other accounts required permissions for the object.

Step 5 Click **Export** to get the object ACL configuration. The file includes the user type, account, object access, and ACL access.

Step 6 Click **Add** to apply specific ACL permissions to an account.

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

Click **OK**.

----End

3.8.9 Application Cases

3.8.9.1 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 to go to the **Objects** page.

Step 2 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 3 Click **Create**.

Step 4 Configure parameters listed in the table below to grant an IAM user the permissions to access the bucket (to list objects in the bucket) and to upload objects.

Table 3-30 Parameters for granting the object listing and upload permissions

Parameter		Description
Configuration method		Choose **Visual Editor**.
Policy Name		Enter a custom policy name.
Policy content	Effect	Select **Allow**.
	Principals	● Select **Current account**. ● Specify an IAM user under the current account.
	Resources	● Method 1: – Select **Entire bucket (including the objects in it)**. ● Method 2: – Select **Current bucket** and **Specified objects**. – Set the resource path to ***** (indicating all objects in the bucket).
	Actions	● Choose **Customize**. ● Select the following actions: – **ListBucket** (to list objects in the bucket and obtain the bucket metadata) – **PutObject** (to upload objects) NOTE In this example, only the upload action among object actions is selected. You can also select other object actions to grant corresponding permissions if needed. The asterisk (*) indicates all actions. To learn the supported actions and their meanings, see **Actions**.

Step 5 Click **Create** in the lower right corner.

----End

3.8.9.2 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 to go to the **Objects** page.

Step 2 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 3 Click **Create**.

Step 4 Configure parameters listed in the table below to grant other accounts the permissions to access the bucket (to list objects in the bucket) and to upload objects.

Table 3-31 Parameters for granting the object listing and upload permissions

Parameter		Description
Configuration method		Choose **Visual Editor**.
Policy Name		Enter a custom policy name.
Policy content	Effect	Select **Allow**.
	Principals	● Select **Other accounts**. **NOTE** 1. You can obtain the account ID and IAM user ID from the **My Credentials** page. 2. Accounts should be configured in the *Domain ID*/*IAM user ID* format, with each one on a separate line. 3. The following describes different authorization scenarios: **Granting permissions to all the other accounts and their IAM users**: Set the account ID and IAM user ID to *. **Granting permissions to an account**: Enter the desired account ID and IAM user ID. **Granting permissions to an account and its IAM users**: Enter the desired account ID, and set the IAM user ID to * (indicating all IAM users under the account). **Granting permissions to certain IAM users**: Enter the account ID and one or more IAM user IDs.

Parameter		Description
	Resources	● Method 1: – Select **Entire bucket (including the objects in it)**. ● Method 2: – Select **Current bucket** and **Specified objects**. – Set the resource path to ***** (indicating all objects in the bucket).
	Actions	● Choose **Customize**. ● Select actions: **ListBucket** (to list objects in the bucket and obtain the bucket metadata) and **PutObject** (to upload objects). **NOTE** In this example, only the upload action among object actions is selected. You can also select other object actions to grant corresponding permissions if needed. The asterisk (*) indicates all actions. To learn the supported actions and their meanings, see **Actions**.

Step 5 Click **Create** in the lower right corner.

----End

3.8.9.3 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 to go to the **Objects** page.

Step 2 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 3 Click **Create**.

Step 4 Configure parameters listed in the table below.

Table 3-32 Restricting access to a bucket for specific addresses

Parameter	Description
Configuration method	Choose **Visual Editor**.
Policy Name	Enter a custom policy name.

Parameter		Description
Policy content	Effect	Select **Deny**.
	Principals	● Select **All accounts**.
	Resources	● Method 1: – Select **Entire bucket (including the objects in it)**. ● Method 2: – Select **Current bucket** and **Specified objects**. – Set the resource path to * (indicating all objects in the bucket).
	Actions	● Choose **Customize**. ● Select * (indicating all actions).
	Conditions	● **Key**: Select **SourceIp**. ● **Condition Operator**: Select **IpAddress**. ● **Value**: Enter **114.115.1.0/24**.

Step 5 Click **Create** in the lower right corner.

----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.

3.8.9.4 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 to go to the **Objects** page.

Step 2 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 3 Click **Create**.

Step 4 Configure parameters listed in the table below.

Table 3-33 Limiting the time when objects in a bucket are accessible

Parameter		Description
Configuration method		Choose **Visual Editor**.
Policy Name		Enter a custom policy name.
Policy content	Effect	Select **Allow**.
	Principals	● Select **All accounts**.
	Resources	● Select **Specified objects**. ● Set the resource path to *****. **NOTE** 1. ***** indicates all objects in a bucket. 2. 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	● Choose **Customize**. ● Select ***** (indicating all object actions). **NOTE** Selecting ***** may cause resources to be deleted. To avoid this risk, select **Get*** that indicates all read permissions.
	Conditions	● Condition 1: – **Key**: Select **CurrentTime**. – **Condition Operator**: Select **DateGreaterThan**. – **Value**: Enter **2019-03-26T12:00:00Z** (UTC). ● Condition 2: – **Key**: Select **CurrentTime**. – **Condition Operator**: Select **DateLessThan**. – **Value**: Enter **2019-03-26T15:00:00Z** (UTC).

Step 5 Click **Create** in the lower right corner.

----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.

3.8.9.5 Granting Anonymous Users Permission to Access Objects

An enterprise stores a large volume of map 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 map data to the new bucket. The map data is stored as an object.

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

Step 4 Choose **Object ACL** > **Public Access** > **Public Read**. In the displayed dialog box, select **I understand the possible impacts of this configuration**, and click **Continue**.

Step 5 Click **Edit** in the **Operation** column of **Anonymous User**. In the displayed dialog box, grant the object read permission to anonymous users and click **OK**.

Step 6 Click **OK**.

----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

3.8.9.6 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 to go to the **Objects** page.

Step 2 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 3 Click **Create**.

Step 4 Configure parameters listed in the table below.

Table 3-34 Granting folder access permissions to anonymous users

Parameter		Description
Configuration method		Choose **Visual Editor**.
Policy Name		Enter a custom policy name.
Policy content	Effect	Allow
	Principals	● Select **All accounts**.
	Resources	● Select **Specified objects**. ● Enter an object name prefix for the resource path, for example, **folder-001/***, which indicates that the policy applies to all objects in folder **folder-001**.
	Actions	● Choose **Customize**. ● Select **GetObject** (to obtain object content and metadata).

Step 5 Click **Create** in the lower right corner.

----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. Anyone can access the object.

----End

3.9 Versioning

3.9.1 Versioning Overview

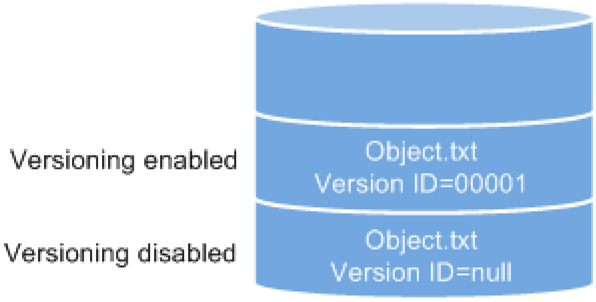
OBS can store multiple versions of an object. You can quickly search for and restore different versions or restore data in the event of accidental deletions or application faults.

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 3-9**.

