Object Storage Service

Console Operation Guide

Issue	01
Date	2025-07-31

Contents

1 Console Function Overview

Table 1-1 lists functions provided by OBS Console.

Table 1-1 OBS Console functions

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

Function	Description
Fragment management	Manages and clears fragments generated due to object upload failures.
Versioning	Stores multiple versions of an object in the same bucket.
Logging	Logs bucket access requests for analysis and auditing.
Permission control	Controls access to OBS using IAM permissions, bucket/object policies, and bucket/object access control lists (ACLs).
Lifecycle management	Allows you to configure lifecycle rules to periodically expire and delete objects or transition objects between storage classes.
Tags	Help you identify and classify buckets in OBS.
Static website hosting	Supports the hosting of static websites in buckets and the redirection of access requests for buckets.
User-defined domain name configuration	Enables you to bind your website domain name to a bucket domain name. If you want to migrate files from your website to OBS while keeping the website address unchanged, you can use this function.
Back to source	Helps you obtain the requested data from the source site if the data is not found in OBS. Usually, if the data you requested is not found in OBS, a 404 error will be returned.
URL validation	Prevents object links in OBS from being stolen by other websites.
Cross origin resource sharing	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).
Direct reading	Allows you to directly download objects in the Archive storage class without restoring them first. Direct reading is a billable function.
Bucket inventory	Periodically provides CSV files that list object information in the bucket and delivers the CSV files to the specified bucket.

2 Web Browser Compatibility

Table 2-1 lists the web browser versions compatible with OBS Console.

Table 2-1 Supported web browser versions

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

3 Getting Started

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
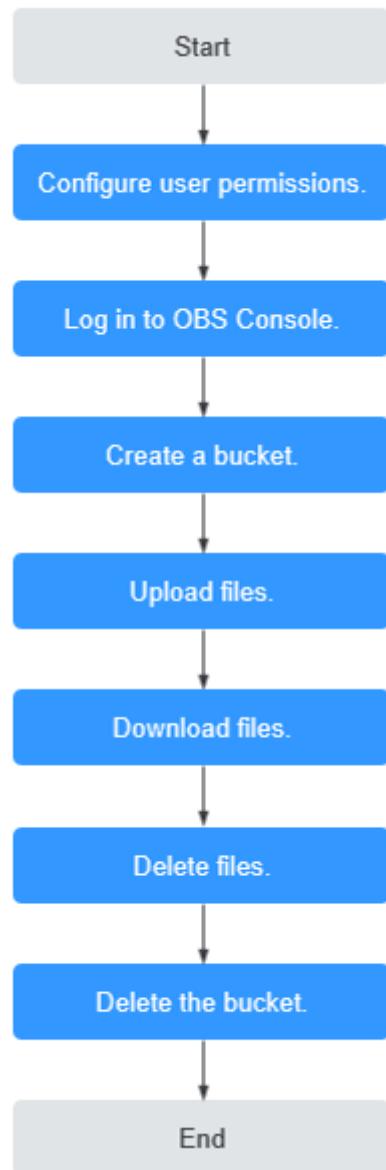
3.2 Configuring User Permissions

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

The below example describes how to grant an IAM user the **Tenant Guest** permission for OBS.

1. **Create a user group and assign permissions**.

   Create a user group on the IAM console, and assign the group the **Tenant Guest** permission.

2. **Create an IAM user and add it to the user group**.

   Create a user on the IAM console and add the user to the group created in **1**.

3. **Log in** and verify the permission granting.

   Log in to OBS Console using the newly created user, and verify that the assigned permission has taken effect:

   – Choose **Object Storage Service** from the service list to go to the OBS homepage. If the list of buckets is displayed and you can view the basic information about any bucket, but you cannot create or delete buckets or perform any other operations, the granted **Tenant Guest** permission has already taken effect.

   – Go to an OBS bucket. If the list of objects is displayed and you can download objects, but you cannot upload or delete objects or perform any other operations, the **Tenant Guest** permission granted has already taken effect.

3.3 Logging In to OBS Console

You can log in to OBS Console using a web browser.

Procedure

Step 1 Visit the **Huawei Cloud official website**.

Step 2 Create a HUAWEI ID.

If you already have one, start from **Step 3**.

1. On the right of the top navigation bar, click **Sign Up**.
2. Complete the creation as instructed.
   After the creation is complete, you will be navigated to your information page.

Step 3 On the right of the top navigation menu, click **Log In**, and enter the username and password.

Step 4 On the right of the top navigation bar, click **Console** to go to the management console.

Step 5 In the upper left corner of the navigation pane, click ☰ and choose **Storage** > **Object Storage Service**. The OBS Console page is displayed.

Step 6 (Recommended) Top up your account or buy OBS resource packages, for you to properly use OBS.

----End

3.4 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 (including all IAM users under this account) can create a maximum of 100 buckets and parallel file systems. You can use the fine-grained access control of OBS to properly plan and use buckets. For example, you can create folders in a bucket based on object prefixes and use **fine-grained permission control** to isolate permissions between departments.

Procedure

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

Step 2 In the upper right corner, click **Create Bucket**. The **Create Bucket** page is displayed. For details, see **Figure 3-3**.

Figure 3-3 Creating a bucket
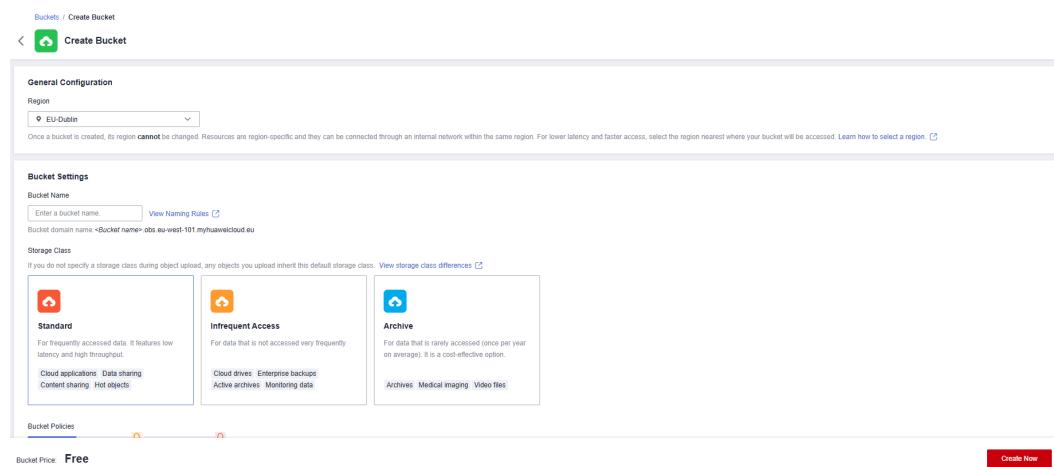
Step 3 Configure bucket parameters.

Table 3-1 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.

Bucket Name	Name of the bucket. A bucket name must be unique across all accounts and regions. Once a bucket is created, its name cannot be changed.
	
	According to the globally applied DNS naming rules, an OBS bucket name:
	
	● Must be unique across all accounts and regions. The name of a deleted bucket can be reused for another bucket or a parallel file system at least 30 minutes after the deletion.
	
	● Must be 3 to 63 characters long. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
	
	● Cannot start or end with a period (.) or hyphen (-), and cannot contain two consecutive periods (..) or contain a period (.) and a hyphen (-) adjacent to each other.
	
	● Cannot be formatted as an IP address.
	
	    NOTE
	    When you access OBS through HTTPS using virtual hosted-style URLs, if the bucket name contains a period (.), the certificate verification will fail. To work around this issue, you are advised not to use periods (.) in bucket names.

Parameter	Description
Storage Class	Storage classes of a bucket. Different storage classes meet different requirements for storage performance and costs. ● The Standard storage class is for storing a large number of hot files or small files that are frequently accessed (multiple times per month on average) and require quick retrieval. ● The Infrequent Access storage class is for storing data that is less frequently accessed (less than 12 times per year on average) and requires quick retrieval. ● The Archive storage class is for archiving data that is rarely accessed (once a year on average) and has no requirements for quick retrieval. For details, see **Storage Classes**.
Bucket Policy	Controls read and write permissions for buckets. ● **Private**: No access beyond the bucket ACL settings is granted. ● **Public Read**: Anyone can read objects in the bucket. ● **Public Read and Write**: Anyone can read, write, or delete objects in the bucket.
Server-Side Encryption	After you enable server-side encryption for the bucket, any object you upload to it will inherit the KMS encryption from the bucket by default. After you enable server-side encryption for the bucket, any object you upload to it will be encrypted with the **obs/default** key by default. You can also click **Create KMS Key** to create a key on the DEW console. Then select the created key on OBS Console for encryption.

Parameter	Description
WORM	When you enable write-once-read-many (WORM), you can configure a retention policy for the current bucket. The object version which the retention policy is applied to cannot be deleted within a specified period. You can only enable WORM when you create a bucket. Once enabled for a bucket, WORM cannot be disabled. When you enable WORM, OBS automatically enables versioning for the bucket, and versioning cannot be suspended later for that bucket. If you choose to enable WORM, configure the following: ● **Default Retention Mode**: This parameter needs to be configured if **Default Retention** is set to **Configure**. Only the compliance retention mode is currently supported. In this mode, no users can delete protected object versions or change their retention mode during the specified retention period. ● **Default Retention Period**: This parameter needs to be configured if **Default Retention** is set to **Configure**. During the specified period, OBS prevents WORM-protected object versions from being deleted. You can configure a retention period in either days (from **1** to **36500**) or years (from **1** to **100**).
Direct Reading	Direct reading allows you to directly download objects from the Archive storage class without restoring them first. Direct reading is a billable function. For details, see **Product Pricing Details**. No matter which default storage class you select, you can enable direct reading for your bucket. For example, if you select the Standard storage class and enable direct reading for your bucket, you can directly download objects stored in the Archive storage class from your bucket.
Enterprise Project	You can add a bucket to an enterprise project for unified management. Create an enterprise project by referring to **Creating an Enterprise Project**. The default enterprise project is named **default**. On the **Enterprise Project Management** page, **create an enterprise project**, and **add a user group to the enterprise project**. By doing so, users in this user group obtain the operation permissions for the buckets and objects in the enterprise project. NOTE Only an enterprise account can configure enterprise projects. OBS ReadOnlyAccess and OBS OperateAccess are the fine-grained authorizations of the enterprise project user group in OBS.

Parameter	Description
Tags	Optional. Tags are used to identify and classify buckets in OBS. Each tag is represented by a key-value pair. For more information, see **Tags**.
Purchase Resource Packages	Optional. By default, you are billed on a pay-per-use basis for using OBS. You can purchase **storage packages** in advance to save more. Storage packages can also be purchased after buckets are created. If you select **Purchase Resource Packages**, you can also select **Only used for the enterprise project you selected for the bucket**. **Only used for the enterprise project you selected for the bucket**: Selecting this option applies the purchased packages only to your bucket's enterprise project.

Step 4 Click **Create Now**.

Step 5 If **Purchase Resource Packages** is selected during bucket creation, confirm the package information on the **Confirm** page, including quantity, required duration, read and agree to the service agreement, and click **Submit**.

Step 6 Click **Confirm**.

----End

3.5 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 puts limits on the size and number of files you can upload.

- In regions that support batch uploads, a maximum of 100 files can be uploaded at a time, with a total size of no more than 5 GB.

- In regions that do not support batch uploads, only one file can be uploaded at a time, with a size of no more than 50 MB.

Therefore, for a single file to be uploaded, its maximum size can be 5 GB in a batch upload or 50 MB in a single upload.

OBS Browser+ allows you to upload up to 500 files at a time. There is no limit on the number of files you can upload using obsutil at a time.

📖 **NOTE**

Batch upload is available only when the following condition is met:

The bucket version is 3.0. To view the bucket version, see **Viewing Basic Information of a Bucket**.

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**.

Prerequisites

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

Procedure

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

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

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

Batch upload is used as an example here. If the region you are using supports only single upload, perform operations as instructed.

📖 **NOTE**

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

Figure 3-4 Uploading objects

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

☐ NOTE

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

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

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

Step 6   **Server-Side Encryption**: If server-side encryption is enabled, new objects uploaded to this bucket will be automatically encrypted. For details, see **Enabling Server-Side Encryption When Uploading an Object**.

☐ NOTE

If your bucket has **Server-Side Encryption** configured, you can select **Inherit from bucket** when uploading an object.

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

☐ NOTE

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

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

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

Configuring tags: Tags are used to identify objects with tag keys and values. For details, see **Adding Tags to an Object**.

Figure 3-5 Configuring metadata or WORM retention
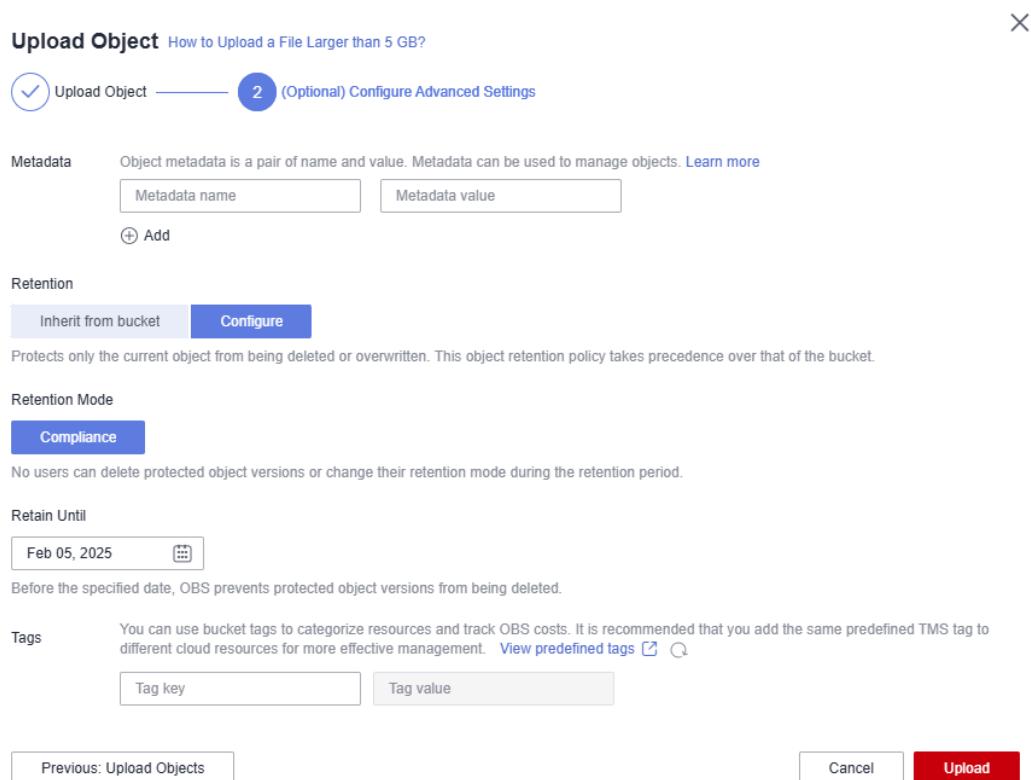
Step 8 Click **Upload**.

----End

3.6 Downloading an Object

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

Constraints

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

Procedure

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

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

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

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

----End

3.7 Deleting an Object

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

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

Procedure

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

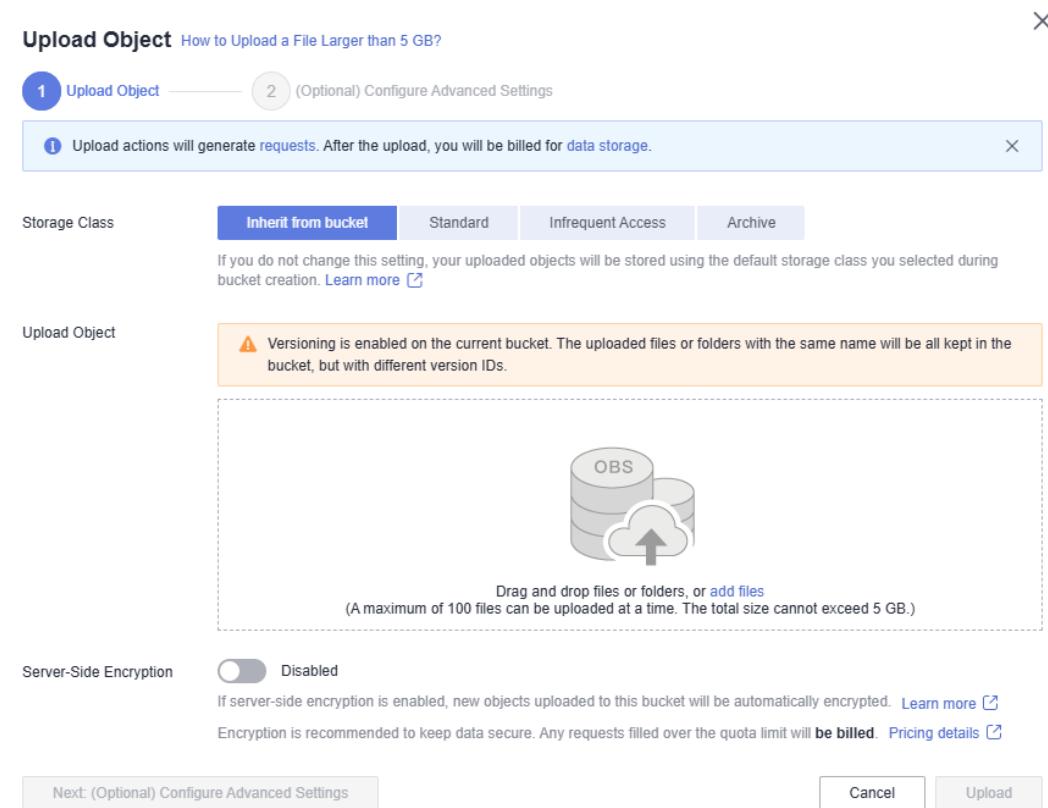**Step 2** In the bucket list, click the bucket you want to operate to go to the **Objects** page.

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

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

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

----End

Important Notes

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

3.8 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 navigation pane of **OBS Console**, choose **Object Storage**.

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

NOTE

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

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

----End

4 Managing Buckets

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 (including all IAM users under this account) can create a maximum of 100 buckets and parallel file systems. You can use the fine-grained access control of OBS to properly plan and use buckets. For example, you can create folders in a bucket based on object prefixes and use **fine-grained permission control** to isolate permissions between departments.

Procedure

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

Step 2 In the upper right corner, click **Create Bucket**. The **Create Bucket** page is displayed. For details, see **Figure 4-1**.

Figure 4-1 Creating a bucket
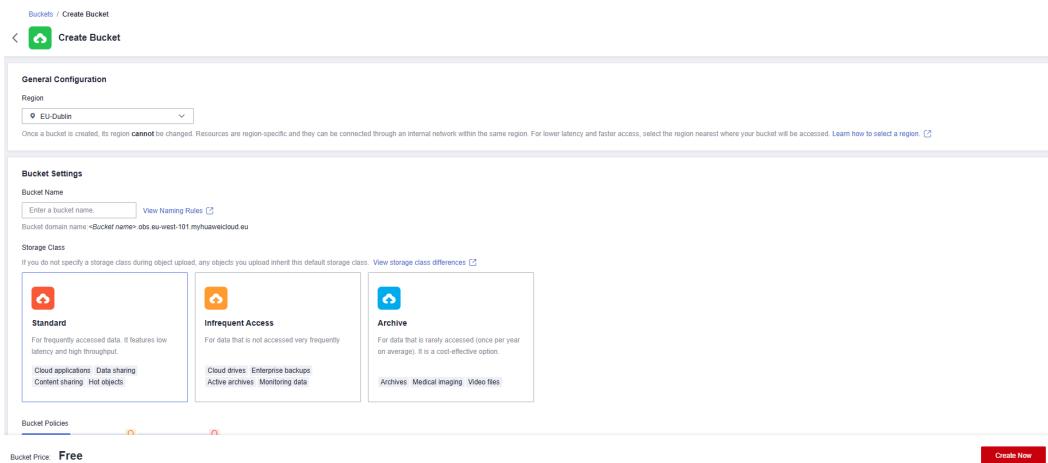
Step 3 Configure bucket parameters.

Table 4-1 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.
Bucket Name	Name of the bucket. A bucket name must be unique across all accounts and regions. Once a bucket is created, its name cannot be changed. According to the globally applied DNS naming rules, an OBS bucket name: ● Must be unique across all accounts and regions. The name of a deleted bucket can be reused for another bucket or a parallel file system at least 30 minutes after the deletion. ● Must be 3 to 63 characters long. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. ● Cannot start or end with a period (.) or hyphen (-), and cannot contain two consecutive periods (..) or contain a period (.) and a hyphen (-) adjacent to each other. ● Cannot be formatted as an IP address. **NOTE** When you access OBS through HTTPS using virtual hosted-style URLs, if the bucket name contains a period (.), the certificate verification will fail. To work around this issue, you are advised not to use periods (.) in bucket names.
Storage Class	Storage classes of a bucket. Different storage classes meet different requirements for storage performance and costs. ● The Standard storage class is for storing a large number of hot files or small files that are frequently accessed (multiple times per month on average) and require quick retrieval. ● The Infrequent Access storage class is for storing data that is less frequently accessed (less than 12 times per year on average) and requires quick retrieval. ● The Archive storage class is for archiving data that is rarely accessed (once a year on average) and has no requirements for quick retrieval. For details, see **Storage Classes**.
Bucket Policy	Controls read and write permissions for buckets. ● **Private**: No access beyond the bucket ACL settings is granted. ● **Public Read**: Anyone can read objects in the bucket. ● **Public Read and Write**: Anyone can read, write, or delete objects in the bucket.

Parameter	Description
Server-Side Encryption	After you enable server-side encryption for the bucket, any object you upload to it will inherit the KMS encryption from the bucket by default.
	After you enable server-side encryption for the bucket, any object you upload to it will be encrypted with the **obs/default** key by default. You can also click **Create KMS Key** to create a key on the DEW console. Then select the created key on OBS Console for encryption.
WORM	When you enable write-once-read-many (WORM), you can configure a retention policy for the current bucket. The object version which the retention policy is applied to cannot be deleted within a specified period. You can only enable WORM when you create a bucket. Once enabled for a bucket, WORM cannot be disabled. When you enable WORM, OBS automatically enables versioning for the bucket, and versioning cannot be suspended later for that bucket.
	If you choose to enable WORM, configure the following:
	• **Default Retention Mode**: This parameter needs to be configured if **Default Retention** is set to **Configure**. Only the compliance retention mode is currently supported. In this mode, no users can delete protected object versions or change their retention mode during the specified retention period.
	• **Default Retention Period**: This parameter needs to be configured if **Default Retention** is set to **Configure**. During the specified period, OBS prevents WORM-protected object versions from being deleted. You can configure a retention period in either days (from **1** to **36500**) or years (from **1** to **100**).
Direct Reading	Direct reading allows you to directly download objects from the Archive storage class without restoring them first. Direct reading is a billable function. For details, see **Product Pricing Details**.
	No matter which default storage class you select, you can enable direct reading for your bucket. For example, if you select the Standard storage class and enable direct reading for your bucket, you can directly download objects stored in the Archive storage class from your bucket.

Parameter	Description
Enterprise Project	You can add a bucket to an enterprise project for unified management.
	Create an enterprise project by referring to **Creating an Enterprise Project**. The default enterprise project is named **default**.

	On the **Enterprise Project Management** page, **create an enterprise project**, and **add a user group to the enterprise project**. By doing so, users in this user group obtain the operation permissions for the buckets and objects in the enterprise project.
	NOTE
	  Only an enterprise account can configure enterprise projects.
	  OBS ReadOnlyAccess and OBS OperateAccess are the fine-grained authorizations of the enterprise project user group in OBS.

Tags	Optional. Tags are used to identify and classify buckets in OBS. Each tag is represented by a key-value pair.
	For more information, see **Tags**.
Purchase Resource Packages	Optional. By default, you are billed on a pay-per-use basis for using OBS. You can purchase **storage packages** in advance to save more.
	Storage packages can also be purchased after buckets are created.
	If you select **Purchase Resource Packages**, you can also select **Only used for the enterprise project you selected for the bucket**.
	Only used for the enterprise project you selected for the bucket: Selecting this option applies the purchased packages only to your bucket's enterprise project.

Step 4 Click **Create Now**.

Step 5 If **Purchase Resource Packages** is selected during bucket creation, confirm the package information on the **Confirm** page, including quantity, required duration, read and agree to the service agreement, and click **Submit**.

Step 6 Click **Confirm**.

----End

Related Operations

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

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

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

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

　　📖 NOTE

- Changing the storage class of a bucket does not change the storage class of existing objects in the bucket.

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

----End

4.2 Viewing Basic Information of a Bucket

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

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

Viewing Bucket Details

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

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

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

Step 4 In the **Usage Statistics** area, view the storage, traffic, and request information of the bucket.

Table 4-2 Bucket usage metrics

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

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

📖 **NOTE**

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

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

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

Step 5 In the **Alarms** area, view the alarm severities and the alarm number of each severity. By clicking **Alarm Details**, you can explore more on the **Alarm Records** page of Cloud Eye.

📖 NOTE

Alarms is only available for buckets that support usage analysis.

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

Step 7 In the **Basic Information** area, view the bucket's basic information, including the number of objects, bucket version, enterprise project, and account ID.

Figure 4-2 Bucket's basic information

Basic Information
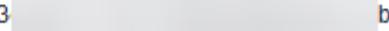
Objects	2
Bucket Version	3.0
Enterprise Project	default
Account ID	e3 b

Table 4-3 Parameters in the Basic Information area

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

Step 8 In the **Basic Configurations** area, view the basic configurations of the bucket, including logging and versioning. You can click a card to make required configurations.

Step 9 In the **FAQs** area, view bucket-related FAQs. You can click **More** in the upper right corner to view more FAQs, as shown in **Figure 4-3**.

Figure 4-3 Bucket FAQs

Step 10 In the **Learn More** area, choose to view best practices or usage guide, as shown in **Figure 4-4**.

Figure 4-4 Learn More

----End

Exporting a Bucket List

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

Step 2 Click **Export** > **Export all data to an XLSX file.** in the upper left corner of the bucket list to export all buckets.

Step 3 Select the buckets to be exported and click **Export** > **Export selected data to an XLSX file.** 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, storage class, region, used capacity, object quantity, bucket version, enterprise project, and bucket creation time.

----End

4.3 Searching for a Bucket

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

Procedure

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

Step 2 Click the search box above the bucket list, select **Bucket Name**, **Storage Class**, **Enterprise Project**, 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.

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, storage class, enterprise project, 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 3 Enter a keyword in the search box and click 🔍 or press **Enter**.

All buckets whose name, storage class, enterprise project, 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, storage class, enterprise project, or region contains keyword **test** are displayed in the bucket list.

----End

Related Operations

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

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 navigation pane of **OBS Console**, choose **Object Storage**.

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

☐ NOTE

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

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

----End

5 Managing Objects

5.1 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 puts limits on the size and number of files you can upload.

- In regions that support batch uploads, a maximum of 100 files can be uploaded at a time, with a total size of no more than 5 GB.
- In regions that do not support batch uploads, only one file can be uploaded at a time, with a size of no more than 50 MB.

Therefore, for a single file to be uploaded, its maximum size can be 5 GB in a batch upload or 50 MB in a single upload.

OBS Browser+ allows you to upload up to 500 files at a time. There is no limit on the number of files you can upload using obsutil at a time.

📖 **NOTE**

Batch upload is available only when the following condition is met:

The bucket version is 3.0. To view the bucket version, see **Viewing Basic Information of a Bucket**.

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**.

Prerequisites

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

Procedure

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

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

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

Batch upload is used as an example here. If the region you are using supports only single upload, perform operations as instructed.

☐ **NOTE**

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

Figure 5-1 Uploading objects

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

📖 **NOTE**

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

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

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

Step 6 **Server-Side Encryption**: If server-side encryption is enabled, new objects uploaded to this bucket will be automatically encrypted. For details, see **Enabling Server-Side Encryption When Uploading an Object**.

📖 **NOTE**

If your bucket has **Server-Side Encryption** configured, you can select **Inherit from bucket** when uploading an object.

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

📖 **NOTE**

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

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

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

Configuring tags: Tags are used to identify objects with tag keys and values. For details, see **Adding Tags to an Object**.

Figure 5-2 Configuring metadata or WORM retention
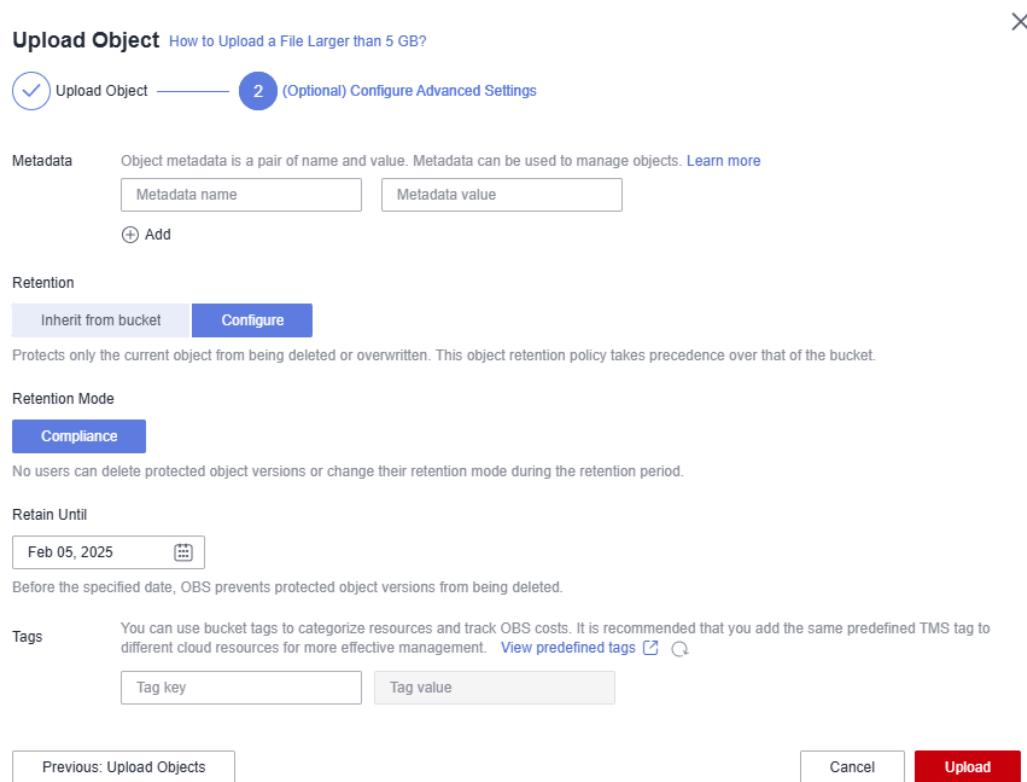
Step 8 Click **Upload**.

----End

Related Operations

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

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

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

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

📖 NOTE

You can also select multiple objects at a time and choose **More** > **Change Storage Class** above the object list, to batch change their storage classes.

Storage classes of unrestored Archive objects cannot be changed in a batch.

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

----End

☐ NOTE

- You can manually change objects between storage classes. Changing objects from Infrequent Access or Archive to other storage classes incurs restore costs. Select an appropriate change option based on your actual needs.
    - From Standard to Infrequent Access, Archive
    - From Infrequent Access to Standard, Archive
    - From Archive to Standard, Infrequent Access. Before changing Archive objects, you must restore them first.
- After an object is changed to Archive, its restore status changes to **Unrestored**.
- You can also configure a lifecycle rule to change the storage class of an object. For details, see **Configuring a Lifecycle Rule**.

Follow-up Procedure

You can click **More** > **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.

5.2 Downloading an Object

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

Constraints

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

Procedure

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

Step 2 In the bucket list, click the bucket you want to operate to go to the **Objects** page.
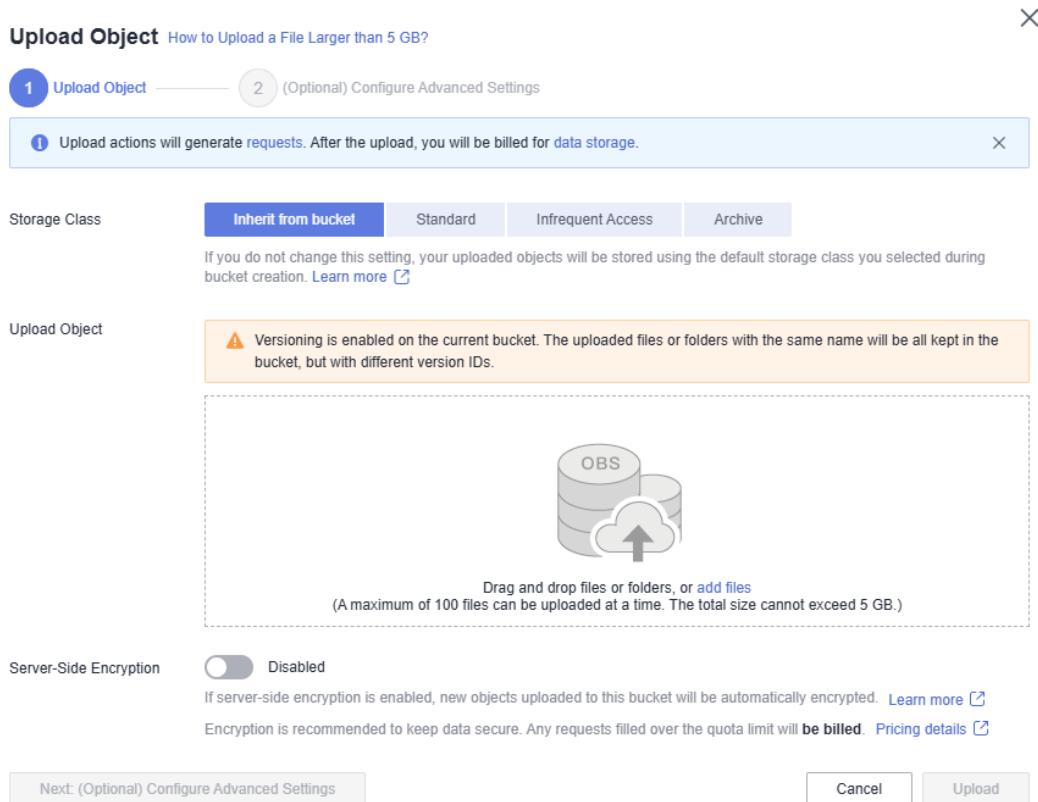
Step 3 Select the file you want to download. Then, click **Download** or **More** > **Download As** on the right.

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

☐ NOTE

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

----End

5.3 Managing Folders

5.3.1 Creating a Folder

You can create folders in a bucket on OBS Console to help organize and manage your data more efficiently.

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 navigation pane of **OBS Console**, choose **Object Storage**.

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

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

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

- You can create single-level or multi-level folders.

- The name cannot contain the following special characters: \:*?"<>|

- The name cannot start or end with a period (.) or slash (/).

- The folder's absolute path cannot exceed 1,023 characters.

- Any single slash (/) separates and creates multiple levels of folders at once.

- The name cannot contain two or more consecutive slashes (/).

Step 5 Click **OK**.

----End

Follow-up Procedure

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

5.3.2 Sharing a Folder

Scenarios

You can share your folders in OBS to other users.

Background Information

Folder sharing is temporary and has a validity period. You can temporarily share folders by access code or URL:

- By access code: Specify a six-digit access code before creating a sharing task. After the sharing task is created, OBS aggregates the download links of all objects in the folder to a static website that is hosted by a public OBS bucket. Then anyone who has the created temporary URL and access code can access the static website and download the shared files.

- By URL: Specify a validity period and then share the generated link with others. Anyone can use a signature to access all objects in the shared folder.

Constraints

- A folder shared from OBS Console can be valid for one minute to 18 hours. If you need a longer validity period, use OBS Browser+ that allows a validity period of up to one year to share the folder. If you want a shared folder permanently valid, **use a bucket policy to grant anonymous users the public read permission for the folder**.

- Folder sharing is restricted to a few regions only.

- Only version 3.0 buckets support folder sharing. You can view the bucket version in the **Basic Information** area on the **Overview** page of a bucket.

- Archive objects in a folder must be restored in the bucket before they can be downloaded.

Procedure

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

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

Step 3 Locate the folder you want to share and click **Share** in the **Operation** column. The **Share Folder** dialog box is displayed.

Step 4 Share the folder by access code or URL.

Step 5 Method 1: Share the folder by access code.

Figure 5-3 Sharing by access code
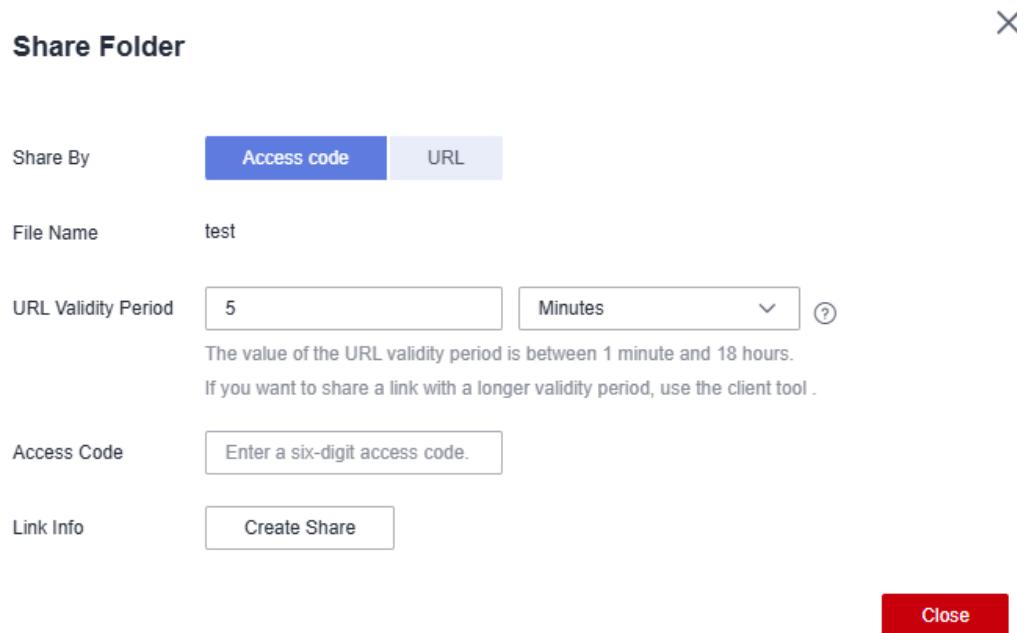
1. Choose **Access code** for **Share By**.
2. Configure parameters.