Figure 3-9 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 3-10 Versioning (for new objects)

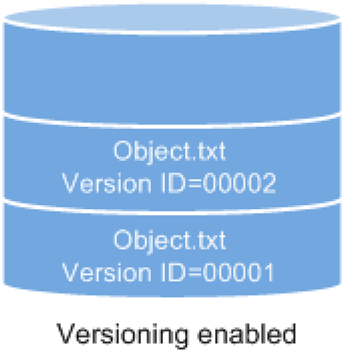


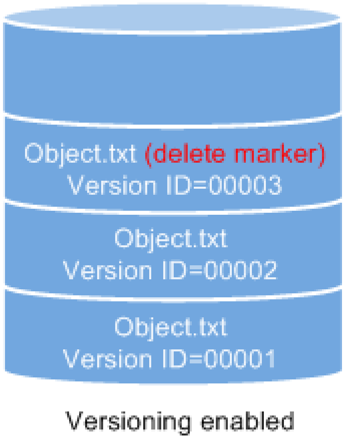
Table 3-35 Version description

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.

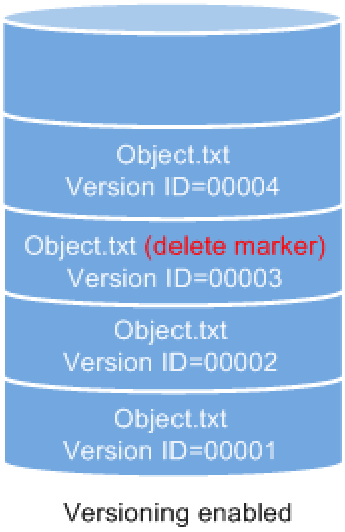
Figure 3-11 Object with a delete marker



- 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 3-12**.

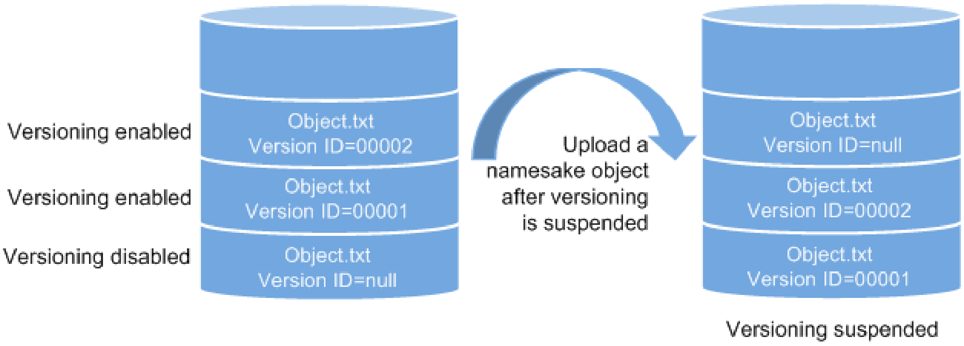
Figure 3-12 Uploading a namesake object after the original one is deleted



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 3-13 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.

3.9.2 Configuring Versioning

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

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

Step 3 In the **Basic Configurations** area, click **Versioning**.

Step 4 Select **Enable**.

Step 5 Click **OK** to enable versioning for the bucket.

Step 6 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.

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

Step 2 In the object list, click the object you want to go to the object details page.

Step 3 On the **Versions** tab page, view all versions of the object.

Step 4 Perform the following operations on object versions:

1. Download a desired version of the object by clicking **Download** in the **Operation** column.
2. Delete a version of the object by clicking **Delete** in the **Operation** column. If you delete the latest version, the most recent version will become the latest version.

----End

3.10 Logging

3.10.1 Logging Overview

You can enable logging to facilitate analysis. 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.

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.

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

```
  ]
}
```

Logs occupy some OBS storage space rented by users. By default, OBS does not collect bucket access logs by default.

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 3-36 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
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)

Parameter	Value Example	Description
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

3.10.2 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 to go to the **Objects** page.

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

Step 3 In the **Basic Configurations** area, click **Logging**. The **Logging** dialog box is displayed.

Step 4 Select **Enable**.