Table 5-1 Parameters for sharing a folder with an access code

Parameter	Description
URL Validity Period	A validity period is from one minute to 18 hours. The default value is five minutes. Within the URL validity period, anyone who has the URL can access the folder.
Access Code	A six-digit code. An access code is required to access objects in the shared folder.

3. Click **Create Share** to generate a sharing URL for the folder.
4. Send the URL and access code to others for them to access the folder.
5. Verification: Verify that other users can access the shared folder in a browser.

Step 6 Method 2: Share the folder by URL.

Figure 5-4 Sharing by URL
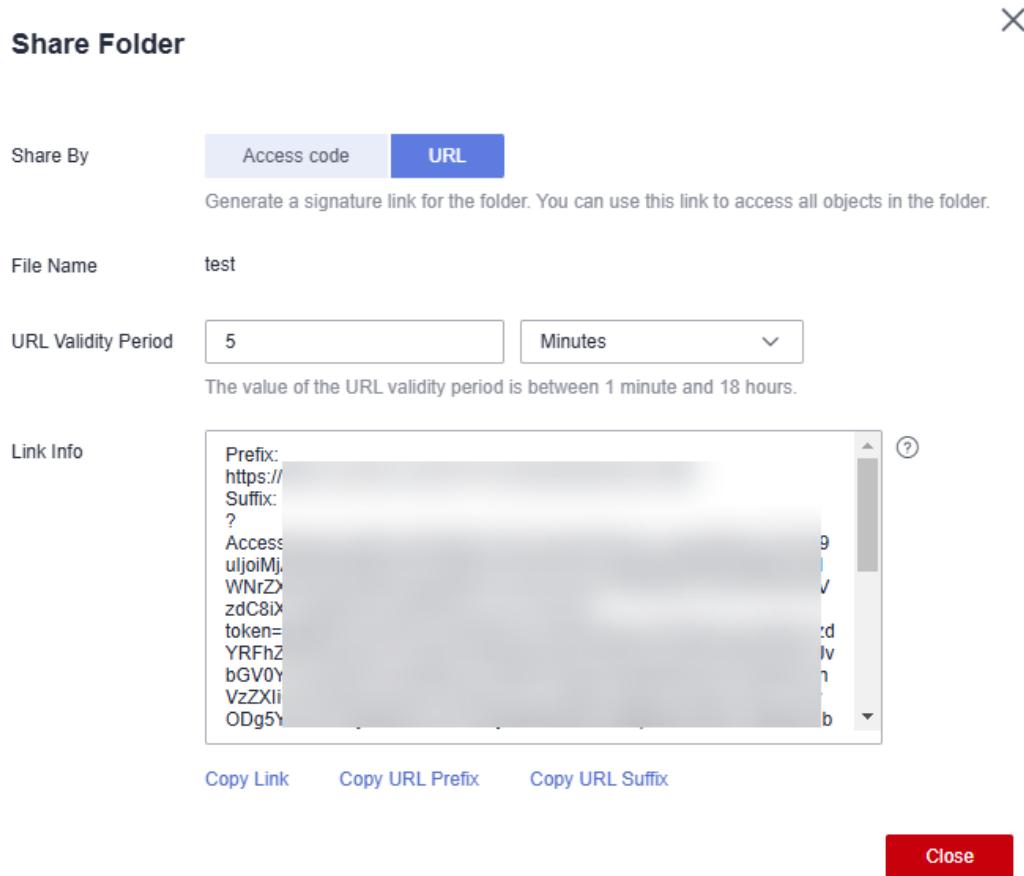
1. Choose **URL** for **Share By**.

2. Configure parameters.

    Table 5-2 Parameters for sharing a folder by URL

Parameter	Description
URL Validity Period	A validity period is from one minute to 18 hours. The default value is five minutes. Within the URL validity period, anyone who has the URL can access the folder.

3. Click **Copy Link** and share the link with another user. The user then can use this link to access all objects in this folder. The sharing link consists of the bucket domain name (prefix) and signature information (suffix). Users can add an object path after the prefix of a sharing link to access or download the specified object in a folder, as shown in **Figure 5-5**.

4. Verify that a user can use the sharing link to access all objects in the folder.

    a. Open a browser.

    b. Enter the sharing link in the address box and press **Enter** to list all objects in the folder.

    c.   Copy the object path and paste it after the prefix.

    d.   Press **Enter**. You can then access and download the specified object.

Figure 5-5 Accessing an object with a sharing link

----End

5.4 Other Object Operations

5.4.1 Listing Objects

On OBS Console, when you go to the object list page of a bucket, objects are displayed by name by default. You can also sort objects by their size or last modification time.

Listing Objects on OBS Console

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

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

Step 3 View the displayed objects. All objects in the bucket will be listed and each page has 50 objects displayed by default.

Step 4 (Optional) If versioning is enabled for a bucket, you can enable or disable **Historical Versions** above the object list to determine whether to display historical versions of objects.

- If **Historical Versions** is enabled, current and historical versions of objects (including deleted objects) are displayed.

- If **Historical Versions** is disabled, only current versions of objects are displayed.

----End

Listing Objects with OBS Tools

- Bucket-level operations on OBS Browser+ are similar to those on OBS Console. You can list objects by following the instructions on OBS Browser+. For details about OBS Browser+, see **Introduction to OBS Browser+**.

- The Java, Python, C, .NET, Node.js and Android SDKs all can be used to list objects in a bucket.
- To use the command line tool obsutil to list objects in a bucket, see **Listing Objects Using obsutil**.
- To call an API to list objects in a bucket, see **Listing Objects in a Bucket**.

Important Notes

- Listing objects by specifying a page number is not allowed.
- Objects cannot be listed by time when they were uploaded. You can search for objects by prefix only. For details, see **Searching for an Object or Folder**.
- The size and last modification time in the object list sort only objects on the current page.

5.4.2 Searching for an Object or Folder

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

Procedure

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

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

Step 3 **Search for objects by storage class**:

1. Click the search box above the object list and choose **Storage Class** from the drop-down list.
2. Select your desired option, or enter a keyword and then select the option displayed.
3. Click **OK**. The searched objects are displayed in the object list.

Suppose you want to search for objects in the Standard storage class. Click the search box and choose **Storage Class** from the drop-down list. Then, select **Standard**, or enter **standard** in the search box and select the option displayed. After that, click **OK**. Objects in the Standard storage class will be displayed in the object list.

Step 4 **Search for objects by last modification time**:

1. Click the search box above the object list and choose **Last Modified** from the drop-down list.
2. Select a start date or an end date.

   ⬜ **NOTE**

   The time can be accurate to seconds.

   Either start date or end date must be specified.

3. Click **OK**. The objects last modified within the specified time range are displayed in the object list.

Suppose you want to search for objects uploaded from 13:00:00 on October 10, 2023. Click the search box and choose **Last Modified** from the drop-down list. Then, click the text box for **Start Date**, specify the time (13:00:00 on October 10,

2023) and click **OK**. Objects uploaded from 13:00:00 on October 10, 2023 will be displayed in the object list.

Step 5 **Search for objects by object name prefix**:

1. Click the search box above the object list and choose **Object Name Prefix** from the drop-down list.

2. In the search box, enter the name prefix of the files or folders you want to search for.

   ☐ **NOTE**

   The object name prefix is case sensitive.

3. Press **Enter**. The searched objects are displayed in the object list.

Suppose you want to search for objects whose prefix is **te**. Click the search box and choose **Object Name Prefix** from the drop-down list box. Then, enter **te**, and press **Enter**. Objects with the **te** prefix will be displayed in the object list.

   ☐ **NOTE**

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

   - In the root directory, click the search box above the object list and choose **Object Name Prefix** from the drop-down list. Then, enter *Folder path*/*Prefix* in the search box. 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 6 Search for objects by entering a storage class keyword or an object name prefix in the search box above the object list.

- Enter a storage class keyword and click a required criterion from those that are automatically displayed. All matched objects will be displayed in the object list.

  Suppose you enter **stan** in the search box. The system will then display the **Standard** storage class. After you click this storage class, all objects whose storage class is Standard will be displayed in the object list.

- Enter an object name prefix and press **Enter**. All files and folders with the specified prefix will be displayed in the object list.

  Suppose you want to search for objects with the **te** prefix. Enter **te** in the search box and press **Enter**. Then, all objects with the **te** prefix will be displayed in the object list.

  ☐ **NOTE**

  You can search for objects based on a combination of different filter criteria.

  - If the filter criteria are of different types, they are in intersection logic. For example, if you select storage class **Standard** and object name prefix **te** as two criteria, objects whose storage class is Standard and prefix is **te** will be displayed in the object list.
  - If the filter criteria are of the same type, they are in union logic. For example, if you select storage class **Standard** and then **Infrequent Access** as two criteria, objects in both Standard and Infrequent Access storage classes will be displayed in the object list.

----End

Related Operations

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

5.4.3 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 All Accounts the Read Permission for Certain Objects**.

☐ NOTE

Encrypted objects cannot be shared.

Procedure

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

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

Step 3 Click the object to be shared. On the top part of the page, you can find the object's sharing link in the **Link** area, as shown in **Figure 5-6**.

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. To learn more about domain names, see **OBS Domain Names**.

Figure 5-6 Object link

Name	object_002.PNG	Storage Class	Standard Change Storage Class
Last Modified	Jun 07, 2022 09:50:12 GMT+08:00	Size	37.51 KB
Link		Version ID	--
Encrypted	No		

☐ NOTE

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

----End

5.4.4 Sharing an Object

Scenarios

You can share temporary URLs of your objects with others for them to access your objects stored in OBS.

Background Information

File sharing is temporary. All sharing URLs are only valid for a limited period of time.

A temporary URL consists of the access domain name and the temporary authentication information of a file. Example:

```
https://bucketname.obs.eu-west-101.myhuaweicloud.eu:443/image.png?
AccessKeyId=xxx&Expires=xxx&response-content-disposition=xxx&x-obs-security-token=xxx&Signature=xxx
```

The temporary authentication information contains the **AccessKeyId**, **Expires**, **x-obs-security-token**, and **Signature** parameters. **AccessKeyId**, **x-obs-security-token**, and **Signature** are used for authentication. The **Expires** parameter specifies the validity period of the authentication. For more information about temporary authentication methods and parameters, see **Authentication of Signature in a URL** in *Object Storage Service API Reference*.

After an object is shared on OBS Console, the system will generate a URL that contains the temporary authentication information, valid for five minutes since its generation by default. Each time you change the validity period of a URL, OBS obtains the authentication information again to generate a new URL for sharing, which takes effect since the time when the validity period is changed.

Constraints

- An object shared from OBS Console can be valid for one minute to 18 hours. If you need a longer validity period, use OBS Browser+ that allows a validity period of up to one year to share the object. If you want a shared object to be permanently valid, **use a bucket policy to grant anonymous users the public read permission for the object**.
- Only version 3.0 buckets support file sharing. You can view the bucket version in the **Basic Information** area on the **Overview** page of a bucket.
- Archive objects can be shared only after they have been restored.

Procedure

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

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

Step 3 Locate the file to be shared and click **Share** in the **Operation** column.

Once the **Share File** dialog box is opened, the URL is effective and valid for five minutes by default. If you change the validity period, the authentication information in the URL changes accordingly, and the URL's new validity period starts upon the change.

Figure 5-7 Sharing a file
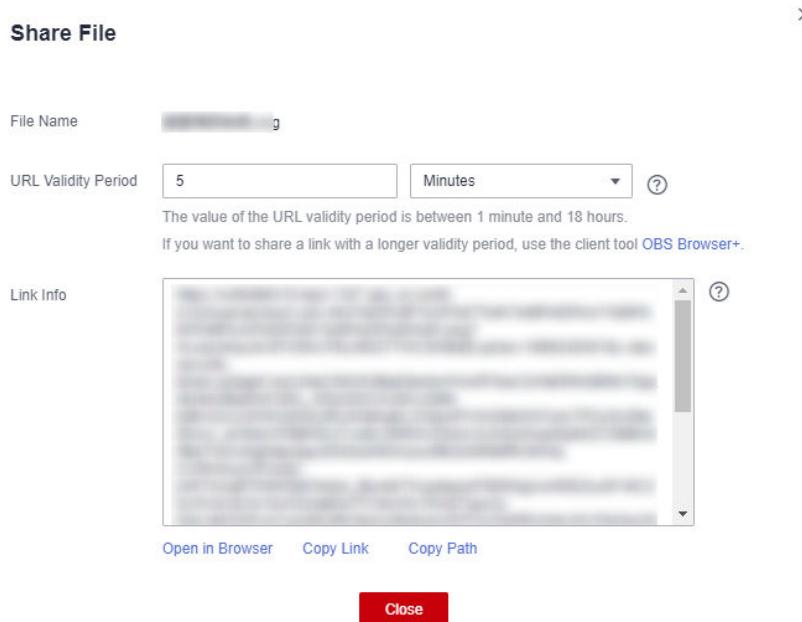
Step 4 Operate the URL as follows:

- Click **Open URL** to preview the file on a new page or directly download it to your default download path.

- Click **Copy Link** to share the link to others for them to access this file using a browser.

- Click **Copy Path** to share the file path to users who have access to the bucket. The users then can search for the file by pasting the shared path to the search box of the bucket.

📖 **NOTE**

Within the URL validity period, anyone who has the URL can access the file.

----End

5.4.5 Restoring an Object from Archive Storage

You must restore an object in the Archive storage class before you can download it or access it with a URL.

Constraints

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

- An object being restored cannot be restored again.

- After an object is restored, an object copy in the Standard storage class will be generated. This way, there is an Archive object and a Standard object copy in the bucket at the same time. During the restore validity period, you will be charged for the space taken up by both the object and its copy. The copy will be automatically deleted once the restoration expires.

Procedure

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

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

Step 3   Select the file you want to restore, and click **Restore** on the right. The following dialog box shown in **Figure 5-8** is displayed.

You can select multiple files and choose **More** > **Restore** above the file list to batch restore them.

📖 **NOTE**

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

Figure 5-8 Restoring an object
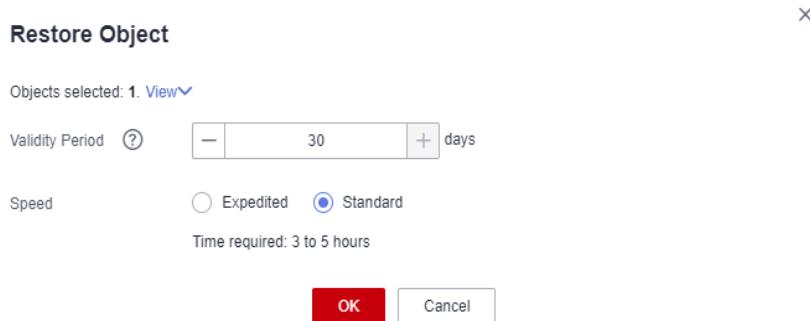
Step 4   Configure the validity period and speed of the restore. The following table describes the parameters.

Table 5-3 Parameters for restoring objects

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

Step 5   Click **OK**.

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

You can click ↻ to manually refresh the restore status.

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

----End

Related Operations

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

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

5.4.6 Configuring Direct Reading

With direct reading enabled for a bucket, you can access objects in the Archive storage class without restoring them first. Downloading or copying Archive objects will incur costs for directly reading these objects. For details, see **Product Pricing Details**.

You can enable direct reading for a bucket during its creation. For details, see **Creating a Bucket**. Alternatively, you can enable direct reading for an existing bucket by doing as follows:

Procedure

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

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

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

Step 4 In the **Basic Configurations** area, click **Direct Reading**. The **Direct Reading** dialog box is displayed.

Step 5 Select **Enable**.

Figure 5-9 Enabling direct reading
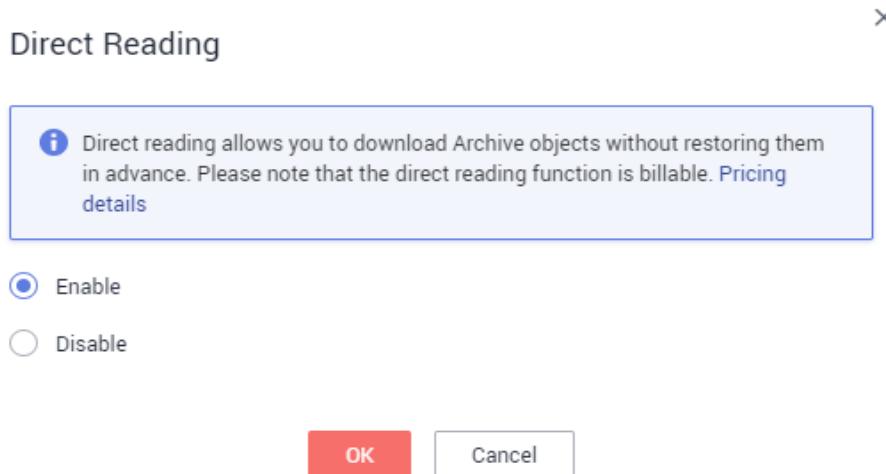
Step 6 Click **OK**.

----End

5.4.7 Configuring Object Metadata

Procedure

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

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

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

Step 4 Click **Add** and specify the metadata information, as shown in **Figure 5-10**.

Figure 5-10 Adding metadata

Step 5 Click **OK**.

----End

5.4.8 Adding Tags to an Object

Tag Rules

Tags can be added to objects to categorize storage. You can add tags to new objects when you upload them or add them to existing objects.

An object tag is a key-value pair. To use tags, follow the rules below:

- An object can have up to 10 tags.
- Tag key constraints:
  - If there are multiple tags specified for an object, each tag key must be unique.
  - A tag key can be 36 characters long at most and cannot contain the following special characters: **= * < > \ , | /** It must be case sensitive and unique, and cannot start or end with a space. It cannot be left blank.
- Tag value constraints:
  - A tag value can be 43 characters long at most and cannot contain the following special characters: **= * < > \ , | /** It must be case sensitive and cannot start or end with a space. It can be left blank.

Important Notes

- Object tags are now during open beta testing and can be used for free. Once the testing period is over, you will be billed for using them.
- To read or write object tags, you must have required permissions. Such permissions can be granted using bucket policies. By default, only the object owner can read or write the object's tags.
- Tags cannot be added to files in parallel file systems.
- Currently, object tags cannot be used in IAM permissions or bucket policies.

Scenarios

Specifying object tags in a lifecycle rule

In a lifecycle configuration, you can specify filter conditions to identify the objects which the lifecycle rule applies to. A filter can be defined based on an object name prefix, an object tag, or both. The following rule deletes objects with the **texta/** prefix and with tags **key1/value1** and **key2/value2** 120 days after their creation.

```
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Filter>
      <And>
        <Prefix>texta/</Prefix>
        <Tag>
          <Key>key1</Key>
          <Value>value1</Value>
        </Tag>
        <Tag>
          <Key>key2</Key>
```

```
                <Value>value2</Value>
              </Tag>
            </And>
          </Filter>
          <Status>Enabled</Status>
          <Expiration>
              <Days>120</Days>
          </Expiration>
        </Rule>
</LifecycleConfiguration>
```

Ways to Configure Object Tags

You can use OBS Console or APIs to configure object tags.

Using OBS Console

On OBS Console, you can add tags to an object when uploading it (see **Procedure**). You can also add tags to an existing object, as described in the following:

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

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

Step 3 In the object list, click the name of the object you want to add tags to.

Step 4 Click the **Tags** tab. Then, click **Add Tag** in the upper left corner.

Step 5 In the **Add Tag** dialog box, enter a tag key and a tag value.

Table 5-4 Parameter description

Parameter	Description
Tag key	The key of an object must be unique and cannot be left blank. You can customize a tag or choose one predefined on TMS. A tag key: ● If there are multiple tags specified for an object, each tag key must be unique. ● A tag key can be 36 characters long at most and cannot contain the following special characters: = * < > \ , | / It must be case sensitive and unique, and cannot start or end with a space. It cannot be left blank.
Tag value	A tag value can be repetitive or left blank. A tag value: ● A tag value can be 43 characters long at most and cannot contain the following special characters: = * < > \ , | / It must be case sensitive and cannot start or end with a space. It can be left blank.

Step 6 Click **OK**.

----End

Using APIs

Adding Tags to an Object

Obtaining Object Tags

Deleting Object Tags

Using SDKs

Java: not suppo rted	Pytho n: not suppo rted	C: not suppo rted	Go: not suppo rted	Brows erJS: not suppo rted	.NET: not suppo rted	Andro id: not suppo rted	iOS: not suppo rted	PHP: not suppo rted	Node. js: not suppo rted

Related Operations

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

5.5 Deleting Objects

5.5.1 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.

Deleting Objects from a Bucket with Versioning Disabled, Suspended, or Enabled

Deleting objects may cause data loss. Exercise caution when performing this operation. Versioning can reduce the risk by restoring objects that are accidentally deleted or overwritten.

Scenario	Deleting the Current Object Version	Deleting a Noncurrent Object Version
A versioning-disabled bucket	OBS permanently deletes objects. Deleted objects cannot be recovered.	N/A
A versioning-suspended bucket	- If an object's current version does not have a delete marker and its version ID is not null, the object deletion operation turns the current version into a noncurrent version and inserts a delete marker with an ID of null. The delete marker then becomes the current version. - If an object's current version does not have a delete marker and its version ID is null, the system creates a delete marker with an ID of null and overwrites the current version. The delete marker then becomes the current version, and the overwritten version cannot be recovered. - If the current version has a delete marker, no action is taken. **NOTE** - After versioning is suspended, the version ID of the newly uploaded object is null. If an object with the same name and an empty version ID exists, the object will be overwritten by the uploaded object. For more information about versioning, see **Versioning**. - By enabling **Historical Versions** in the object list, you can view the version ID of an object.	Deleted noncurrent versions cannot be recovered.

Scenario	Deleting the Current Object Version	Deleting a Noncurrent Object Version

A versioning-enabled bucket	● If an object's current version does not have a delete marker, the system creates a delete marker. Then, the current version becomes a noncurrent version and the delete marker becomes the current version.
	
	● If the current version has a delete marker, no action is taken.
	
	  NOTE
	    ● To permanently delete the object, delete it again from the **Deleted Objects** list.
	    ● To recover the deleted object, undelete it from the **Deleted Objects** list. For details, see **Undeleting an Object**.
		Deleted noncurrent versions cannot be recovered.

Procedure

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

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

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

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

Step 4 Click **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, complete the following steps:

Step 5 Permanently delete a file or a folder from a versioning-enabled OBS bucket:

1. Enable **Historical Versions**.
2. Select all the versions of the object to be permanently deleted and click **Permanently Delete** above the search bar.

Figure 5-11 Permanently deleting an object
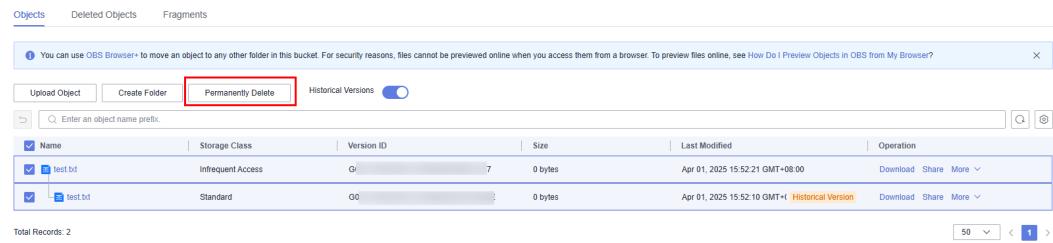
3. Click **OK**.

----End

5.5.2 Undeleting an Object

Scenarios

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

Important Notes

- Files can be undeleted, but folders cannot.

  After you undelete a deleted file, the file is recovered and will appear in the **Objects** list. Then you can perform basic operations on the file as you normally do on other objects. If the file was stored in a folder before the deletion, it will be recovered to its original path after you undelete it.

- 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 file is recovered later.

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 navigation pane of **OBS Console**, choose **Object Storage**.

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

Step 3 Click **Deleted Objects**.

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

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

Figure 5-12 Undeleting a file
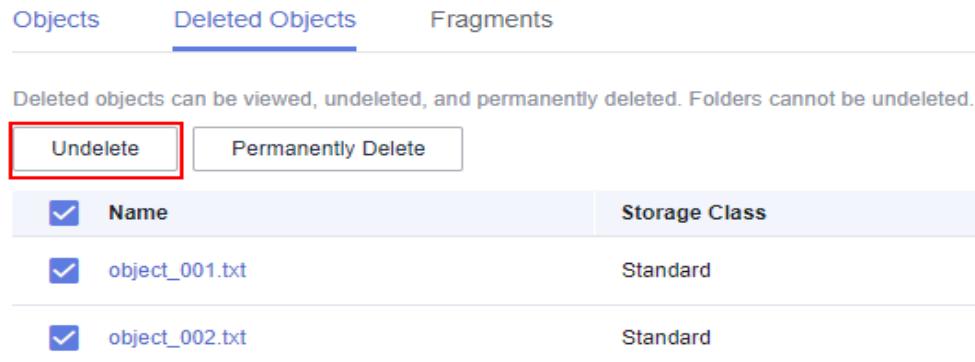
----End

Related Operations

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

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

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

Step 3   Enable **Historical Versions**.

Step 4   Locate the file to be retrieved. All its noncurrent versions will be displayed.

- 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

5.5.3 Managing Fragments

Background Information

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

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

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

> **NOTICE**
>
> Generated fragments take up storage space that is billable.

Procedure

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

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

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

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

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

----End

6 Permissions Control

6.1 Configuring IAM Permissions

6.1.1 Creating an IAM User and Granting OBS Permissions

You can use **IAM** for fine-grained access control over your OBS resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing OBS resources.

- Manage permissions on a principle of least permissions (PoLP) basis.

- Entrust a Huawei Cloud account or cloud service to perform efficient O&M on your OBS resources.

If your Huawei Cloud account does not require individual IAM users, skip this chapter.

Figure 6-1 shows the procedure for granting permissions.

Prerequisites

You have learned about the **OBS permissions** that can be assigned to a user group.

Process

Figure 6-1 Process of granting an IAM user the OBS permissions

The below example describes how to grant an IAM user the **Tenant Guest** permission for OBS.

1. **Create a user group and assign permissions**.

   Create a user group on the IAM console, and assign the group the **Tenant Guest** permission.

2. **Create an IAM user and add it to the user group**.

   Create a user on the IAM console and add the user to the group created in **1**.

3. **Log in** and verify the permission granting.

   Log in to OBS Console using the newly created user, and verify that the assigned permission has taken effect:

   – Choose **Object Storage Service** from the service list to go to the OBS homepage. If the list of buckets is displayed and you can view the basic information about any bucket, but you cannot create or delete buckets or perform any other operations, the granted **Tenant Guest** permission has already taken effect.

   – Go to an OBS bucket. If the list of objects is displayed and you can download objects, but you cannot upload or delete objects or perform any other operations, the **Tenant Guest** permission granted has already taken effect.

6.1.2 OBS Custom Policies

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

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

- Visual editor: Select cloud services, actions, resources, and request conditions without the need to know policy syntax.

- JSON: Edit JSON policies from scratch or based on an existing policy.

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

Example Custom Policies

- Example 1: Grant users all OBS permissions.

  This policy allows users to perform any operation on OBS.

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

- Example 2: Grant users all OBS Console permissions.

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

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

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

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

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

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

```
    ]
}
```

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

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

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

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

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

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

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

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

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

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

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

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

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

6.1.3 OBS Resources

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

Table 6-1 OBS resources and their paths

Resource Type	Resource Name	Path
Buckets	Bucket	[Format] **obs:*:*:bucket:**_Bucket name_ [Notes] IAM automatically generates the prefix **obs:*:*:bucket:** for bucket resource paths. By adding _Bucket name_ to the end of the generated prefix, you can define a specific path. An asterisk ***** is allowed to indicate any bucket. An example is given as follows: **obs:*:*:bucket:***

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

6.1.4 OBS Request Conditions

Request conditions are useful in determining when a custom policy is in effect. A request condition consists of a condition key and an operator. Condition keys are either global or service-level and are used in the condition elements of a policy statement. **Global condition keys** (starting with **g:**) are available for operations of all services, while service-level condition keys (starting with a service name acronym like **obs:**) are available only for operations of a specific service. An operator is used together with a condition key to form a complete condition statement.

OBS has a group of predefined condition keys that can be used in IAM. For example, to define an allow permission, you can use the condition key **obs:SourceIp** to filter matching requesters by IP address.

The condition keys and operators supported by OBS are the same as those in the bucket policy. When configuring condition keys in IAM, start them with **obs:**. For details, see **Policy Format**.

6.2 Configuring a Bucket Policy

6.2.1 Creating a Bucket Policy with a Template

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

Procedure

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

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

Step 3 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 4 Click **Create Bucket Policy**.

Step 5 Choose a policy template. For details about the parameters, see **Bucket Policies**.

Figure 6-2 Choosing the Public Read template
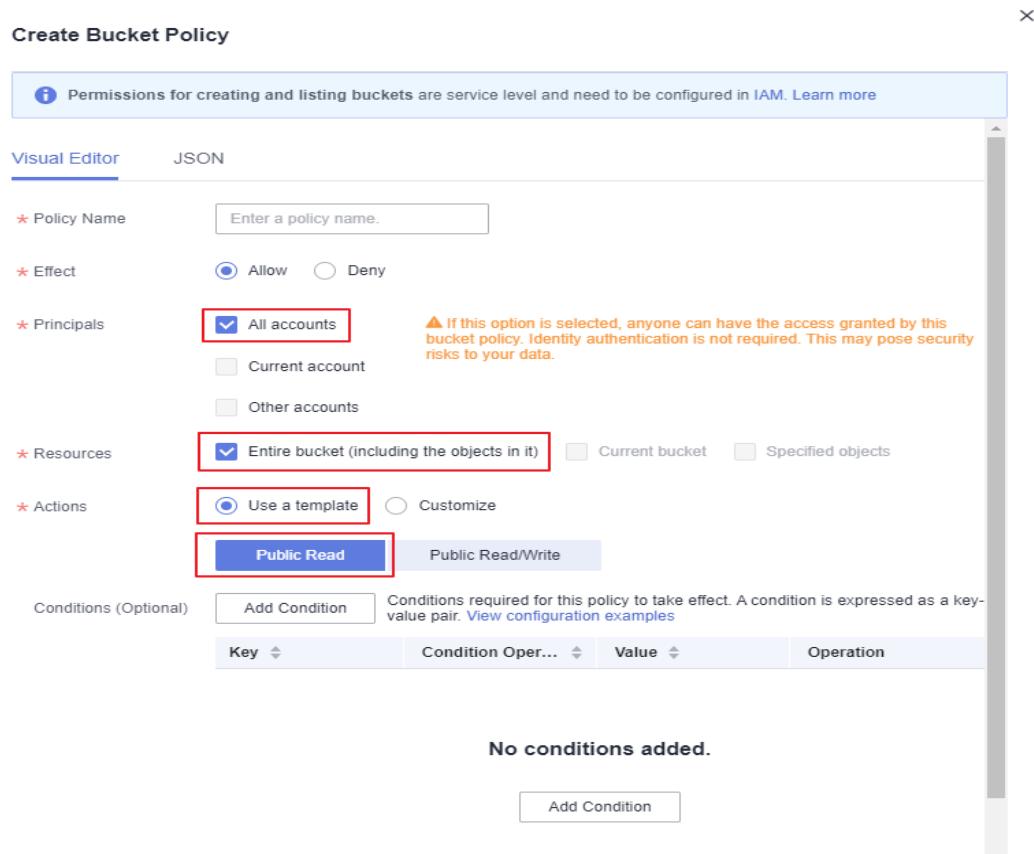
Figure 6-3 Choosing the Public Read/Write template
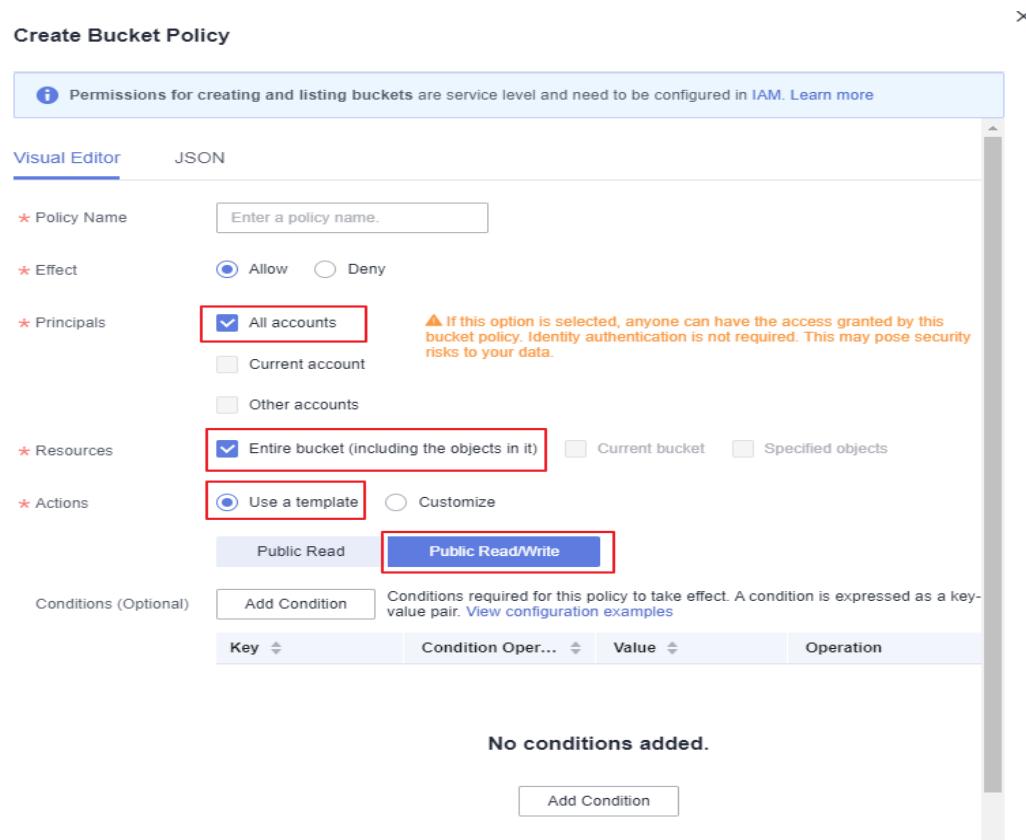
Figure 6-4 Choosing the Bucket Read-Only template
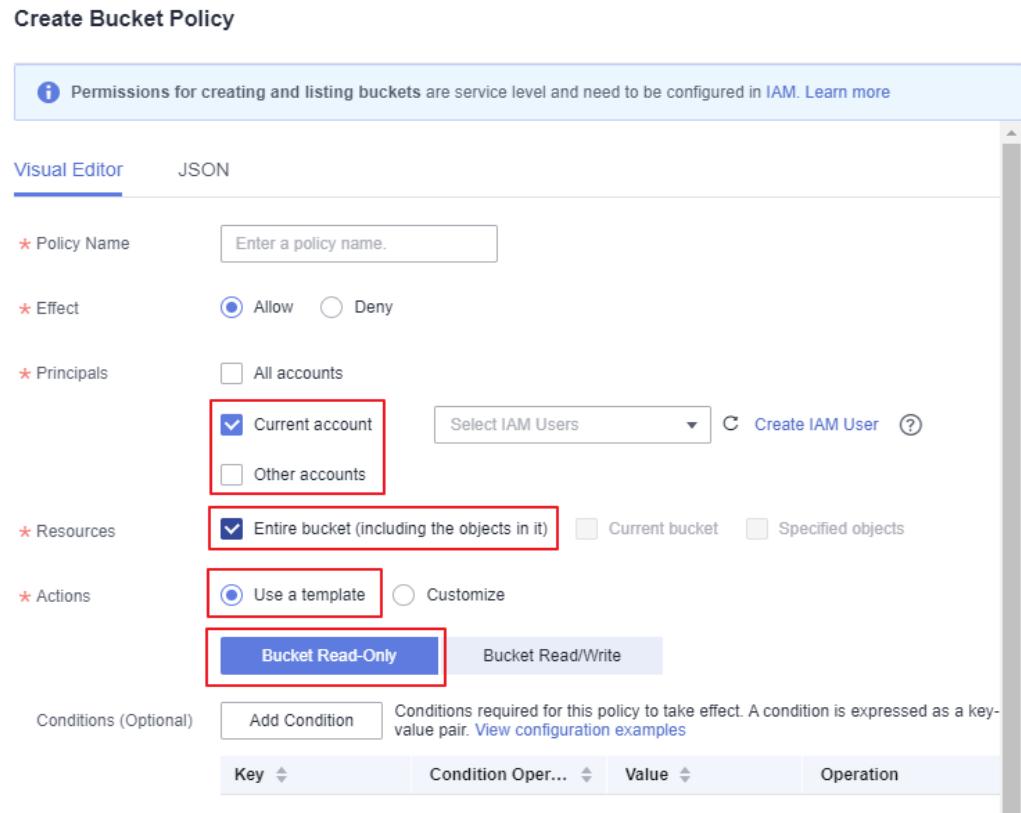
Figure 6-5 Choosing the Bucket Read/Write template