Step 5 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 6 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 7 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.

```
{
    "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 8 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, click **Disable** and then click **OK**. After logging is disabled, logs are not recorded, but existing logs in the target bucket will be retained.

3.11 Event Notifications

3.11.1 SMN-Enabled Event Notifications

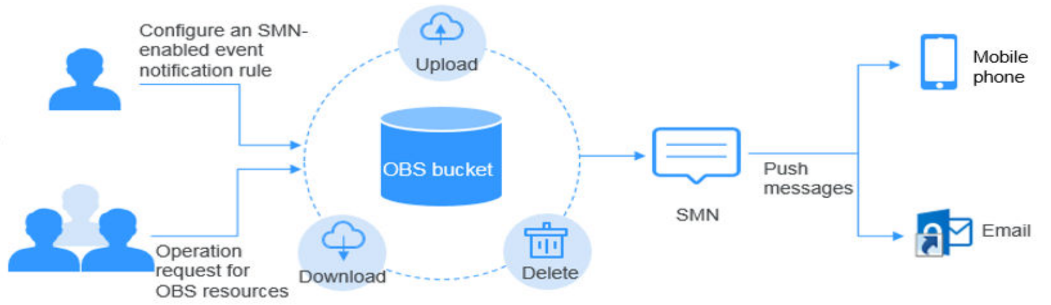
Simple Message Notification (SMN) is a reliable and extensible message notification service that can handle a huge number of messages. It significantly simplifies system coupling and can automatically push messages to endpoints via email or text message.

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 3-14 SMN-enabled event notification


Configure an SMN-enabled event notification rule
Operation request for OBS resources
Upload
OBS bucket
Download
Delete
SMN
Push messages
Mobile phone
Email

3.11.2 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 to go to the **Objects** page.

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

Step 3 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 4 Click **Create**. The **Create Event Notification** dialog box is displayed.

Step 5 Configure event notification parameters, as described in **Table 3-37**.

Table 3-37 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.
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**: Merges 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.

Parameter	Description
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.
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 6 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.

3.11.3 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 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 **Objects** page. Then, 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 3-37**.

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 3-38 Event notification parameters

Parameter	Value
Name	test
Events	ObjectCreated, ObjectRemoved
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

3.12 Cross-Region Replication

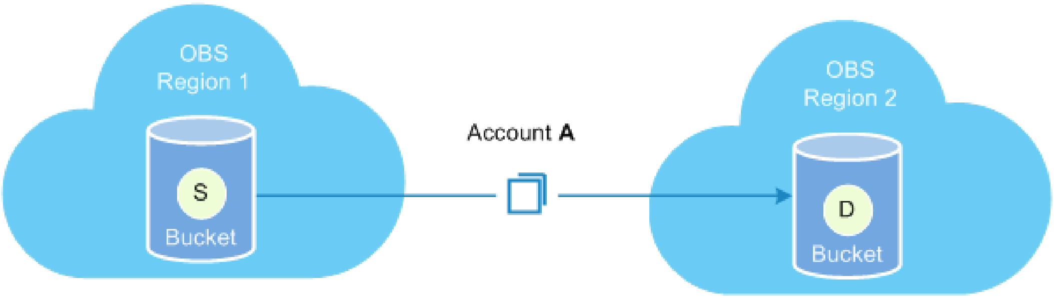
3.12.1 Cross-Region Replication Overview

OBS offers disaster recovery across regions, catering to your needs for remote backup.

Cross-region replication refers to the process of automatically and asynchronously replicating data from a source bucket in one region to a destination bucket in another region based on the created replication rule. Source and destination buckets must belong to the same account. Replication across accounts is currently not supported.

You can configure a rule to replicate only objects with a specified prefix or replicate all objects in a bucket. Replicated objects in the destination bucket are copies of those in the source bucket. Objects in both buckets have the same name and metadata (including the content, size, last modification time, creator, version ID, custom metadata, and ACL).

Figure 3-15 Cross-region replication



Contents Replicated

After the cross-region replication rule is enabled, objects that meet the following conditions are copied to the destination bucket:

- Newly uploaded objects
- Updated objects, for example, objects whose content or ACL information is updated
- Historical objects in a bucket (The function of synchronizing existing objects must be enabled.)

Application Scenarios

- The same OBS resources need to be accessed in different locations. To minimize the access latency, you can use cross-region replication to create object copies in the nearest region.
- Due to business reasons, you need to migrate OBS data to the data center in another region.
- To ensure data security and availability, you need to create explicit backups for all data written to OBS in the data center of another region. Therefore, secure backup data is available if the source data is damaged irrevocably.

Constraints

Cross-region replication has the following constraints:

- Currently, only buckets of version 3.0 support cross-region replication. To check the bucket version, go to the **Overview** page of the bucket on OBS Console. Then you can view the bucket version in the **Basic Information** area.
- By default, objects uploaded before cross-region replication is enabled are not replicated to the destination bucket unless the function for synchronizing existing objects is enabled.

- The source bucket and the destination bucket must belong to different regions separately. Data cannot be copied between buckets in the same region.
- The versioning status of the source and destination buckets must keep the same.
- Objects in a source bucket can be copied to only one destination bucket, and cannot be copied again from the destination bucket to another bucket. For example, bucket A and bucket B are in two different regions. You can copy data from bucket A to bucket B or the other way round. However, data copies in either bucket A or bucket B cannot be replicated anymore.
- Object deletion actions made on the source bucket are usually not synchronized to the destination bucket. The object deletion synchronization will happen only when both the source and destination buckets have versioning enabled and you delete an object from the source bucket without specifying a version.
- For an enabled cross-region replication rule, if you change the versioning status of the destination bucket, the replication of objects will fail. If you want to change the versioning status of the source bucket, delete the replication configuration first, and then make the change.
- Ensure that owners of the source and destination buckets have the read and write permissions to the two buckets. Otherwise, data cannot be synchronized. If the system does not have the permissions to read the source bucket or write the destination bucket due to read/write permission errors, objects cannot be copied successfully, and such replication will not be resumed even if the permission error is rectified.
- For a source bucket, you can create only one cross-region replication rule that applies to the whole bucket for replication of all objects in the bucket. However, you can create a maximum of 100 cross-region replication rules based on object prefixes for the replication of objects that match the prefixes.
- OBS currently only supports the replication between one source bucket and one destination bucket. Replication from one source bucket to multiple destination buckets is not supported. The destination bucket can be modified. However, modifying the destination bucket will change the destination bucket of all existing rules.
- If you delete the OBS agency for an enabled cross-region replication rule, the object replication will be in the **FAILED** status.
- Do not delete, overwrite object replicas in the destination bucket, or modify their ACLs, which may cause inconsistency of latest object versions or permission control settings between the destination bucket and the source bucket.
- If the function for synchronizing existing objects is enabled, modifying the cross-region replication configuration may cause failures in synchronizing existing objects. Therefore, do not modify the cross-region replication configuration before the synchronization finishes.
- If cross-region replication is enabled, data cannot be added to the end of objects in the source bucket.
- If cross-cluster or cross-region replication has been configured for a bucket, configuring another replication policy will overwrite the existing one.
- After a replication with **Synchronize Existing Objects** enabled is complete, if the replication policy keeps unchanged, any ACL changes of source objects

will be synchronized to object copies. However, ACL changes of source historical objects will not be synchronized to the copies of historical objects.

- A bucket configured with two-AZ DR does not support cross-region replication, and vice versa.
- When the active region becomes faulty, buckets cannot be queried, created, or deleted.

3.12.2 Configuring Cross-Region Replication

To replicate objects from a source bucket to a destination bucket in a different region, you can configure a single cross-region replication rule that is applied to all objects in the bucket, or you can configure multiple rules that are applied to a set of objects by specifying a prefix.

NOTE

A cross-region replication rule may not take effect immediately upon its configuration. Accordingly, the objects that this rule is applied to may not be replicated immediately after the rule is configured.