Figure 6-6 Choosing the Directory Read-Only template
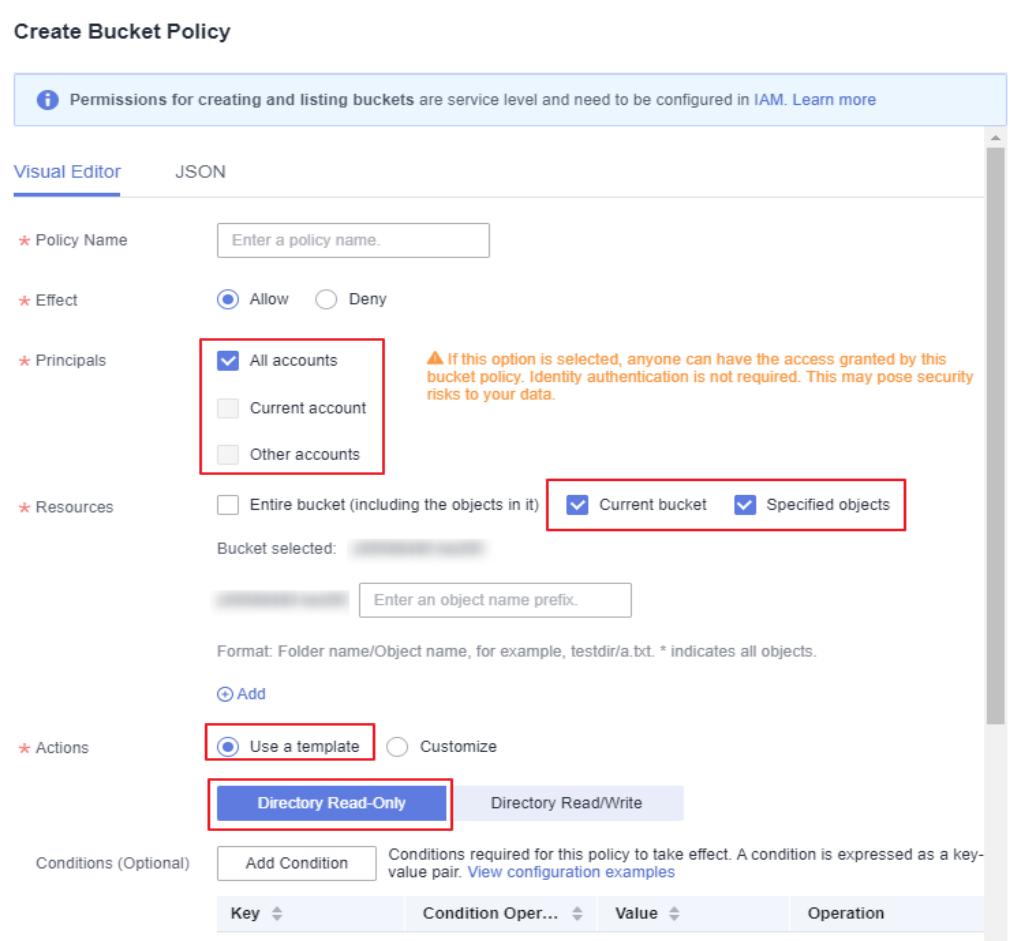
Figure 6-7 Choosing the Directory Read/Write template

Create Bucket Policy

Permissions for creating and listing buckets are service level and need to be configured in IAM. Learn more

Visual Editor   JSON

Policy Name   Enter a policy name.

Effect   ◉ Allow   ○ Deny

Principals   ☑ All accounts   ⚠ If this option is selected, anyone can have the access granted by this bucket policy. Identity authentication is not required. This may pose security risks to your data.
☐ Current account
☐ Other accounts

Resources   ☐ Entire bucket (including the objects in it)   ☑ Current bucket   ☑ Specified objects
Bucket selected:

Enter an object name prefix.

Format: Folder name/Object name, for example, testdir/a.txt. * indicates all objects.

⊕ Add

Actions   ◉ Use a template   ○ Customize

Directory Read-Only   **Directory Read/Write**

Conditions (Optional)   Add Condition   Conditions required for this policy to take effect. A condition is expressed as a key-value pair. View configuration examples

Key ⇕	Condition Oper... ⇕	Value ⇕	Operation

Figure 6-8 Choosing the Object Read-Only template
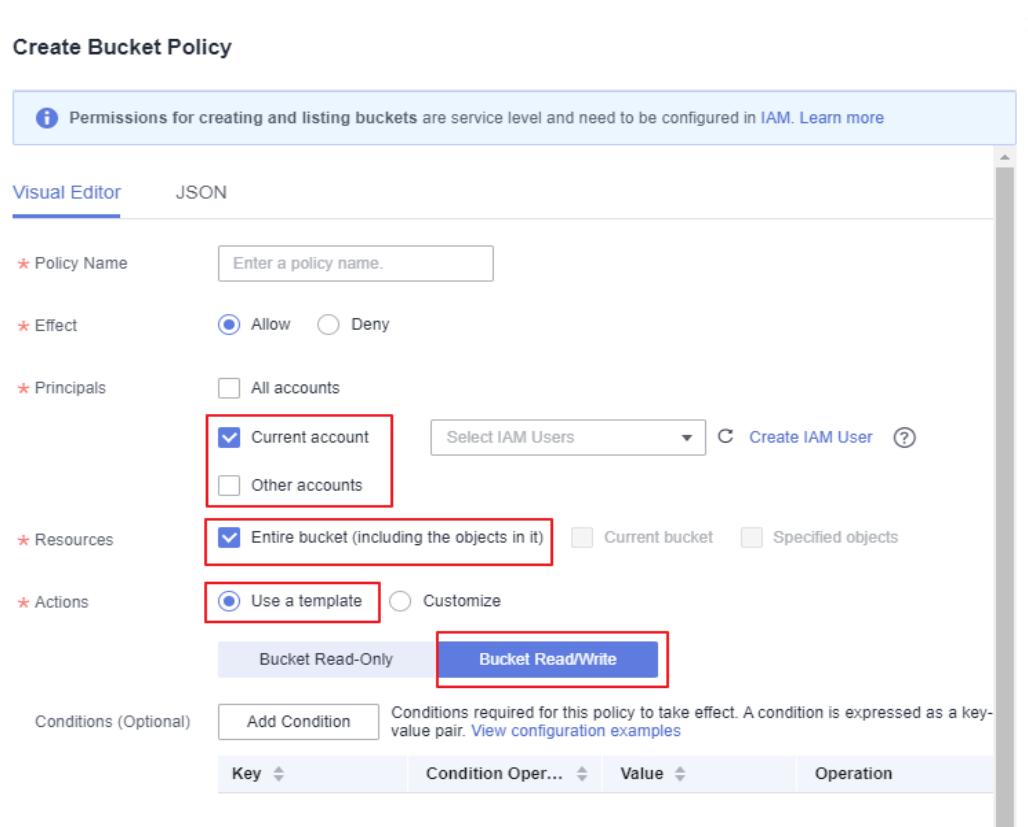
Figure 6-9 Choosing the Object Read/Write template

Create Bucket Policy

> Permissions for creating and listing buckets are service level and need to be configured in IAM. Learn more

Visual Editor   JSON

* Policy Name   [Enter a policy name.]

* Effect   ⦿ Allow   ○ Deny

* Principals   ☑ All accounts   ⚠ If this option is selected, anyone can have the access granted by this
                     ☐ Current account   bucket policy. Identity authentication is not required. This may pose security
                     ☐ Other accounts   risks to your data.

* Resources   ☐ Entire bucket (including the objects in it)   ☐ Current bucket   ☑ Specified objects

                     [   ] [Enter an object name prefix.]

                     Format: Folder name/Object name, for example, testdir/a.txt. * indicates all objects.

                     ⊕ Add

* Actions   ⦿ Use a template   ○ Customize

                     Object Read-Only   **Object Read/Write**

Conditions (Optional)   [Add Condition]   Conditions required for this policy to take effect. A condition is expressed as a key-
                                            value pair. View configuration examples

                     Key ⇕   Condition Oper... ⇕   Value ⇕   Operation

Table 6-2 Bucket policy templates

Principal	Resource	Template	Actions Allowed	Advanced Settings
All accounts	Entire bucket (including the objects in it)	Public Read See **Figure 6-2**.	**Allows anonymous users to perform the following actions on a bucket and the objects in it:** HeadBucket (to check whether the bucket exists and obtain the bucket metadata) GetBucketLocation (to get the bucket location) GetObject (to obtain object content and metadata) RestoreObject (to restore objects from Archive storage) GetObjectVersion (to obtain the content and metadata of a specified object version)	Excluding the specified actions is not allowed.

Pri nci pal	Resource	Templa te	Actions Allowed	Advanced Settings
		Public Read/ Write See **Figure 6-3**.	**Allows anonymous users to perform the following actions on a bucket and the objects in it:**	Excluding the specified actions is not allowed.
			ListBucket (to list objects in the bucket and obtain the bucket metadata)	
			ListBucketVersions (to list object versions in the bucket)	
			HeadBucket (to check whether the bucket exists and obtain the bucket metadata)	
			GetBucketLocation (to get the bucket location)	
			PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts)	
			GetObject (to obtain object content and metadata)	
			ModifyObjectMetaData (to modify object metadata)	
			ListBucketMultipartUploads (to list multipart uploads)	
			ListMultipartUploadParts (to list uploaded parts)	
			AbortMultipartUpload (to abort multipart uploads)	
			RestoreObject (to restore objects from Archive storage)	
			GetObjectVersion (to obtain the content and metadata of a specified object version)	
			PutObjectAcl (to configure the object ACL)	
			GetObjectVersionAcl (to obtain the ACL of a specified object version)	
			GetObjectAcl (to obtain the object ACL)	

Principal	Resource	Template	Actions Allowed	Advanced Settings
Current account/ Other accounts/ Delegated accounts	Entire bucket (including the objects in it)	Bucket Read-Only See **Figure 6-4**.	**Allows specified accounts to perform the following actions on a bucket and the objects in it:** Get* (all GET actions) List* (all LIST actions) HeadBucket (to check whether the bucket exists and obtain the bucket metadata)	Excluding the specified actions is not allowed.
		Bucket Read/ Write See **Figure 6-5**.	**Allows specified accounts to perform all actions excluding the following ones on a bucket and the objects in it:** DeleteBucket (to delete the bucket) PutBucketPolicy (to configure a bucket policy) PutBucketAcl (to configure the bucket ACL)	The specified actions are excluded.

Principal	Resource	Template	Actions Allowed	Advanced Settings
All acc oun ts/ Cur ren t acc oun t/ Oth er acc oun ts/ Del ega ted acc oun ts	Current bucket + Specified objects	Director y Read-Only See **Figure 6-6**.	**Allows all accounts or specified accounts to perform the following actions on the current bucket and the specified resources in it:** GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL) RestoreObject (to restore objects from Archive storage) ListBucket (to list objects in the bucket and obtain the bucket metadata) ListBucketVersions (to list object versions in the bucket) HeadBucket (to check whether the bucket exists and obtain the bucket metadata) GetBucketLocation (to get the bucket location) **NOTE** If you apply the policy to **All accounts**, **ListBucket** and **ListBucketVersions** are not included in the template.	Excluding the specified actions is not allowed.

Principal	Resource	Template	Actions Allowed	Advanced Settings
		Directory Read/Write See **Figure 6-7**.	**Allows all accounts or specified accounts to perform the following actions on the current bucket and the specified resources in it:** PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts) GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) ModifyObjectMetaData (to modify object metadata) ListBucketMultipartUploads (to list multipart uploads) ListMultipartUploadParts (to list uploaded parts) AbortMultipartUpload (to abort multipart uploads) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL) PutObjectAcl (to configure the object ACL) RestoreObject (to restore objects from Archive storage) ListBucket (to list objects in the bucket and obtain the bucket metadata) ListBucketVersions (to list object versions in the bucket) HeadBucket (to check whether the bucket exists and obtain the bucket metadata) GetBucketLocation (to get the bucket location)	Excluding the specified actions is not allowed.

Principal	Resource	Template	Actions Allowed	Advanced Settings
All accounts/ Current account/ Other accounts/ Delegated accounts	Specified objects	Object Read-Only See **Figure 6-8**.	**Allows all accounts or specified accounts to perform the following actions on specified resources in the bucket:** GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) GetObjectVersionAcl (to obtain the ACL of a specified object version) GetObjectAcl (to obtain the object ACL) RestoreObject (to restore objects from Archive storage)	Excluding the specified actions is not allowed.
		Object Read/ Write See **Figure 6-9**.	**Allows all accounts or specified accounts to perform the following actions on specified resources in the bucket:** PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts) GetObject (to obtain object content and metadata) GetObjectVersion (to obtain the content and metadata of a specified object version) ModifyObjectMetaData (to modify object metadata) ListMultipartUploadParts (to list uploaded parts) AbortMultipartUpload (to abort multipart uploads) GetObjectVersionAcl (to obtain the ACL of an object version) GetObjectAcl (to obtain the object ACL) PutObjectAcl (to configure the object ACL) RestoreObject (to restore objects from Archive storage)	Excluding the specified actions is not allowed.

Step 6 Complete the bucket policy configuration.

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

Step 7 Click **Create** in the lower right corner.

----End

6.2.2 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. For details, see **Bucket Policy Parameters**.

Procedure

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

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

Step 3 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 4 Click **Create Bucket Policy**.

Step 5 Configure a bucket policy.

Table 6-3 Parameters for configuring a custom bucket policy

Parameter	Description
Policy View	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.
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. The format is *Account ID/IAM User ID*. To specify multiple accounts, enter each one on a separate line. *Account ID*/***** indicates that permission is granted to all users under the account. ● **Delegated accounts**: Delegated accounts can be added only after **Other accounts** is selected. **NOTE** The format is *Account ID/Agency Name*. To specify multiple agencies, enter 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** – Multiple resource paths can be specified. – 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/***. – You can specify a specific object, an object set, or a directory. ***** indicates all objects in the bucket. To specify a specific object, enter the object name. To specify a set of objects, enter *Object name prefix******, ******Object name suffix*, or *****. For example, **testdir/*** indicates objects in the **testdir** folder, and **testprefix*** indicates objects whose prefix is **testprefix**.

Parameter	Description

Actions	● **Actions**: Choose **Customize**.
	● **Select Actions**: See **Bucket Policy Parameters**.
	NOTE
	  1. If you select **Entire bucket (including the objects in it)** for **Resources**, common actions, bucket actions, and object actions will be available for you to choose from.
	  2. If you select **Current bucket** for **Resources**, common actions and bucket actions will be available for you to choose from.
	  3. If you select **Specified objects** for **Resources**, common actions and object actions will be available for you to choose from.
	  4. If you select both **Current bucket** and **Specified objects** for **Resources**, common actions, bucket actions, and object actions will be available for you to choose from.

Conditions (Optional)	● **Key**: See **Bucket Policy Parameters**. ● **Conditional Operator**: See **Bucket Policy Parameters**. ● **Value**: The entered value is associated with the key.

Advanced Settings > Exclude (Optional)	● **Specified principals**: By selecting this option, the bucket policy applies to users except the specified ones.
	NOTE
	  – **Exclude** not selected: The bucket policy applies to the specified users.
	  – **Exclude** selected: The bucket policy applies to users except the specified ones.
	● **Specified resources**: By selecting this option, the bucket policy applies to resources except the specified ones.
	NOTE
	  – **Exclude** not selected: The bucket policy applies to the specified OBS resources.
	  – **Exclude** selected: The bucket policy applies to OBS resources except the specified ones.
	● **Specified actions**: By selecting this option, the bucket policy applies to actions except the specified ones.
	NOTE
	  – **Exclude** not selected: The bucket policy applies to the specified actions.
	  – **Exclude** selected: The bucket policy applies to actions except the specified ones.
	  – By default, **Specified actions** is selected for **Exclude** in the bucket read/write template only. The action exclusion setting in bucket policy templates cannot be modified.

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

----End

6.2.3 Creating a Custom Bucket Policy (JSON View)

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

Procedure

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

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

Step 3 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 4 In the upper right corner of the page, click **JSON** and then **Edit**.

Step 5 Edit the bucket policy. Below gives a bucket policy example in JSON:

```
{
  "Statement":[
    {
      "Sid":"test-obs-policy",
      "Effect":"Allow",
      "Principal":{
        "ID":[
          "domain/account ID",
          "domain/account ID.user/User ID"
        ]
      },
      "Action":[
        "PutObject",
        "GetObject"
      ],
      "Resource":[
        "test-bucket/test.txt"
      ],
      "Condition":{
        "StringNotEquals":{
          "Referer":[
            "sdf"
          ]
        },
        "IpAddress":{
          "SourceIp": [
            "114.115.1.0/24"
          ]
        }
      }
    }
  ]
}
```

Table 6-4 Parameters for creating a bucket policy in JSON

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

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/**_Domain ID_ (indicating that the principal is an account) ● **domain/**_Domain ID_**:user/**_User ID_ (indicating that the principal is a user under an account)
Condition	Conditions under which the bucket policy takes effect. For details, see **Bucket Policy Parameters**.
Resource	Resources the bucket policy is applied to. For details, see **Bucket Policy Parameters**.

Step 6 Click **Save**.

----End

6.3 Configuring an Object Policy

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

Procedure

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

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

Step 3 In the row containing the object for which you want to configure a policy, choose **More** > **Configure Object Policy** in the **Operation** column. The **Configure Object Policy** page is displayed.

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

● **Using a preset template**: The system presets object policy templates for multiple typical scenarios. You can use the templates to quickly configure object policies. For details about each template, see **Bucket Policy Parameters**.

● **Customizing a policy**: You can also customize an object policy based on your needs. A custom object policy consists of five basic elements: 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

6.4 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 navigation pane of **OBS Console**, choose **Object Storage**.

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

Step 3 In the navigation pane, choose **Permissions** > **Bucket ACL**.

Step 4 On the **Bucket ACL** page, choose **Private**, **Public Read**, or **Public Read/Write** to grant the bucket ACL permission for anonymous users.

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

Figure 6-10 Changing a public access permission
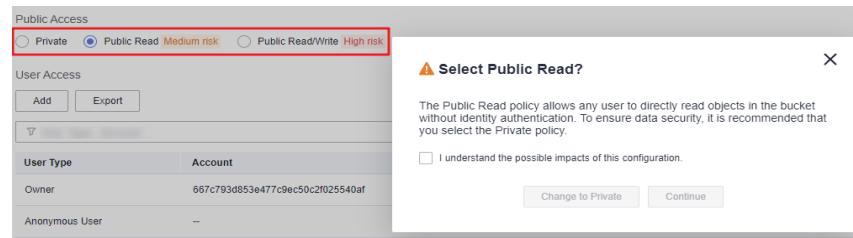
Step 5 In the **Operation** column, click **Edit** to grant the owner, anonymous user, or log delivery user required ACL permissions for the bucket.

Step 6 In the middle of the page, click **Export** to get the bucket ACL configuration. The file includes the user type, account, bucket access, and ACL access.

Step 7 In the middle of the page, click **Add** to apply specific ACL permissions to an account.

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

Click **OK**.

----End

Follow-up Procedure

After a specified account is granted the ACL permissions for a bucket, the authorized user can use the AK and SK to access that bucket by adding the bucket to OBS Browser+.

After certain permissions are granted to an anonymous user, the anonymous user can access the bucket without any authentication. The anonymous user can be either registered or non-registered. A registered anonymous user can use either of the methods above to access the bucket, while a non-registered anonymous user can access the bucket in any of the following ways:

- Access the bucket's domain name in a browser to view the objects in the bucket.
- Configure the bucket's domain name in a third-party system to directly connect to the bucket.

6.5 Configuring an Object ACL

Prerequisites

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

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

Procedure

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

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

Step 3 Click a desired object.

Step 4 On the **Object ACL** page, choose a permission from **Private** and **Public Read** to grant object ACL permission for anonymous users.

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

Figure 6-11 Changing a public access permission
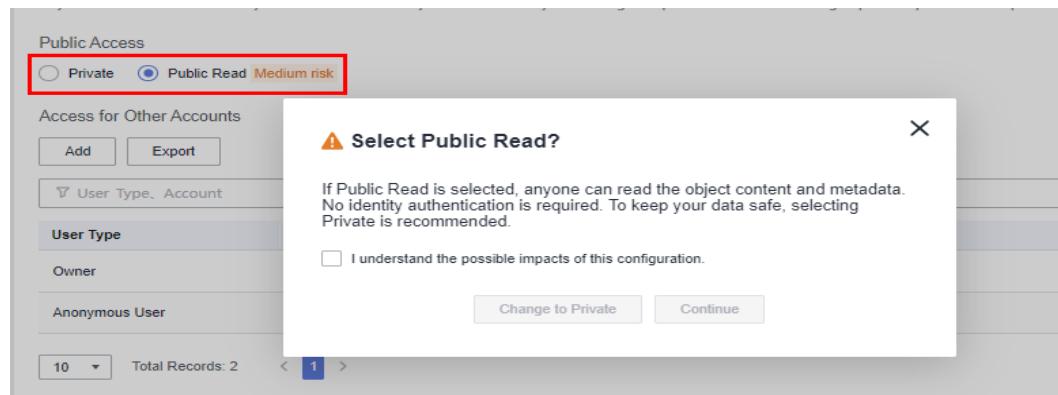
Step 5 Click **Edit** to grant the owner, anonymous user, or other accounts required permissions for the object.

📖 **NOTE**

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

Step 6 Click **Export** to get the object ACL configuration. The file includes the user type, account, object access, and ACL access.

Step 7 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

7 Data Management

7.1 Configuring a Lifecycle Rule

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

- Transition objects from Standard to Infrequent Access or Archive.
- Transition objects from Infrequent Access to Archive.
- Expire objects and then delete them.

Procedure

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

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

Step 3   In the navigation pane, choose **Data Management** > **Lifecycle Rules**.

Step 4   Click **Create**. A dialog box shown in **Figure 7-1** is displayed.

Figure 7-1 Creating a lifecycle rule
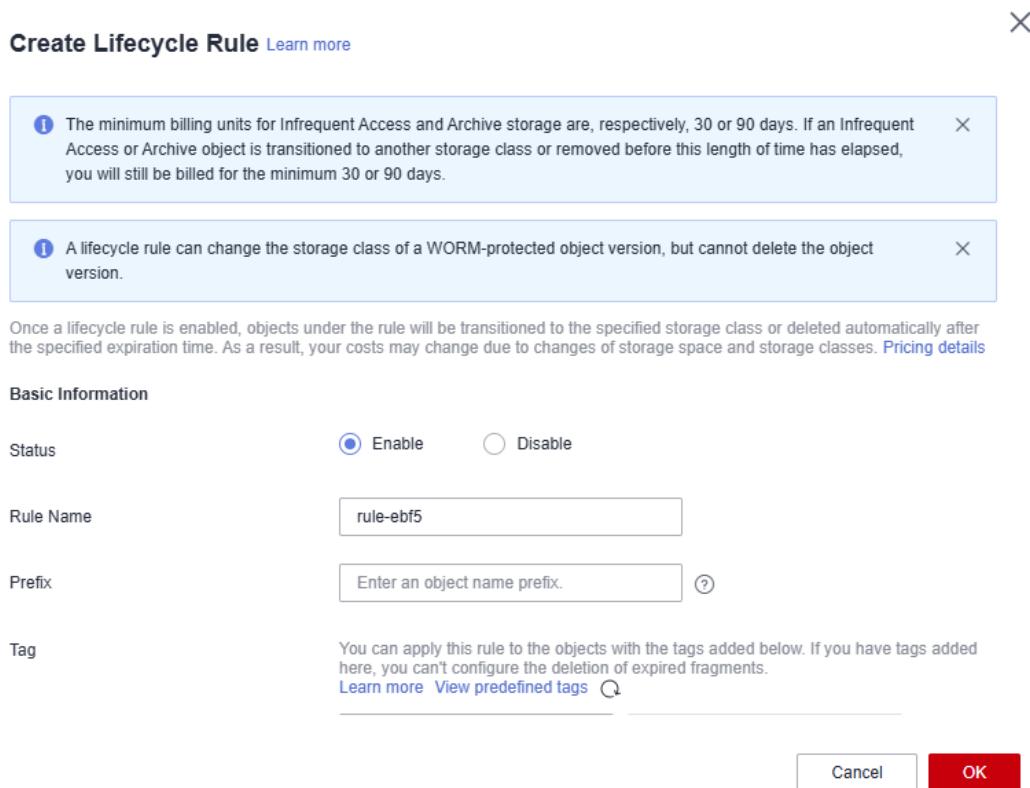
Step 5 Configure a lifecycle rule.

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.

- **Tag** (optional): You can apply a lifecycle rule to the objects with specific tags. A maximum of 10 tags can be added. Once tags are added, the deletion of expired fragments cannot be configured.

  📖 **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.

Current Version or **Historical Version**:

📖 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. For more information, see **Versioning**.
- You can configure either the **Current Version** or **Historical Version**, or both of them.

- **Transition to Infrequent Access After (Days)**: After this number of days since the last update, objects meeting specified conditions will be transitioned to Infrequent Access. This number must be at least 30.

- **Transition to Archive After (Days)**: After this number of days since the last update, objects meeting specified conditions will be transitioned to Archive. If you configure to transition objects first to Infrequent Access and then Archive, the objects must stay Infrequent Access at least 30 days before they can be transitioned to Archive. If transition to Archive is used, but transition to Infrequent Access is not, there is no limit on the number of days for transition.

- **Delete Objects After (Days)**: After this number of days since the last update, objects meeting certain conditions will be expired and then deleted. This number must be an integer larger than that specified for any of the transition operations.

- **Delete Fragments After (Days)**: After this number of days since the fragment generation, OBS will automatically delete fragments in the bucket.

- **Remove Expired Delete Marker**: An expired delete marker is the only single delete marker that remains after all historical versions of an object are deleted. This action will remove expired delete markers and help improve the list performance.

📖 NOTE

The object update time refers to when common objects were uploaded or when historical objects became historical.

- If an object of the current version is configured to be deleted upon expiration, its delete marker will also be deleted. Deleting all current and historical versions of an object will also delete its all delete markers.
- If the current and historical versions of an object are configured to be deleted upon expiration, its all delete markers will also be deleted.
- If **Remove Expired Delete Marker** takes effect, the expired delete markers will be 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.

One day, supposed you configure objects with the **log** prefix to be transitioned to Infrequent Access 30 days later, to Archive 60 days later, and then to be deleted 100 days later. OBS would perform these actions on **log/clientlog.log**, **log/serverlog.log**, **log/test1.log**, and **log/test2.log** as you defined.

📖 **NOTE**

In theory, it takes 24 hours at most to execute a lifecycle rule. After an object is updated, OBS calculates its lifecycle from the next 00:00 (UTC time), so there may be a delay of up to 48 hours in transitioning objects between storage classes or deleting expired objects. If you make changes to an existing lifecycle rule, the rule will take effect again.

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

----End

Follow-up Procedure

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

You can also select multiple lifecycle rules at a time and click **Delete** or **Disable** (or **Enable**) above the list to batch delete or disable (or enable) them.

7.2 Adding Tags to a Bucket

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

Procedure

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

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

Step 3   In the navigation pane, choose **Basic Configurations** > **Tagging**.

Step 4   Click **Add Tag**. The **Add Tag** dialog box is displayed. See **Figure 7-2** for details.

Figure 7-2 Add Tag
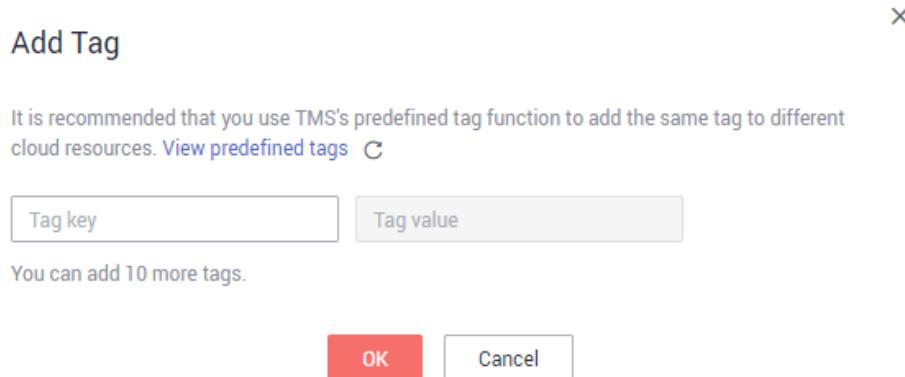
Step 5 Set the key and value based on **Table 7-1**.

Table 7-1 Parameter description

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

Step 6 Click **OK**.

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

----End

Related Operations

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

7.3 Configuring a Bucket Inventory

Procedure

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

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

Step 3 In the navigation pane, choose **Inventories**. The inventory list is displayed.

Step 4 Click **Create**. The **Create Inventory** dialog box is displayed.

Figure 7-3 Inventory settings

Step 5 Configure required parameters.

Table 7-2 Parameters for configuring a bucket inventory

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

Parameter	Description
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 6 Click **Next** to go to the **Configure Report** page.

Figure 7-4 Configuring the report
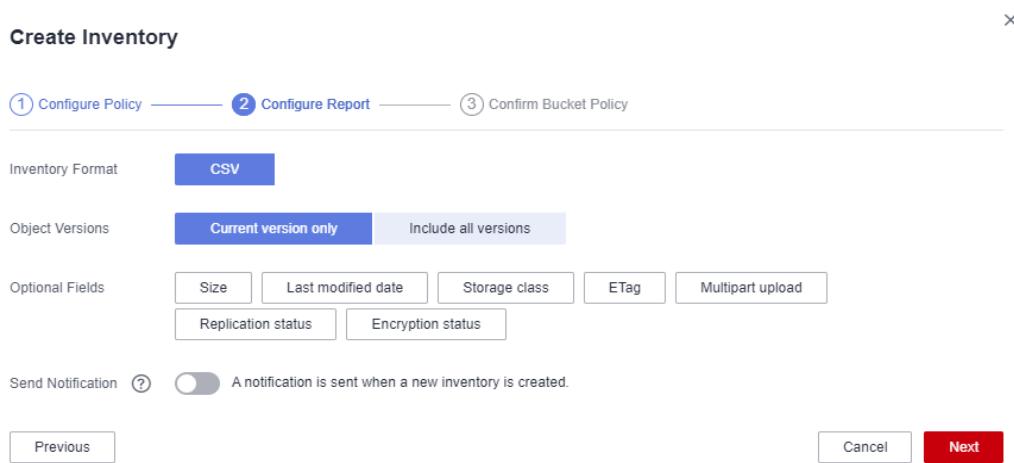
Step 7 Configure the report.

Table 7-3 Report related parameters

Parameter	Description
Inventory Format	Inventory files can only be saved in CSV format.
Object Versions	Object versions that you want to list in an inventory file. It can be set to **Current version only** or **Include all versions**.
Optional Fields	Object information fields that can be contained in an inventory file, including **Size**, **Last modified date**, **Storage class**, **ETag**, **Multipart upload**, **Encryption status**, and **Replication status**. For details about the fields, see **Metadata in an Inventory File**.

Parameter	Description
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 8 Click **Next** to confirm the bucket policy.

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

Step 9 Click **OK**.

----End

7.4 Configuring Event Notifications

7.4.1 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 **Event Notifications**.

Procedure

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

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

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

Step 4 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 5 Click **Create**. The **Create Event Notification** dialog box is displayed. See **Figure 7-5** for details.

Figure 7-5 Creating an event notification rule
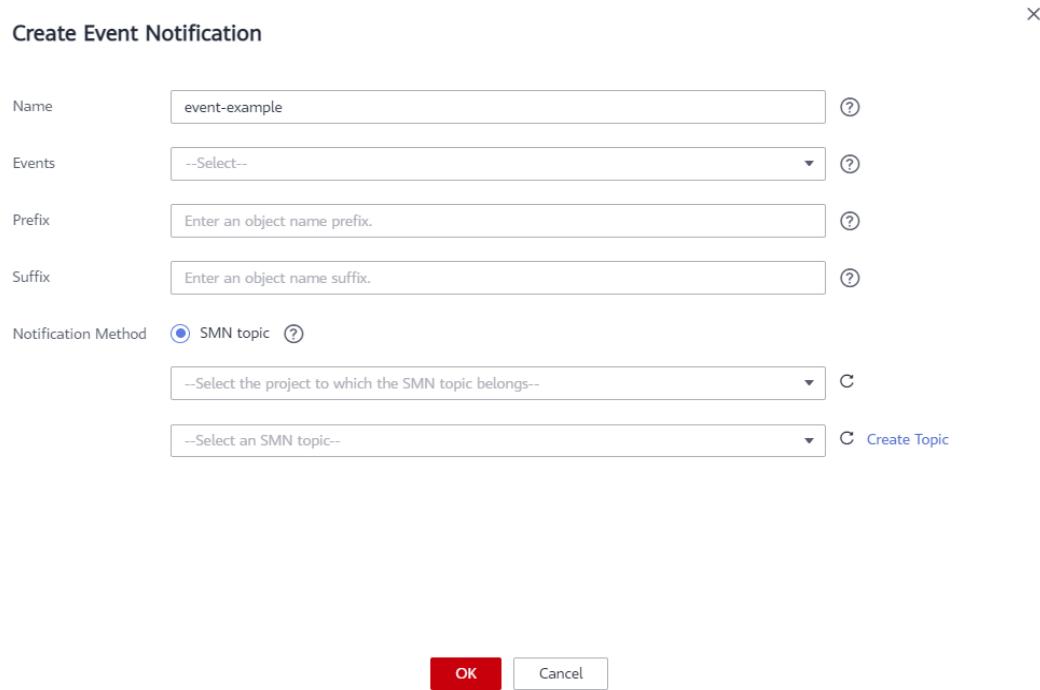
Step 6 Configure event notification parameters, as described in **Table 7-4**.

Table 7-4 Event notification parameters

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

Parameter	Description

Events	Various types of events. Currently, OBS supports event notification for the following types of events:
	
	● **ObjectCreated**: all kinds of object creation operations, including PUT, POST, COPY, and part assembling
	
	  – **Put**: Creates or overwrites an object using the PUT method.
	
	  – **Post**: Creates or overwrites an object using the POST (browser-based upload) method.
	
	  – **Copy**: Creates or overwrites an object using the COPY method.
	
	  – **CompleteMultipartUpload**: Assembles parts of a multipart upload.
	
	● **ObjectRemoved**: Deletes an object.
	
	  – **Delete**: Deletes an object with a specified version ID.
	
	  – **DeleteMarkerCreated**: Deletes an object without specifying a version ID.
	
	Multiple event types can be applied to the same object. For example, if you have selected **Put**, **Copy**, and **Delete** in the same event notification rule, a notification will be sent to you when the specified object is uploaded to, copied to, or deleted from the bucket. **ObjectCreated** contains **Put**, **Post**, **Copy**, and **CompleteMultipartUpload**. If you select **ObjectCreated**, the events **ObjectCreated** contains are automatically selected. Similarly, if you select **ObjectRemoved**, **Delete** and **DeleteMarkerCreated** are automatically selected.

Prefix	Object name prefix for which notifications will be triggered. **NOTE** If neither the **Prefix** nor the **Suffix** is configured, the event notification rule applies to all objects in the bucket.
Suffix	Object name suffix for which notifications will be triggered. **NOTE** ● A folder path ends with a slash (/). Therefore, if you want to configure event notification rules for operations on folders and you need to filter folders by suffix, the suffix must also end with a slash (/). ● If neither the **Prefix** nor the **Suffix** is configured, the event notification rule applies to all objects in the bucket.

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

Step 7 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.

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

Step 2 Create a bucket.

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

Step 3 Create a folder.

Click the name of the bucket created in **Step 2** to go to the **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.

Figure 7-6 Creating a folder
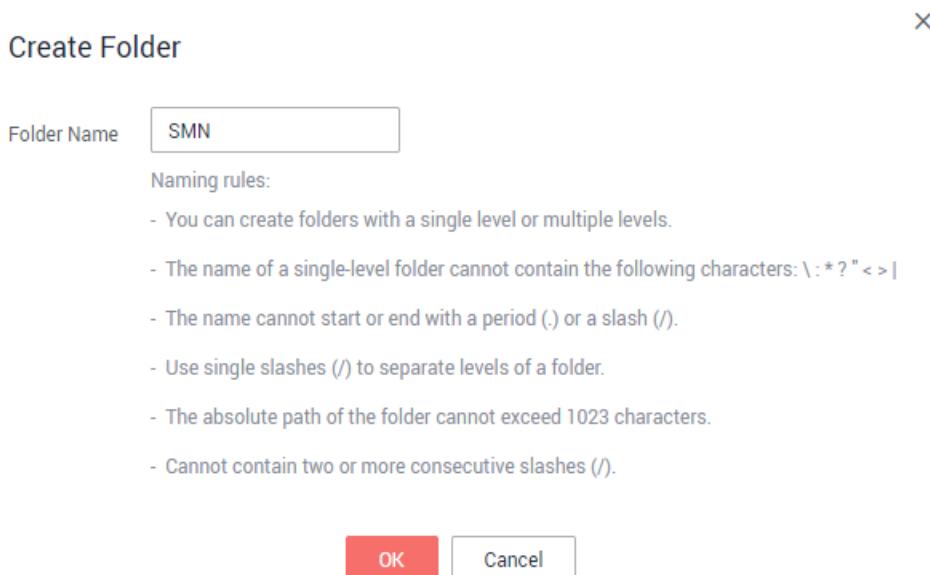
Step 4 In the upper left corner of the page, click ☰ and choose **Simple Message Notification**. On the displayed SMN page, create a topic.