Prerequisites

The source bucket version is 3.0 or later, and cross-region replication is available in the region of the source bucket.

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

Step 2 In the navigation pane, click **Cross-Region Replication**.

Step 3 Click **Create Rule**. The **Create Cross-Region Replication Rule** dialog box is displayed.

Step 4 Configure a cross-region replication rule according to your service needs. For details about the parameters, see **Table 3-39**.

Table 3-39 Cross-region replication parameters

Parameter		Description
Status		Indicates whether the rule is enabled or disabled after being created. The versioning status of the source and destination buckets must keep the same.
Source bucket	Replicate	Indicates the objects the rule will apply to. ● **All objects**: The rule applies to all objects in the bucket. ● **Match by prefix**: The rule applies only to objects with the specified prefix.

Parameter		Description
	Prefix	- To apply the rule to objects with the specified prefix, you must set **Prefix** to a value no longer than 1,024 characters. - If the specified prefix overlaps with the prefix of an existing rule, OBS regards these two rules as one and forbids you to configure the one you are configuring. For example, if there is already a rule with prefix **abc** in OBS, you cannot configure another rule whose prefix starts with **abc**. - To copy a folder, end the prefix with a slash (/), for example, **imgs/**.
	Synchronize Existing Objects	Indicates whether to synchronize the objects that were already in the bucket before the rule configuration to the destination bucket. By default, these objects are not synchronized. Historical objects will be replicated only 15 minutes later a cross-region replication rule was configured.
Destination bucket	Region	Indicates the region of the destination bucket. The destination and source buckets must be in different regions.
	Bucket	Indicates the destination bucket.
Permissions	IAM Agency	Delegates OBS to operate your resources, so that OBS can use this agency to implement cross-region replication. If there is no IAM agency available, click **Create Agency** to create one. If you have already created IAM agencies, select one from the drop-down list. **NOTE** Agency requirements: The IAM agency selected here must be of OBS. The OBS project must have the **Tenant Administrator** permission.

Step 5 (Optional) Create an IAM Agency. For details, see **Creating an Agency**.

Step 6 Click **OK**. The cross-region replication rule is created.

----End

3.13 Cross-Cluster Replication

Scenarios

Cross-cluster replication provides the capability for data disaster recovery across AZs, catering to your needs for cross-AZ backup in the same region.

Each OBS bucket must belong to a region, AZ, and cluster. You can use cross-cluster replication to implement cross-AZ disaster recovery. Specifically, replicate the data in a bucket (source bucket) in one AZ to the bucket (destination bucket) in a cluster in another AZ. If one AZ is faulty, data in the other AZ can still be used to ensure service continuity.

For a cross-cluster replication rule, you can configure it to match a pre-defined object prefix so that objects with the prefix will be replicated. Alternatively, you can configure the rule to apply to the entire bucket so that all objects in the bucket will be replicated. Objects replicated to the destination bucket are precise copies of objects in the source bucket. These objects have the same names, metadata, content, sizes, last modification time, creators, version IDs, user-defined metadata, and ACLs.

Contents Replicated

After the cross-cluster replication rule is enabled, objects that meet the following conditions are copied to the destination bucket:

- Newly uploaded objects
- Updated objects. For example, the object content is updated or the ACL information of a copied object is updated.
- Historical objects in a bucket (The function of synchronizing existing objects must be enabled.)

Constraints

- By default, objects uploaded before cross-cluster replication is enabled are not replicated to the destination bucket.
- The source bucket and destination bucket must belong to different AZs in the same region.
- The versioning status of the source and destination buckets must keep the same.
- Objects in a source bucket can be copied to only one destination bucket, and cannot be copied again from the destination bucket to another bucket. For example, bucket A and bucket B are in two different AZs. You can copy data from bucket A to bucket B or the other way round. However, data copies in either bucket A or bucket B cannot be replicated anymore.
- Only when versioning is enabled for both the source and destination buckets, deleting an object from the source bucket without specifying a version will result in the deletion of the object from the destination bucket. In other situations, deletion operations are not synchronized to the destination bucket.
- For an enabled cross-cluster replication rule, if you change the versioning status of the destination bucket, the replication of objects will fail. If you want to change the versioning status of the source bucket, delete the replication configuration first, and then make the change.
- Ensure that owners of the source and destination buckets have the read and write permissions to the two buckets. Otherwise, data cannot be synchronized. If the system does not have the permissions to read the source bucket or write the destination bucket due to read/write permission errors, objects cannot be copied successfully, and such replication will not be resumed even if the permission error is rectified.

- If the owner of objects in the source bucket is not the owner of the source bucket, the object owner needs to grant the source bucket owner the object read and ACL read permissions of the object through the object ACL. If the source bucket and destination bucket belong to different owners, the destination bucket owner needs to configure a bucket policy that grants the source bucket owner the **ReplicateObject** and **ReplicateDelete** permissions.
- For a source bucket, you can create only one cross-cluster replication rule that applies to the entire bucket for replication of all objects in the bucket. However, you can create a maximum of 100 cross-cluster replication rules based on object prefixes for the replication of objects that match the prefixes.
- OBS currently only supports the replication between one source bucket and one destination bucket. Replication from one source bucket to multiple destination buckets is not supported. The destination bucket can be modified. However, modifying the destination bucket will change the destination bucket of all existing rules.
- If you delete the OBS agency for an enabled cross-cluster replication rule, the object replication will be in the **FAILED** status.
- Do not delete, overwrite object replicas in the destination bucket, or modify their ACLs, which may cause inconsistency of latest object versions or permission control settings between the destination bucket and the source bucket.
- If cross-cluster or cross-region replication has been configured for a bucket, configuring another replication policy will overwrite the existing one.

Procedure

Step 1 In the navigation pane, click **Cross-Cluster Replication**.

Step 2 Click **Create Rule**. The **Create Cross-Cluster Replication Rule** dialog box is displayed.

Step 3 Configure the cross-cluster replication rule according to your service needs. For details about the parameters, see **Table 3-40**.

Table 3-40 Cross-cluster replication parameters

Parameter		Description
Status		Indicates whether the rule is enabled or disabled after being created. The versioning status of the source and destination buckets must keep the same.
Source bucket	Replicate	Indicates the objects the rule will apply to. ● **All objects**: The rule applies to all objects in the bucket. ● **Match by prefix**: The rule applies only to objects with the specified prefix.

Parameter		Description
	Prefix	● To apply the rule to objects with the specified prefix, you must set **Prefix** to a value no longer than 1,023 characters. ● If the specified prefix overlaps with the prefix of an existing rule, OBS regards these two rules as one and forbids you to configure the one you are configuring. For example, if there is already a rule with prefix **abc** in OBS, you cannot configure another rule whose prefix starts with **ab** or **abcd**. ● To copy a folder, end the prefix with a slash (/), for example, **imgs/**.
	Synchronize Existing Objects	Indicates whether to synchronize the objects that were already in the bucket before the rule configuration to the destination bucket. By default, these objects are not synchronized. Historical objects will be replicated only 15 minutes later a cross-cluster replication rule was configured. This option is available only when the administrator enables it in the background.
Destination bucket	Bucket	Indicates the destination bucket. The destination bucket cannot be in the cluster where the source bucket resides. If you want to configure multiple cross-cluster replication rules for a source bucket, you must specify the same destination bucket for all the rules. Modifying the destination bucket of one rule will change the destination bucket of the other rules.
Permissions	IAM Agency	Delegates the resource operation permissions to OBS, so that OBS uses this agency to perform the cross-cluster replication. If no IAM agency is available, refer to **Creating an Agency for Cross-Region or Cross-Cluster Replication** to create one. If you have already created IAM agencies, select one from the drop-down list. **NOTE** Agency requirements: The IAM agency selected here must be of OBS and must have the OBS administrator permission (all OBS permissions).

Step 4 Click **OK**. The cross-cluster replication rule is created.

----End

3.14 Lifecycle Management

3.14.1 Lifecycle Management Overview

Lifecycle management means periodically deleting objects in a bucket by configuring rules.

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).

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

Lifecycle rules have the following key elements:

- Prefix

  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).
- Expiration time

  You can specify the number of days since the last object update after which objects meeting specified conditions are automatically expired and then deleted.

3.14.2 Configuring a Lifecycle Rule

You can configure a lifecycle rule for a bucket or a set of objects to:

- Expire objects and then delete them.

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

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

Step 3 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 4 Click **Create**.

Step 5 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.

- **Prefix**: It is optional.
  - If this field is configured, objects with the specified prefix will be managed by the lifecycle rule. The prefix cannot start with a slash (/) or contain two consecutive slashes (//), and cannot contain the following special characters: **\:*?"<>|**
  - If this field is not configured, all objects in the bucket will be managed by the lifecycle rule.

NOTE

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

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

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.

- **Delete Objects After (Days)**: After this number of days since the last update, objects meeting certain conditions will be expired and then deleted.

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.

NOTE

In theory, it takes 24 hours at most to execute a lifecycle rule. Because OBS calculates the lifecycle of an object from the next 00:00 (UTC time) after the object is uploaded, there may be a delay in deleting expired objects. Generally, the delay does not exceed 48 hours. If you make changes to an existing lifecycle rule, the rule will take effect again.

Step 6 Click **OK** to complete the lifecycle rule configuration.

----End

Follow-up Procedure

You can click **Enable**, **Edit**, or **Disable** in the **Operation** column of a lifecycle rule to enable, edit, or disable the rule.

If you want to delete more than one lifecycle rule at a time, select them and click **Delete** above the list.

3.15 Configuring User-Defined Domain Names

3.15.1 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.

3.15.2 Configuring a User-Defined Domain Name

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

Step 2 In the navigation pane, choose **Domain Name Mgmt**.

Step 3 Click **Configure User Domain Name** in the upper part of the page. Alternatively, click **Configure User Domain Name** in the lower card area of the page when no user-defined domain names are available. In the displayed dialog box, enter the domain name to configure.

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.

----End

3.16 Static Website Hosting

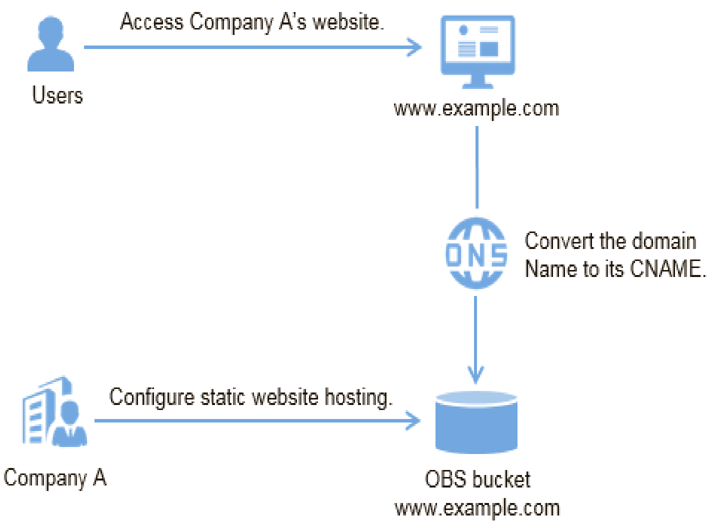
3.16.1 Static Website Hosting Overview

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.

The configuration of static website hosting takes effect within two minutes. After the static website hosting is effective in OBS, you can access the static website by using the URL provided by OBS.

Figure 3-16 Static website hosting



3.16.2 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.

3.16.3 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.

The configuration of static website hosting takes two minutes at most 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 all users.

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

Step 2 (**Optional**) If the static website files in the bucket are not accessible to everyone, perform this step. If they are already accessible to everyone, skip this step.

To grant required permissions, see **Granting Anonymous Users Permission to Access Objects**.

If the bucket contains only static website files, configure the **Object Read-Only** policy for the bucket, so that all files in it are publicly accessible.

1. Choose **Permissions** > **Bucket Policies**.
2. Click **Create**.
3. Configure bucket policy information.

Table 3-41 Parameters for configuring a public read policy

Parameter		Description
Configuration method		**Visual Editor** and **JSON** are available. Choose **Visual Editor** here. For details, see **Creating a Custom Bucket Policy (JSON View)**.
Policy Name		Enter a custom policy name.
Policy content	Effect	Select **Allow**.
	Principals	Select **All accounts**.
	Resources	– Select **Specified objects**. – Set the resource path to * (indicating all objects in the bucket).
	Actions	– Choose **Use a template**. – Select **Object Read-Only**.

4. Click **Create**. The bucket policy is created.

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

Step 4 In the **Basic Configurations** area, click **Static Website Hosting**. The **Static Website Hosting** page is displayed.

Alternatively, you can choose **Basic Configurations** > **Static Website Hosting** from the navigation pane on the left.

Step 5 Click **Configure Static Website Hosting**. The **Configure Static Website Hosting** dialog box is displayed.

Step 6 Enable **Status**.

Step 7 Set the hosting type to the current bucket.

Step 8 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 such a restriction, but the object **Content-Type** 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 such a restriction, but the object **Content-Type** must be specified.

Step 9 **Optional**: In **Redirection Rules**, configure redirection rules. Requests that comply with the redirection rules are redirected to the specific host or page.

A redirection rule is compiled in the JSON or XML format. Each rule contains a **Condition** and a **Redirect**. The parameters are described in **Table 3-42**.

Table 3-42 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.

Container	Key	Description
	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**.
	ReplaceKeyWith	The object name used in the redirection request. OBS replaces the entire object name in the request with the value you specified here for **ReplaceKeyWith**. For example, to redirect requests for all objects in the **docs** directory to **documents/error.html**, set **KeyPrefixEquals** to **docs** under **Condition** and **ReplaceKeyWith** to **documents/error.html** under **Redirect**. This way, requests for both objects **docs/a.html** and **docs/b.html** will be redirected to **documents/error.html**.
	HttpRedirectCode	HTTP status code returned to the redirection request. The default value is **301**, indicating that requests are permanently redirected to the location specified by **Redirect**. You can also set this parameter based on your service needs.

Example of setting a redirection rule

- Example 1: All requests for objects prefixed with **folder1/** are automatically redirected to pages prefixed with **target.html** on host **www.example.com** using HTTPS.