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 **SMN Topic Management**.

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, as shown in **Figure 7-7**. After the notification is configured, an employee will be informed of all specified operations on the **SMN** folder in bucket **testbucket**. For details about the parameters, see **Table 7-4**.

📖 **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.

Figure 7-7 Adding an event notification rule
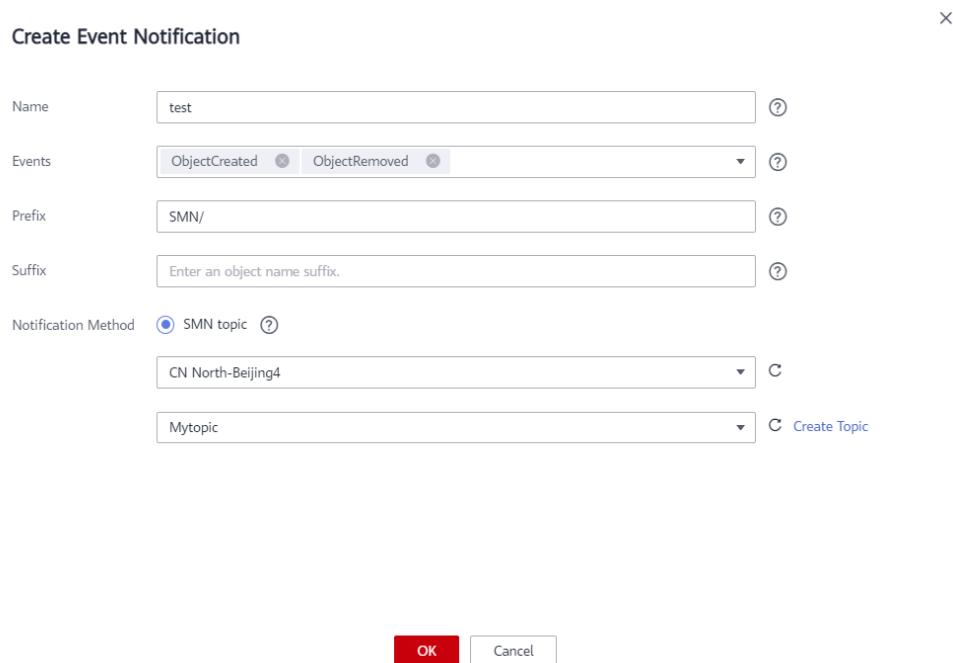
----End

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

7.5 Using Back to Source by Mirroring to Retrieve Data

Usually, if the data requested by a client using GET is not found in OBS, a 404 error will be returned. To avoid this error, OBS provides back to source that pulls the requested data from the source site (origin server) if it is not found in OBS.

Process of Back to Source by Mirroring

If the data you requested is not found in a bucket that has a back-to-source by mirroring rule configured and the request matches the rule, OBS will pull the data from its origin server and return it to you. This process does not interrupt services. Therefore, you can use this function to seamlessly migrate data from the origin server to OBS without being sensed by users, at low costs. **Figure 7-8** shows the process of back to source by mirroring.

Figure 7-8 Process of back to source by mirroring
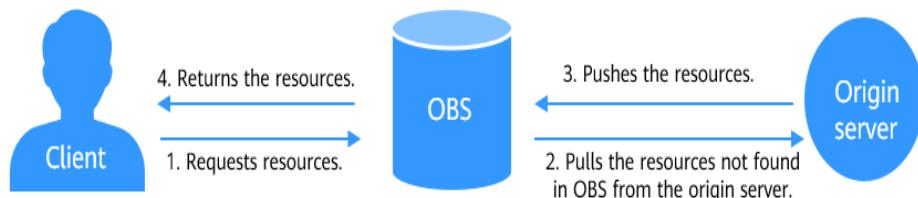
You can further configure a redirection address according to the settings of back to source by mirroring. If OBS fails to obtain resources from the origin server and the redirection rule is matched, OBS will return a 302 status code and a redirection address to the client. The client can directly request resources from the redirection address.

Figure 7-9 shows the redirection process.

Figure 7-9 Process of redirecting a request
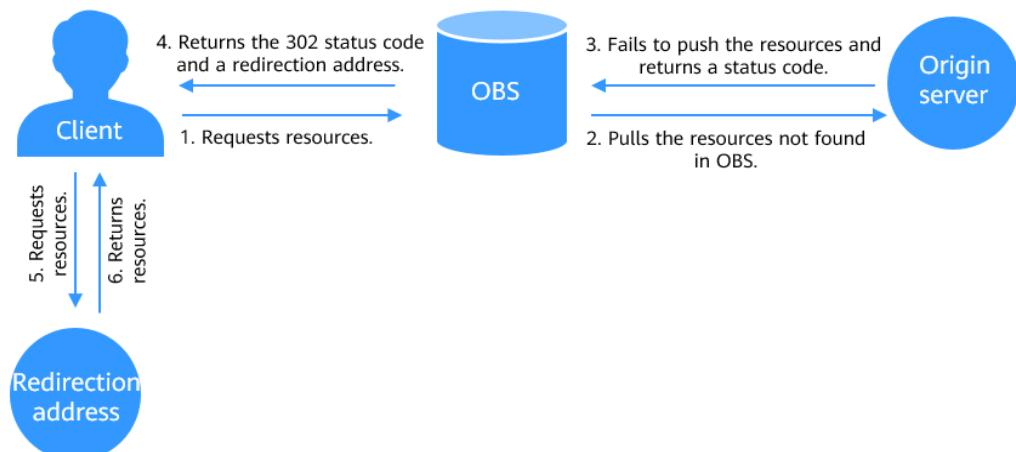
Back-to-Source Rules

Item	Description
Conditions for triggering an origin pull rule	OBS pulls data from the origin server only when a GetObject request results in a 404 response.
Rules for naming the retrieved files	If OBS retrieves a file from the origin server via **http(s)://MirrorURL/ObjectName**, the file is stored in OBS under the same name, **ObjectName**. For example, if the source address set for a bucket is **https://yun.com** and the requested file **example.jpg** is not found in the bucket, OBS will retrieve the file via **https://yun.com/example.jpg** and file **example.jpg** is then pushed to OBS.
Status code return rules in the case of retrieval failures	If OBS fails to retrieve a file from the source, it returns 404 to the client by default.
Rules for updating the retrieved files	If a file has been pushed by the origin server to OBS through mirroring, OBS does not update the file even if the source file changes.
HTTP request rules	By default, headers and query strings included in requests to OBS are not transferred to the origin server. However, you can configure the back to source settings to specify whether they should be passed along.

Constraints

Table 7-5 Back to source constraints

Category	Description
Bucket versions	Only buckets of version 3.0 or later support back to source by mirroring.
Time	It takes about five minutes to apply any changes to a back-to-source by mirroring rule.
Number of rules	A maximum of 10 back-to-source by mirroring rules can be configured for a bucket.

Category	Description

Functions	- Anonymous users are not allowed to configure back-to-source by mirroring rules. - Federated users cannot use back to source by mirroring. For more information about federated users, see **What Are the Relationships Between a Huawei Cloud Account, HUAWEI ID, IAM User, and Federated User?** and **Introduction to Identity Providers**. - Parallel file systems do not support back-to-source by mirroring rules. - Static website hosting does not support back to source by mirroring. Specifically, if 404 is returned when you use a static website domain name to download an object, back to source by mirroring will not be triggered. - A bucket cannot mirror itself. - Currently, back to source by mirroring from private buckets is only supported for some cloud vendors. **Transfer-Encoding: chunked** cannot be used for the origin server to transmit data, or the origin pull will fail. The response to the request for downloading an object from the origin server must contain the **Content-Length** header to specify the size of the source object. To achieve this, in the **Create/Edit Back-to-Source Rule** window, choose **Pass all parameters** for **HTTP Header Pass Rule**, enable **Do not pass specified parameters**, and add **Accept-Encoding**. - If NGINX was deployed as a reverse proxy for your origin server, turn off **chunked_transfer_encoding** of the NGINX.
	`location / {`
	`    chunked_transfer_encoding off;`
	`}`

Permissions	- To configure, obtain, or delete back-to-source by mirroring rules, you must have the **Tenant Administrator** permission assigned by using IAM. - You must create a cloud service agency using IAM to delegate OBS to pull data from the origin server. The agency must include the **obs:object:PutObject**, **obs:object:GetObject**, **obs:bucket:ListBucket**, and **obs:object:AbortMultipartUpload** permissions. - If SSE-KMS is enabled for a bucket, the **kms:cmk:get**, **kms:cmk:list**, **kms:cmk:create**, **kms:dek:create**, and **kms:dek:crypto** permissions must be configured for the IAM agency for OBS.
Others	- Back to source by mirroring is free now. - An object cannot match two different back-to-source by mirroring rules.

Creating a Back-to-Source by Mirroring Rule

You can use OBS Console to create a mirroring back-to-source rule.

Using OBS Console

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

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

Step 3 In the navigation pane, choose **Back to Source**.

Step 4 Click **Create**.

Step 5 Configure parameters by referring to **Table 7-6**.

Table 7-6 Parameters for configuring a back-to-source by mirroring rule

Parameter	Description
Resource Type	The type of the origin server ● **Public**: The origin server data comes from public object storage. ● **Private**: The origin server data comes from private object storage of some cloud vendors.
Back-to-Source Condition	The conditions that trigger the back-to-source rule The back-to-source rule is triggered when the following conditions are met: a. The requested object starts with the specified file name prefix. b. An HTTP status code 404 is returned because the object is not found in the bucket. The file name prefix: ● Cannot exceed 1,023 characters. ● Cannot contain or overlap with any other file name prefix specified for an existing rule. ● Can be left blank, which means that the rule applies to all files that do not meet the conditions of other back-to-source rules configured for the bucket. A bucket can have only one back-to-source rule that does not have a file name prefix specified. Example: The file name prefix is set to **123/**. A request for **123/456.txt** (that is not found in OBS) triggers OBS to pull **123/456.txt** from the origin server.

Parameter	Description
Add Prefix or Suffix	When OBS requests data from the origin server, it adds the specified prefix or suffix to the name of the object requested by the client. However, the object returned to OBS and the client retains its original name, without the added prefix or suffix. Example: The prefix is set to **123**. A request for **abc.txt** triggers OBS to pull **123abc.txt** from the origin server. However, the object is saved in OBS and returned to the client under its original name **abc.txt**.
Replace Prefix With	When OBS requests data from the origin server, it uses the characters specified here to replace the file name prefix defined in the back-to-source condition. However, the object returned to the client retains its original name. Example: The file name prefix is set to **123** as the back-to-source condition and **abc** is used to replace the prefix. In this case, a request for **123456.txt** triggers OBS to fetch **abc456.txt** from the origin server. However, the object is saved in OBS and returned to the client under its original name **123456.txt**.

Source URL	The address of the origin server. Both active and standby sites can be configured.
	
	During the back-to-source process, OBS preferentially pulls data from the active site, or polls all active sites if multiple are available. If there are two or more active sites, when the first attempt to an active site fails and the retry conditions are met, OBS will retry another active site. There must be at least one active site. Up to five active sites can be configured. If data retrieval from all active sites fails, OBS will pull data from standby sites.
	
	Example site address: **http://***Source domain name***/***Static path* or **https://***Source domain name***/***Static path*
	
	● *Source domain name* indicates the domain name of the origin server.
	  – If the origin server is a bucket accessible over HTTP, use the bucket's domain name as *Source domain name*.
	  – If the origin server is a private bucket hosted by another cloud vendor, use the regional domain name as *Source domain name*. At present, only private buckets of some cloud vendors are supported.
	● *Static path* indicates the directory that stores the target file. For example, if the static path is **123/**, the target file is in the **123/** directory.
	● The source URL can be an origin server IP address, and it must be a public IP address.

Parameter	Description
Retry Condition	When a retry is triggered **4XX** and specific status codes starting with **4** cannot be configured together. The same applies to **5XX** and specific status codes starting with **5**. A maximum of 20 status codes can be configured.
Include request strings	After this option is enabled, query parameters in the request URL will be passed to the origin server.
Redirect requests to the returned location	After this option is enabled, if redirection is configured for the origin server, the resources will be requested from the location included in the 3*xx* response returned by the origin server and be saved to OBS. Each request supports up to 10 redirects.
Redirect without Referer	If redirection has been configured for the origin server, enabling this option will remove the Referer header from requests during the redirection process.
HTTP Header Pass Rule	You can specify the HTTP header parameters that can be passed to the origin server when a request sent to OBS triggers the back-to-source by mirroring rule. **Example for Configuring an HTTP Header Pass Rule** gives a configuration example. ● **Pass all parameters**/**Pass specified parameters**: Set the HTTP header parameters that can be passed. ● **Do not pass specified parameters**: Set the HTTP header parameters that cannot be passed. If a client's request includes the specified headers, OBS will not pass them to the origin server. If a header is specified for both the pass and do-not-pass categories, it is deemed as a do-not-pass parameter. ● **Configure custom parameters**: You can set a custom value for a specified header. If a client request includes that header, OBS overrides it with the custom value before passing it to the origin server.
IAM Agency	An IAM agency is required to delegate OBS to pull data from the origin server. The agency must include the **obs:object:PutObject**, **obs:object:GetObject**, **obs:bucket:ListBucket**, and **obs:object:AbortMultipartUpload** permissions. If no such IAM agency is available, create one by referring to **Creating an Agency for Back to Source**.

Parameter	Description
Redirection Settings	After a redirection address is configured, if OBS fails to pull the requested resource from the origin server, OBS returns a 302 status code along with the redirection address to the client. The client then requests the resource from that address. **Redirection Match**: ● **Forward match**: Back to source by redirection will be performed if the returned status code is any of those you specified. ● **Reverse match**: Back to source by redirection will be performed if the returned status code is not among those you specified. **Redirection URL**: **http://***Source domain name/Static path* or **https://***Source domain name/Static path* ● A source domain name is the domain name of the origin server to which requests are redirected. ● *Static path* indicates the directory that stores the target file. For example, if the static path is **123/**, the target file is in the **123/** directory.

Step 6 Click **OK**. After the rule is created, it appears in the list on the back to source page.

----End

References

Example for Configuring an HTTP Header Pass Rule

Assume that the parameters are set as shown in **Figure 7-10**.

Figure 7-10 Configuring an HTTP header pass rule

Based on the preceding configuration, if the header of the request sent to OBS is as follows:

```
GET /ObjectName HTTP/1.1
Host: bucketname.obs.eu-west-101.myhuaweicloud.eu
aaa:aaa
bbb:bbb
ccc:ccc
```

After the back-to-source rule is triggered, OBS sends the following request to the origin server:

```
GET /ObjectName HTTP/1.1
Host: source.com
aaa:aaa
ccc:111
```

Notes for passing HTTP headers during back to source

- HTTP headers that can be passed from an origin server to a client:
  - Content-Type
  - Content-Language
  - Content-Encoding
  - Content-Disposition
  - Cache-Control
  - Expires
- HTTP headers that cannot be passed from a client to an origin server:
  a. HTTP headers starting with the prefix below:

     x-obs-

  b. All standard HTTP headers, including:

     - Content-Length

     - Authorization2

     - Authorization

     - Range

     - Date

Creating an Agency for Back to Source

Step 1 In the **Create Back-to-Source Rule** dialog box on OBS Console, click **View Agencies** to jump to the **Agencies** page on the IAM console.

Step 2 Click **Create Agency**.

Step 3 Enter an agency name.

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

Step 5 Select **OBS** for **Cloud Service**.

Step 6 Set a validity period.

Step 7 Click **Next**.

☐ NOTE

The console for creating an agency has the new and old editions. The following steps use the new edition.

Step 8 Click **Done**. In the displayed dialog box, click **Authorize**.

Step 9 If there is already a policy meeting your requirements, go to **Step 11**. Otherwise, click **Create Policy** in the upper right corner. Then, specify a policy name and choose **Visual editor** for **Policy View**.

Figure 7-11 Authorizing an agency
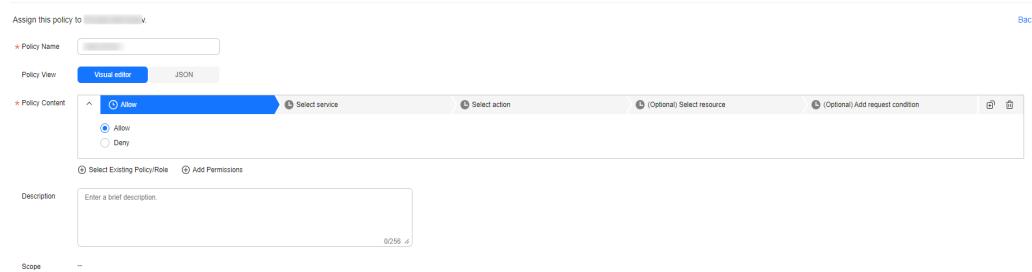
Step 10 Configure **Policy Content** as follows:

1. Select **Allow**.

2. For the service, select **Object Storage Service (OBS)**.

3. For actions, under **ReadOnly**, select **obs:object:GetObject**, under **Read/Write**, select **obs:object:PutObject** and **obs:object:AbortMultipartUpload**, and under **ListOnly**, select **obs:bucket:ListBucket**.

4. For **Resources**, select **All**.

5. Click **Next**.

Step 11 Select the policy and click **Next**.

Step 12 On the **Select Scope** page, select **All resources** for **Scope** and click **OK**.

☐ NOTE

All resources is the recommended option for **Scope**. In this policy, the **All resources** option means that OBS can use all resources, including those in enterprise projects, region-specific projects, and global services under the account based on assigned permissions.

----End

8 Data Access

8.1 Static Website Hosting

8.1.1 Configuring Static Website Hosting

You can configure static website hosting for a bucket and then use the bucket's domain name to access static websites hosted in the bucket.

It can take up to two minutes for the configuration of static website hosting to take effect.

📖 NOTE

In static website hosting scenarios, anonymous users must be granted access to hosted static website files. When they access the hosted files, there will be costs on outbound Internet traffic and requests.

Precautions

For security and compliance purposes, Huawei Cloud OBS prohibits the use of static website hosting based on the default OBS domain name (**a bucket domain name or static website domain name**). When you use such a domain name to access web pages in a browser, no content will be displayed. Instead, the content is downloaded as an attachment.

Prerequisites

Web page files required for static website hosting have been uploaded to the specified bucket.

The static website files hosted in the bucket are accessible to anonymous users.

Static web page files in the Archive storage class have been restored. For more information, see **Restoring an Object from Archive Storage**.

Procedure

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

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

Step 3 (**Optional**) If the static website files in the bucket are not accessible to everyone, perform this step. If they are already accessible to everyone, skip this step.