```
[
   {
   "Condition": {
       "KeyPrefixEquals": "folder1/"
       },
   "Redirect":{
       "Protocol": "https",
       "HostName": "www.example.com",
       "ReplaceKeyPrefixWith": "target.html"
       }
   }
]
```

- Example 2: All requests for objects prefixed with **folder2/** are automatically redirected to objects prefixed with **folder/** in the same bucket.

```
[
   {
   "Condition": {
       "KeyPrefixEquals": "folder2/"
       },
   "Redirect":{
       "ReplaceKeyPrefixWith": "folder/"
       }
   }
]
```

- Example 3: All requests for objects prefixed with **folder.html** are automatically redirected to the **folderdeleted.html** object in the same bucket.

```
[
   {
   "Condition": {
       "KeyPrefixEquals": "folder.html"
       },
   "Redirect":{
       "ReplaceKeyWith": "folderdeleted.html"
       }
   }
]
```

- Example 4: If the HTTP status code 404 is returned, the request is automatically redirected to the page prefixed with **report-404/** on host **www.example.com**.

  For example, if you request the page **ExamplePage.html** but the HTTP 404 error is returned, the request will be redirected to the **report-404/ExamplePage.html** page on the **www.example.com**. If the 404 redirection rule is not specified, the default 404 error page configured in the previous step is returned when the HTTP 404 error occurs.

```
[
   {
   "Condition": {
       "HttpErrorCodeReturnedEquals": "404"
       },
   "Redirect":{
       "HostName": "www.example.com",
       "ReplaceKeyPrefixWith": "report-404/"
       }
   }
]
```

Step 10 Click **OK**.

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

3.16.4 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 all users.

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

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

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 **Hosting Type** to **Redirect requests**. In the text box of **Redirect To**, enter the bucket's access domain name or URL.

Step 7 Click **OK**.

Step 8 In the bucket list, click the bucket to which requests for the static website are redirected.

Step 9 (**Optional**) If the static website files in the bucket are not accessible to everyone, perform this step. If they are already accessible to everyone, skip this step.

To grant required permissions, see **Granting Anonymous Users Permission to Access Objects**.

If the bucket contains only static website files, configure the **Object Read-Only** policy for the bucket, so that all files in it are publicly accessible.

1. Choose **Permissions** > **Bucket Policies**.
2. Click **Create**.
3. Configure bucket policy information.

Table 3-43 Parameters for configuring a public read policy

Parameter		Description
Configuration method		**Visual Editor** and **JSON** are available. Choose **Visual Editor** here. For details, see **Creating a Custom Bucket Policy (JSON View)**.
Policy Name		Enter a custom policy name.
Policy content	Effect	Select **Allow**.
	Principals	Select **All accounts**.

Parameter		Description
	Resources	– Select **Specified objects**. – Set the resource path to * (indicating all objects in the bucket).
	Actions	– Choose **Use a template**. – Select **Object Read-Only**.

4. Click **Create**. The bucket policy is created.

Step 10 **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

3.17 Cross-Origin Resource Sharing

3.17.1 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.

3.17.2 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 to go to the **Objects** page.

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

Step 3 In the **Basic Configurations** area, click **CORS Rules**. The **CORS Rules** page is displayed.

Alternatively, you can choose **Permissions** > **CORS Rules** in the navigation pane.

Step 4 Click **Create**. The **Create CORS Rule** dialog box is displayed.

NOTE

A bucket can have a maximum of 100 CORS rules configured.

Step 5 In the **CORS Rule** dialog box, configure **Allowed Origin**, **Allowed Method**, **Allowed Header**, **Exposed Header**, and **Cache Duration (s)**.

Table 3-44 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.

Parameter	Description
Allowed Header	Optional Specifies the allowed headers in cross-origin requests. Only CORS requests matching the allowed headers are valid. You can enter multiple allowed headers (one per line) and each line can contain one wildcard character (*) at most. Spaces and special characters including **&:<** are not allowed.
Exposed Header	Optional Specifies the exposed headers in CORS responses, providing additional information for clients. By default, a browser can access only headers **Content-Length** and **Content-Type**. If the browser wants to access other headers, you need to configure those headers in this parameter. You can enter multiple exposed headers (one per line). Spaces and special characters including ***&:<** are not allowed.
Cache Duration (s)	Mandatory Specifies the duration that your browser can cache CORS responses, expressed in seconds. The default value is **100**.

Step 6 Click **OK**.

Message "The CORS rule created successfully." is displayed. The CORS configuration takes effect within two minutes.

After CORS is successfully configured, only the addresses specified for **Allowed Origin** can access the bucket using the methods specified for **Allowed Method**. For example, you can configure CORS parameters for bucket **testbucket** as follows:

- **Allowed Origin**: **https://www.example.com**
- **Allowed Method**: **GET**
- **Allowed Header**: *
- **Exposed Header**: *
- **Cache Duration (s)**: **100**

By doing so, OBS only allows GET requests from **https://www.example.com** to access bucket **testbucket**, without restrictions on request headers. The client can cache CORS responses for 100 seconds.

----End

3.18 URL Validation

3.18.1 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.

3.18.2 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 to go to the **Objects** page.

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

Step 3 In the **Basic Configurations** area, click **URL Validation**. The **URL Validation** page is displayed.

Step 4 Click ✎ next to the text box of **Whitelisted Referers** or **Blacklisted Referers**, and enter the referers.

Principles for setting **Referers**:

- The length of a whitelist or blacklist cannot exceed 1024 characters.
- Referer format:
  - You can enter multiple referers, each in a line.
  - The referer parameter supports asterisks (*) and question marks (?). An asterisk works as a wildcard that can replace zero or multiple characters, and a question mark (?) can replace a single character.
  - If the referer header field contains **http** or **https** during download, the referer must contain **http** or **https**.
- If **Whitelisted Referers** is left blank but **Blacklisted Referers** is not, all websites except those specified in the blacklist are allowed to access data in the target bucket.

- If **Whitelisted Referers** is not left blank, only the websites specified in the whitelist are allowed to access the target bucket no matter whether **Blacklisted Referers** is left blank or not.

NOTE

If **Whitelisted Referers** is configured the same as **Blacklisted Referers**, the blacklist takes effect. For example, if both **Whitelisted Referers** and **Blacklisted Referers** are set to **https://www.example.com**, access requests from this address will be blocked.

- If **Whitelisted Referers** and **Blacklisted Referers** are both left blank, all websites are allowed to access data in the target bucket by default.
- Before determining whether a user has the four types of permissions (read, write, ACL read, and ACL write) for a bucket or objects in the bucket, check whether this user complies with the URL validation principles of the **Referer** field.

Step 5 Click ✓ to save the settings.

----End

3.19 Task Center

When you upload objects or delete folders, corresponding records of the tasks will be displayed in the task center for you to view the tasks' progress and status.

NOTE

Refreshing or closing the browser will cancel ongoing tasks and clear all records.

Procedure

Step 1 In the object list of your bucket, click **Task Center** in the upper right corner.

Step 2 View the records of uploading objects or deleting folders.

- Click **Clear Records** to clear all task records.
- On the **Upload** tab page, you can click **Pause All** or **Start All** to manage upload tasks in batches.

----End

3.20 Two-AZ DR

3.20.1 Overview

Background Information

An AZ is a physical location with independent power supplies and network in a region. A region contains one or more AZs that are physically isolated but interconnected through internal networks. Because AZs are isolated from each other, any fault that occurs in one AZ will not affect other AZs.

When two-AZ DR is disabled for a bucket, data is stored only in one single AZ at a lower cost. When two-AZ DR is enabled for a bucket, data is stored in two AZs to achieve higher reliability.

Buckets with two-AZ DR enabled are called fusion buckets.

Application Scenario

Two-AZ DR allows you to synchronize data to another AZ for remote backups. It is ideal for scenarios that demand high reliability.

With two-AZ DR, you can store data in two different AZs in the same region. If one AZ becomes unavailable, data can still be properly accessed from the other AZ. This ensures data reliability in case of ventilation, fire-extinguish, moisture-proof, and electricity failures.

Objects synchronized to the destination bucket in the other AZ are precise copies of those in the source bucket. Objects in both buckets have the same name and metadata (including the content, size, last modification time, creator, version ID, custom metadata, and ACL).

Contents Synchronized

When two-AZ DR is enabled, OBS will synchronize the following contents to the other AZ:

- Newly uploaded objects
- Updated objects (for example, the object content is updated)
- Deleted objects. The object copies in the other AZ will also be deleted.

Constraints

- Once configured, a two-AZ DR policy cannot be modified, so carefully plan the two-AZ DR policy in advance.
- A two-AZ DR policy may not take effect immediately upon its configuration. Accordingly, the objects that this policy is applied to may not be synchronized immediately after the policy is configured.
- Only buckets whose version is 3.0 or later support two-AZ DR. The bucket version can be viewed in the **Basic Information** area of the bucket's **Overview** page on OBS Console.
- A bucket configured with two-AZ DR does not support cross-region replication, and vice versa.
- Two-AZ DR is not available for append upload.
- After a fusion bucket is deleted, its name can be used for another bucket only 12 hours later.
- When the active cluster becomes faulty, buckets cannot be queried, created, or deleted.

3.20.2 Configuring Two-AZ DR

This section describes how to configure two-AZ DR for an existing bucket. To configure two-AZ DR for newly created buckets, see **Creating a Bucket**.

Prerequisites

A single-AZ bucket has been created and the bucket has no cross-region replication rule configured.

Procedure

Step 1 In the bucket list, click the bucket you want to operate to go to the **Objects** page.

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

Step 3 In the **Basic Configurations** area, click **Disaster Recovery**.

Step 4 In the displayed **Disaster Recovery** dialog box, select **Enable** and click **OK**.

NOTE

With disaster recovery enabled, data is stored in two AZs in the same region. This delivers a higher reliability but costs more. Once enabled, two-AZ DR cannot be disabled.

----End

3.21 Related Operations

3.21.1 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 Click **Create Agency**.

Step 3 Enter an agency name.

Step 4 Select **Cloud service** for the **Agency Type**.

Step 5 Select **Object Storage Service (OBS)** for **Cloud Service**.

Step 6 Set a validity period.

Step 7 Click **Next**.

Step 8 On the **Select Policy/Role** page, select a custom policy that has the permission to upload data to the log storage bucket and click **Next**.

If no custom policy is available, create one by choosing **Permissions** > **Policies/Roles** in the navigation pane

Select **JSON** for **Policy View**. The policy content is as follows.

NOTE

When coding the policy content in an actual scenario, replace **mybucketlogs** with the actual bucket name:

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

Step 9 On the **Select Scope** page, select **Global services** for **Scope** and click **OK**.

----End

Creating an Agency for Cross-Region or Cross-Cluster Replication

Step 1 Go to the **Agencies** page on the **Identity and Access Management** console.

Step 2 Click **Create Agency**.

Step 3 Enter an agency name.

Step 4 Select **Cloud service** for the **Agency Type**.

Step 5 Select **Object Storage Service (OBS)** as the cloud service.

Step 6 Set a validity period.

Step 7 In the **Permissions** area, find **Global service** > **OBS** and click **Attach Policy** on the right.

Step 8 Select the custom policy with OBS administrator permissions (all operation permissions on OBS) and click **OK**.

1. If no custom policy is available, create one by choosing **Permissions** > **Policies/Roles** in the navigation pane

   When creating a custom policy, select **Global services** for **Scope** and select **JSON** for **Policy View**. The policy content is as follows.

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

Step 9 Select the scope where the permission applies to. In the permission area, search for and select **OBS Administrator**.

Step 10 Click **OK** to complete the agency creation.

----End

3.22 Troubleshooting

3.22.1 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.

3.22.2 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**.

Step 3 Click **OK**.

----End

3.22.3 The Object Name Changes After an Object with a Long Name Is Downloaded to a Local Computer

Question

After an object with a relatively long name is downloaded to a local path, the object name changes.

Answer

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.

3.22.4 Failed to Configure Event Notifications

Question

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.

Answer

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*.

3.22.5 Time Difference Is Longer Than 15 Minutes Between the Client and Server

Question

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.

Answer

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.

3.22.6 Error Code 500 Is Returned When a Bucket with the Name of the Bucket Failing to Be Created in AZ 2 Is Created in AZ 1

Question

In a dual-AZ environment, when AZ 2 is powered off, a request for creating a bucket in AZ 2 is sent to AZ 1. As a result, the bucket creation fails. Then, another request is sent to AZ 1 for creating a bucket in AZ 1, with the name of the one failing to be created in AZ 2. Then, error code 500 is returned.

Answer

After the bucket fails to be created in AZ 2, the bucket information will be stored in AZ 2 for 30 minutes. If another bucket with the same name as the one failing to be created is created in the other AZ within 30 minutes, the creation will fail. You are advised to create a bucket with the name of the bucket failing to be created 30 minutes later after the bucket creation failure, or use a new name to create one.

3.22.7 Object Upload or Download Errors

This section describes the common errors you may encounter when using OBS Console to upload or download objects (usually larger than 8 MB) and the corresponding solution. Google Chrome is used as an example in the following scenarios.

Scenario 1

Symptom

During object upload or download, error "ERR_NAME_NOT_RESOLVED" is displayed, indicating that the server's IP address cannot be found.

Solution

Configure the local **hosts** file by referring to **Configuring the Local hosts File**.

NOTE

The local **hosts** file should be configured in the *[OBS backend IP address] [Bucket domain name]* format. An example is given as follows:

```
xx.xx.xx.xx bucket-name.obsv3.example-region.com
```

In this example:

[OBS backend IP address] indicates the value of **obsv3_address** provided in the "1.1 Basic Parameters" sheet of the parameter summary file exported during OBS installation.

[Bucket domain name] can be found on the bucket's basic information page on the console.

After the configuration is complete, refresh the page and message "Your connection isn't private" may be displayed. For more information, see **Scenario 3**.

Scenario 2

Symptom

During object upload or download, error "ERR_CONNECTION_TIMED_OUT" is displayed, indicating that the connection timed out.

Solution

Check whether the local **hosts** file is incorrectly configured by referring to **Configuring the Local hosts File**, or check whether the IP address is correct and network connectivity is normal.

To check the IP address, do as follows:

Open a new browser window and enter the bucket domain name (an address starting with **https://**). Then, the browser should display a warning that your connection is not private.

Skip the warning. If the XML messages are displayed, the IP address is correct. If the messages are of any other type, further check whether the IP address is correct.

Scenario 3

Symptom

During object upload or download, error "NET::ERR_CERT_AUTHORITY_INVALID" is displayed, indicating a privacy error.

Solution

This happens because OBS has no commercial certificate and cannot be trusted by the browser. The workaround for this problem is to click **Advanced** at the bottom of the page and click "Continue to *XXX*".

To completely resolve this problem, download and install the OBS certificate. When installing the certificate, select **Place all certificates in the following store (Trusted Root Certification Authorities)**.

3.23 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 3-45** lists some common error codes in OBS error responses.

Table 3-45 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.

Error Code	Description
Obs.0003	The connection timed out.
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.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.
Obs.0017	The end time of the new validity period must be later than that of the old validity period.

Error Code	Description
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.

4 FAQ

4.1 OBS Basics

4.1.1 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.

4.1.2 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.

4.1.3 Which Types of Data Can Be Stored in OBS?

OBS can store all types of data.

4.1.4 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 the size of a file exceeds 5 GB, use tools (such as OBS Browser+) or the multipart interface of OBS APIs to upload the file.
- If you use OBS Browser+ or an API, you can upload a single object of up to 48.8 TB.

4.1.5 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.

4.1.6 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.

4.1.7 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.

4.1.8 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.

4.1.9 Does OBS Support Resumable Transfer?

Resumable transfer is supported for all transfer methods except API.

Table 4-1 Support for resumable transfer by different OBS tools

OBS Tool	Resumable Data Transfer
OBS Console	Not supported
OBS Browser+	Supported
APIs	Not supported

4.1.10 Does OBS Support Batch Upload?

The following table lists the batch upload support for different OBS tools.

Table 4-2 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.
APIs	Not supported

4.1.11 Does OBS Support Batch Download?

The following table lists the batch download support for different OBS tools.

Table 4-3 Support for batch download by different OBS tools

Tool	Batch Download
OBS Console	Not supported
OBS Browser+	Supported
APIs	Not supported

4.1.12 Does OBS Support Batch Deletion of Objects?

The following table lists the batch deletion support for different OBS tools.

Table 4-4 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. For details, see **Deleting an Object or Folder**.
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.
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 delay.

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.

4.1.13 What Are the Factors That Affect Upload and Download Speeds of OBS?

The OBS upload and download speeds may be affected by:

- Upper limit of the read/write bandwidth allowed by a single account
- Disk I/O, NICs, and resources consumed by other processes

4.1.14 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.

4.1.15 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.

4.1.16 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.

4.1.17 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.

4.2 Access Control

4.2.1 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.

4.2.2 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 in OBS.

A bucket policy only applies to the bucket the policy was configured for.

4.2.3 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.

4.3 Buckets and Objects

4.3.1 Why Am I Unable to Create a Bucket?

- If the number of buckets created 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 network connectivity between the local computer and OBS is normal. If the network is faulty, restore the network connection.
- If one of the above are the cause, check the returned error code to further locate the fault.

4.3.2 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 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 the administrator.

4.3.3 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 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.
- If the fault persists, contact the administrator.

4.3.4 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 the administrator.

4.3.5 Can I Rename an Object?

Yes. You can rename objects in a bucket one at a time, but currently only using OBS Browser+.

4.3.6 Can I Modify the Region of a Bucket?

No. After a bucket is created, the region cannot be changed.

4.3.7 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 4-5 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.
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.

4.3.8 Why Can't I Search for Certain Objects in My Bucket?

On OBS Console and OBS Browser 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.

4.3.9 Does OBS Support Storage Quota Management?

OBS allows you to manage the storage quota for buckets.

By default, the storage space of a bucket is not limited. You can configure a quota to limit a bucket's storage space that can be used. If the configured storage quota is used up, object upload will fail.

Quota management is currently not available for parallel file systems. By default, there is no quota limit.

⚠ CAUTION

A quota takes effect 15 minutes after its configuration is complete. OBS checks whether the quota has been used up 15 minutes later each time the storage statistics are collected. If OBS checks that the storage quota has been used up when objects are being continuously written to a bucket, no more objects can be written to the bucket. Therefore, in some cases, the actual capacity of a bucket may be slightly greater than the configured storage quota.

Bucket Quota Configuration

Manage bucket storage quotas using OBS Console and APIs.

For details about how to configure a storage quota using APIs, see the API for quota configuration in the *API Reference*.

The following procedure describes how to configure a storage quota on OBS Console.

Step 1 In the navigation pane, choose **Overview**.

Step 2 In the **Basic Configurations** area, click the **Storage Limit** card to change the limit on the current bucket's storage.

By default, the storage capacity is not limited. The minimum quota allowed is 1 MB.

Step 3 Click **OK**.

----End

4.4 Security

4.4.1 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.

4.4.2 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.

4.4.3 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).

4.4.4 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.

4.4.5 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.

4.4.6 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.

4.5 Durability and Availability

4.5.1 How Are the Durability and Availability of OBS?

OBS provides multi-level reliability assurance for storage media, servers, cabinets, data centers, and regions by leveraging the following technologies: slow disk or bad sector detection, intra-AZ device and data redundancy, cross-AZ data disaster recovery, cross-region replication, and more. It offers up to 99.9999999999% (12 nines) of durability and up to 99.95% of availability for multi-AZ storage, much higher than those of the conventional architecture.

The 12 nines of durability means that the average annual loss rate of objects is expected to be 0.0000000001%. For example, if you store 100 million objects in OBS, only one object may be lost every 10,000 years.

Availability is the service continuity. A 99.95% availability means that if you keep accessing OBS for 10,000 minutes (about 7 days), you can expect no more than 5 minutes of unavailability.

4.5.2 What Are the Differences Between Single-AZ and Multi-AZ Storage in OBS?

Question 1:

Q: For selecting data redundancy policy upon bucket creation, what is the difference between single-AZ storage and multi-AZ storage?

A: Multi-AZ storage means data is stored in multiple AZs, improving data reliability. Single-AZ storage means data is stored in a single AZ, with lower costs.

Question 2:

Q: Is data stored as copies in multiple AZs when the data redundancy policy is set to multi-AZ storage? If an AZ is faulty, is data complete in other AZs?

A: The Erasure Code (EC) algorithm, instead of multiple copies, is used to ensure data redundancy in the multi-AZ mode. If the multi-AZ storage is enabled for a bucket, data is stored in multiple AZs in the same region. If an AZ is unavailable, data can still be properly accessed in other AZs. The multi-AZ mode is suitable for data storage scenarios that require high reliability. The multi-AZ storage tolerates only faults of a single AZ.

Question 3:

Q: Can I change the data redundancy policy without deleting the bucket?

A: No. Once a bucket is created, you cannot change the data redundancy policy. You can create a bucket with the wanted data redundancy policy, and migrate data to the new bucket.

4.5.3 What Redundancy Techniques Does OBS Use?

OBS uses the erasure coding (EC) algorithm, instead of multiple copies, to ensure data redundancy.

Compared with the multi-copy redundancy, EC delivers a higher storage space utilization while maintaining the same reliability level.

A bucket with single-AZ storage uses the EC algorithm for data redundancy among nodes in one AZ. A bucket with multi-AZ storage not only ensures redundancy for the data among nodes in an AZ, but also across multiple AZs.

4.5.4 What Are the OBS SLA and Its Constraints?

A service level agreement (SLA) documents what service quality a provider will furnish and defines the uptime guarantee (service availability) the provider is obligated to meet. The OBS SLA ensures a service availability of no less than 99.9% for OBS Standard single-AZ storage in each service period and of no less than 99.95% for OBS Standard multi-AZ storage in each service period. The SLA availability achievement strongly relies on the hardware replacement speed. If the availability is 99.95% (indicating an annual service interruption tolerance of no more than 4.38 hours), hardware replacement must be completed within 8 hours. If the availability is 99.9% (indicating an annual service interruption tolerance of no more than 8.76 hours), hardware replacement must be completed within 30 hours.

The OBS SLA does not apply to any of the following performance or availability issues:

1. Edge access failures due to network faults (such as edge node network disconnection and carrier line latency and jitter or faults), hardware faults (such as device's power supply faults, as well as hard disk, memory, and backplane faults), data center faults (such as power outages, typhoons, floods, earthquakes, and epidemics), and others.
2. Issues of devices that are deployed in a customer's data center and depend on the infrastructure and network provided by the customer.

4.6 Fragment Management

4.6.1 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.

4.6.2 How Do I Manage Fragments?

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.

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.

4.7 Versioning

4.7.1 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.

4.7.2 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**.

4.8 Event Notifications

4.8.1 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**: Merges 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.

For details about how to configure event notifications, see **Configuring Event Notifications**.

4.9 How Do I Use Lifecycle Management?

4.9.1 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).

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 4-6** lists the parameters for configuring such a lifecycle rule on OBS Console.

Table 4-6 Parameters for deletion upon expiration

Parameter	Value
Status	Enable
Rule Name	Example: **rule-delete**

Parameter		Value
Prefix		Optional. - If this parameter is configured, objects with the specified prefix will be deleted in a batch. - If this parameter is not configured, all objects in the bucket will be deleted.
Current Version	Delete Objects After (Days)	1 day
Historical Version	Delete Objects After (Days)	1 day

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.

4.10 How Do I Use Static Website Hosting?

4.10.1 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.

4.10.2 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.

4.10.3 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*

4.11 How Do I Use Cross-Region Replication?

4.11.1 What Are the Application Scenarios of Cross-Region Replication?

- The same OBS resources need to be accessed in different locations. To minimize the access latency, you can use cross-region replication to create object copies in the nearest region.
- Due to business reasons, you need to migrate OBS data to the data center in another region.
- To ensure data security and availability, you need to create explicit backups for all data written to OBS in the data center of another region. Therefore, secure backup data is available if the source data is damaged irrevocably.

4.11.2 Will an Object Deletion in a Source Bucket Be Synchronized to the Destination Bucket?

No. Object deletion is not synchronized.

After the cross-region replication rule is enabled, objects that meet the following conditions are copied to the destination bucket:

- Newly uploaded objects
- Updated objects, for example, objects whose content or ACL information is updated
- Historical objects in a bucket (The function of synchronizing existing objects must be enabled.)

4.11.3 Why Objects Are Not Copied to the Destination Bucket After the Cross-Region Replication Rule Has Been Created?

- If the function of synchronizing existing objects is not enabled for a cross-region replication rule, existing objects in a bucket will not be copied to the destination bucket.
- A cross-region replication rule may not take effect immediately upon its configuration. Accordingly, the objects that this rule is applied to may not be replicated immediately after the rule is configured.

A Change History

Released On	Description
2024-04-15	This is the first official release.