To grant required permissions, see **Granting Anonymous Users the Read Permission for Certain 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 Bucket Policy**.
3. Configure bucket policy information.

Table 8-1 Parameters for configuring a public read policy

Parameter		Description
Configuration method		**Visual Editor** and **JSON** are available. Choose **Visual Editor** here. For details about using JSON configuration method, 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 4 In the navigation pane, choose **Data Management** > **Static Website Hosting**.

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

Step 6 Enable **Status**.

Step 7 Set the hosting type to the current bucket. For details, see **Figure 8-1**.

Figure 8-1 Configuring static website hosting
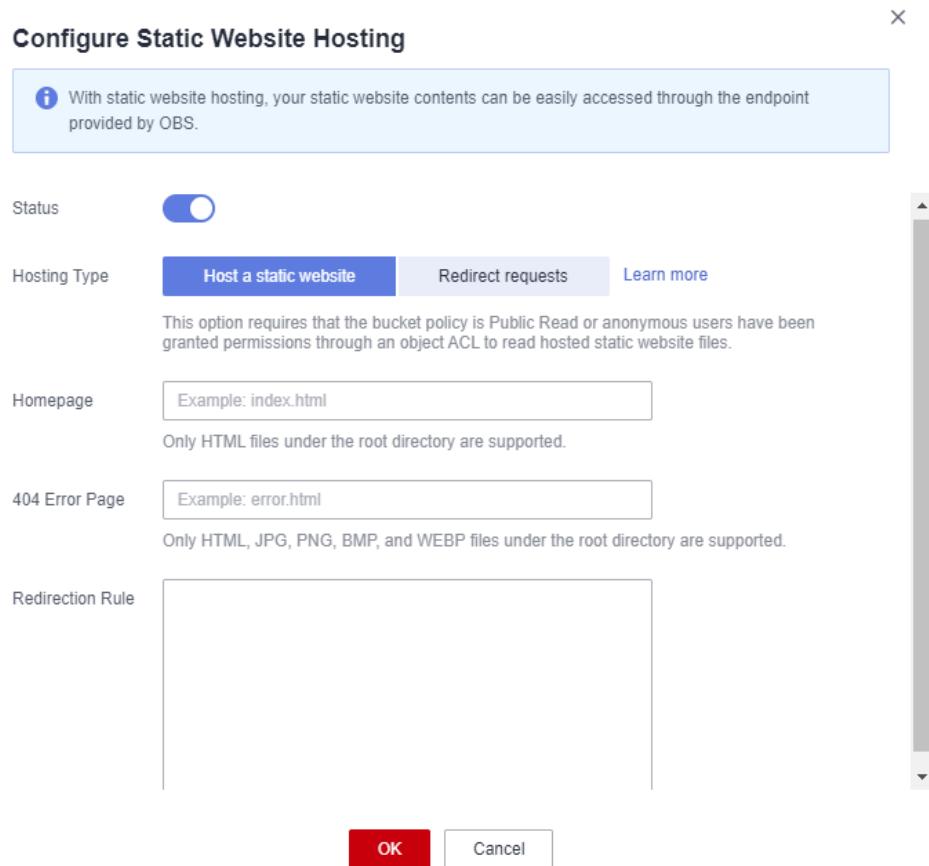
Step 8 Configure the homepage and 404 error page.

- **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 any restriction but the **Content-Type** of objects must be specified.

  OBS only allows files such as **index.html** in the root directory of a bucket to function as the default homepage. Do not set the default homepage with a multi-level directory structure (for example, **/page/index.html**).

- **404 Error Page**: specifies the error page returned when an error occurs during static website access. When OBS Console is used to configure static website hosting, only HTML, JPG, PNG, BMP, and WebP files under the root directory are supported. When APIs are used to configure static website hosting, OBS does not have any restriction but the **Content-Type** of objects must be specified.

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

A redirection rule is compiled in the JSON or XML format. Each rule contains a **Condition** and a **Redirect**. The parameters are described in **Table 8-2**.

Table 8-2 Parameter description

Container	Key	Description
Condition	KeyPrefixEquals	Object name prefix on which the redirection rule takes effect. When a request is sent for accessing an object, the redirection rule takes effect if the object name prefix matches the value specified for this parameter. For example, to redirect the request for object **ExamplePage.html**, set the **KeyPrefixEquals** to **ExamplePage.html**.
	HttpErrorCodeReturnedEquals	HTTP error codes upon which the redirection rule takes effect. The specified redirection is applied only when the error code returned equals the value specified for this parameter. For example, if you want to redirect requests to **NotFound.html** when HTTP error code 404 is returned, set **HttpErrorCodeReturnedEquals** to **404** in **Condition**, and set **ReplaceKeyWith** to **NotFound.html** in **Redirect**.
Redirect	Protocol	Protocol used for redirecting requests. The value can be **http** or **https**. If this parameter is not specified, the default value **http** is used.
	HostName	Host name to which the redirection is pointed. If this parameter is not specified, the request is redirected to the host from which the original request is initiated.
	ReplaceKeyPrefixWith	The object name prefix used in the redirection request. OBS replaces the value of **KeyPrefixEquals** with the value you specified here for **ReplaceKeyPrefixWith**. For example, to redirect requests for **docs** (objects in the **docs** directory) to **documents** (objects in the **documents** directory), set **KeyPrefixEquals** to **docs** under **Condition** and **ReplaceKeyPrefixWith** to **documents** under **Redirect**. This way, requests for object **docs/a.html** will be redirected to **documents/a.html**.

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

Example of setting a redirection rule

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

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

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

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

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

```
[
  {
   "Condition": {
```

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

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

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

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

Step 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

8.1.2 Configuring Redirection

You can redirect all requests for a bucket to another bucket or URL by configuring redirection rules.

Prerequisites

Web page files required for static website hosting have been uploaded to the specified bucket.

The static website files hosted in the bucket are accessible to anonymous users.

Static web page files in the Archive storage class have been restored. For more information, see **Restoring an Object from Archive Storage**.

Procedure

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

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

Step 3 Click **Configure Static Website Hosting**. The **Configure Static Website Hosting** dialog box is displayed.

Step 4 Enable **Status**.

Step 5 Set **Hosting Type** to **Redirect requests**, as shown in **Figure 8-2**. In the text box of **Redirect To**, enter the bucket's access domain name or URL.

Figure 8-2 Configuring redirection
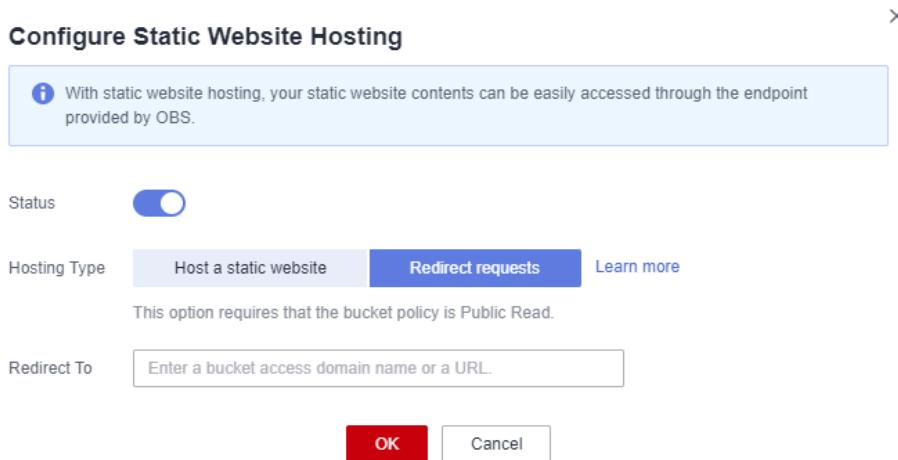
Step 6 Click **OK**.

Step 7 In the bucket list, click the bucket to which requests for the static website are redirected.

Step 8 (**Optional**) If the static website files in the bucket are not accessible to everyone, perform this step. If they are already accessible to everyone, skip this step.

To grant required permissions, see **Granting Anonymous Users the Read Permission for Certain 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 Bucket Policy**.
3. Configure bucket policy information.

Table 8-3 Parameters for configuring a public read policy

Parameter	Description
Configuration method	**Visual Editor** and **JSON** are available. Choose **Visual Editor** here. For details about using JSON configuration method, see **Creating a Custom Bucket Policy (JSON View)**.
Policy Name	Enter a custom policy name.

Parameter		Description
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 9 **Verification**: Input the access domain name of the bucket in the web browser and press **Enter**. The bucket or URL to which requests are redirected will be displayed.

☐ NOTE

In some conditions, you may need to clear the browser cache before the expected results are displayed.

----End

8.2 Configuring a User-Defined Domain Name

☐ NOTE

If an acceleration domain name is also needed, to prevent objects in OBS buckets from being directly downloaded upon access, you need to perform other required operations after the custom domain name and the acceleration domain name have been configured. For details, see **With CDN Acceleration Enabled, Why Are the Objects in My OBS Bucket Directly Downloaded When I Access Them?**

Procedure

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

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

Step 3 In the navigation pane, choose **Basic Configurations** > **Domain Name Mgmt**.

Figure 8-3 Domain name management page
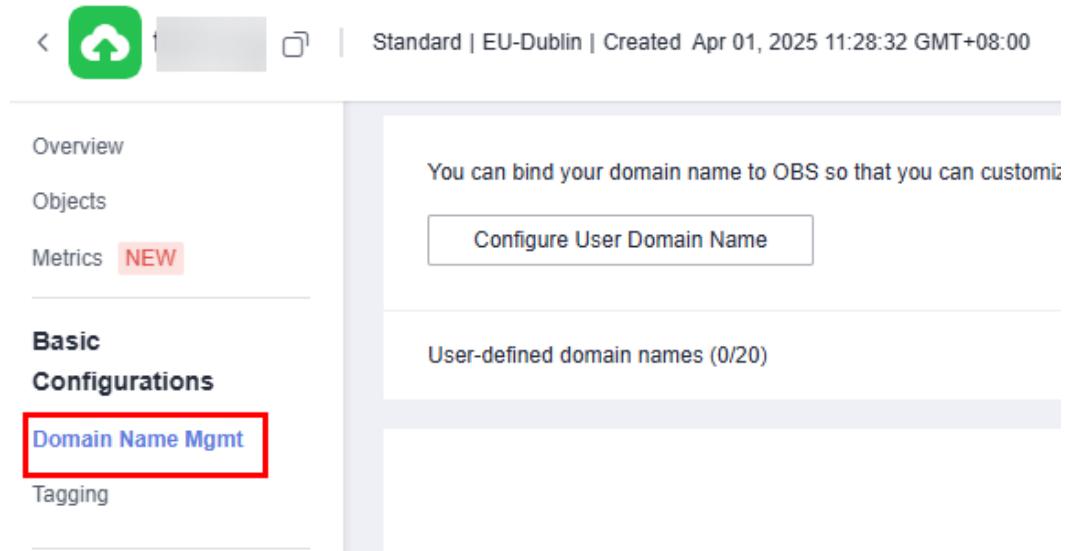
Step 4 Click **Configure User Domain Name** in the upper part of the page. Then, enter the domain name to configure, as shown in **Figure 8-4**. If you want to choose one of the existing Huawei Cloud domain names from the drop-down list on OBS Console, you must have the **Domains:domains:getDetails** permission. You can contact the administrator to use IAM to grant you this permission. If you do not have this permission, you can only manually type a domain name.

The suffix of a user-defined domain name can contain 2 to 6 uppercase or lowercase letters.

Figure 8-4 Configuring a user-defined domain name
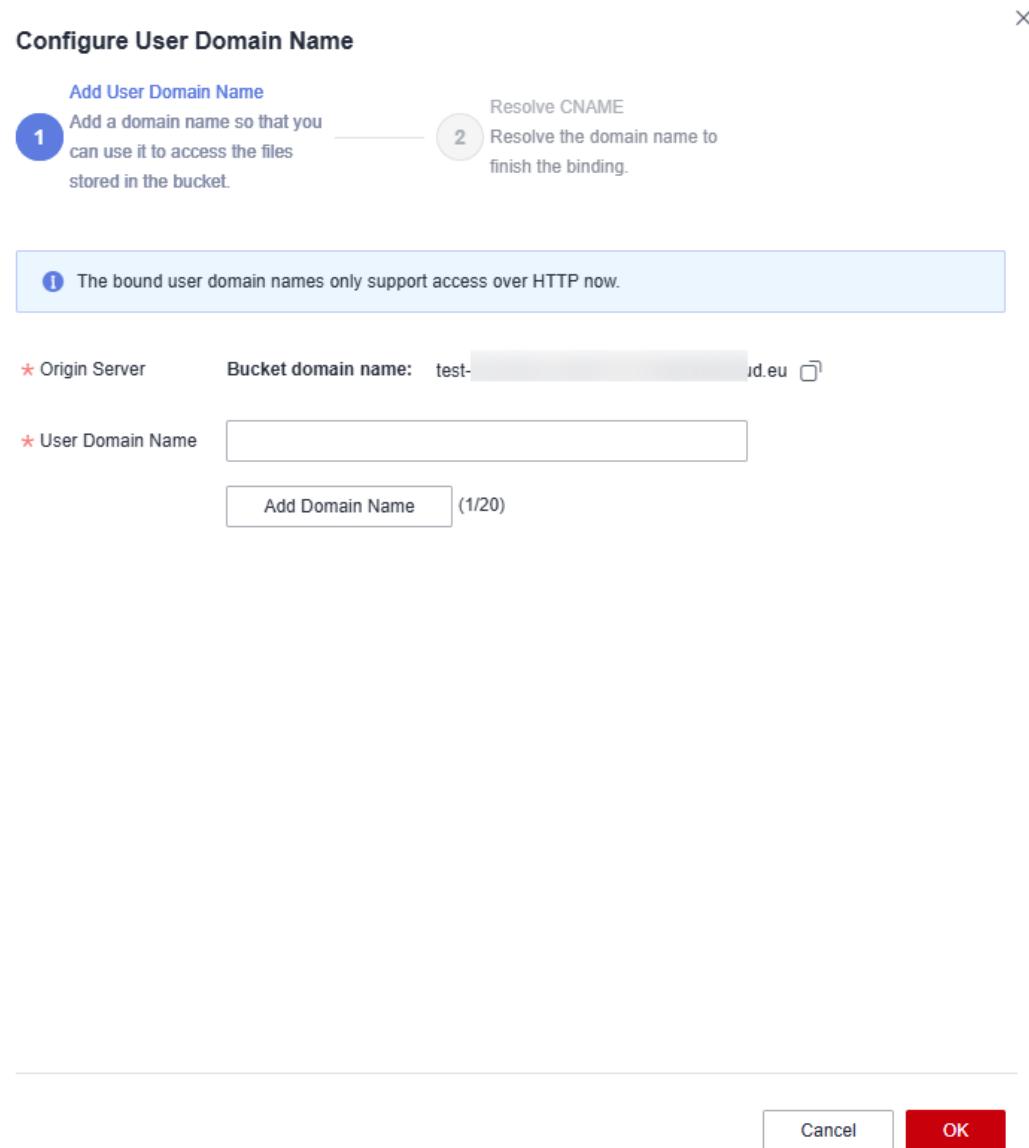
Step 5 Click **OK**.

Step 6 Configure a CNAME record on the DNS, and map the user-defined domain name (for example, **example.com**) to the domain name of the bucket.

The CNAME configuration varies depending on DNS providers.

If your DNS service is provided by Huawei Cloud, perform the following steps to configure a CNAME record:

1. Log in to the Huawei Cloud console. On the homepage, choose **Networking** > **Domain Name Service**. The DNS console is displayed.

2. In the navigation pane, choose **Public Zones**. The domain name list page is displayed.

3. Click the domain name which you want to add a record set to.

4. Choose the **Record Sets** tab and click **Add Record Set**.

5. Configure the parameters based on **Table 8-4**. Retain the default values for those not listed in the table below.

Table 8-4 Parameters for adding a record set

Parameter	Description	Example Value
Name	Prefix of the domain name	www
Type	Type of the record set, which should be a CNAME-Canonical name here.	CNAME – Map one domain to another
Line	Resolution line. The DNS server will return the IP address of the specified line, depending on where the visitor comes from. You must add a **Default** line to ensure that the website is accessible to anyone.	Default
TTL (s)	Cache duration of the record set, in seconds	The default interval is **5 min** (300 seconds).
Value	Domain name to be pointed to	– If CDN acceleration is not used, set this parameter to the bucket domain name. – If CDN acceleration is used, set this parameter to the CNAME record allocated by CDN.

6. Click **OK**.

7. Check whether the CNAME configuration takes effect.

   Open the Windows command line interface and run the following command:

   ```
   nslookup -qt=cname User-defined domain name bound to the bucket
   ```

   – Without CDN acceleration: If the output is the bucket domain name, the CNAME configuration has taken effect.

   – With CDN acceleration: If the output is the CNAME record allocated by CDN, the CNAME configuration has taken effect.

   ----End

9 Data Security

9.1 Configuring Server-Side Encryption

9.1.1 Configuring Bucket Server-Side Encryption

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

You can enable encryption (by choosing SSE-KMS or SSE-OBS) when creating a bucket (see **Creating a Bucket**). You can also enable or disable encryption for an existing bucket.

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

Enabling Server-Side Encryption for a Bucket

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

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

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

Step 4 In the **Basic Configurations** area, click **Server-Side Encryption**. The **Server-Side Encryption** dialog box is displayed.

Step 5 Enable **Server-Side Encryption**. Select **SSE-KMS** or **SSE-OBS**.

If you choose **SSE-KMS** for encryption, you must specify an encryption key type (**Default** or **Custom**). If **Default** is used, the default key of the current region will be used to encrypt your objects. If there is no such a default key, OBS creates one the first time you upload an object. If **Custom** is used, you can click **View KMS Keys** to switch to the KMS console to create a custom key. Then go back to OBS Console and select the key from the drop-down list.

Figure 9-1 Choosing SSE-KMS for a bucket

Server-Side Encryption

> ⓘ Data is automatically encrypted upon upload, improving data storage security.Learn more ⧉

Server-Side
Encryption   🔵 Enabled

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

Encryption Method | SSE-KMS | SSE-OBS |

Encryption keys managed by KMS are used to encrypt your objects. Learn more ⧉

Encryption Key Type | Default | Custom |

You can use a custom key below to encrypt your objects.

Custom   --Select--   ▾   🔍 View KMS Keys

Cancel   OK

When **SSE-OBS** is chosen, the keys created and managed by OBS are used for encryption.

Figure 9-2 Choosing SSE-OBS for a bucket

Server-Side Encryption

> ⓘ Data is automatically encrypted upon upload, improving data storage security.Learn more ⧉

Server-Side
Encryption   🔵 Enabled

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

Encryption Method | SSE-KMS | SSE-OBS |

Keys created and managed by OBS are used to encrypt your objects. Learn more ⧉

Cancel   OK

Step 6 Click **OK**.

----End

Disabling Server-Side Encryption for a Bucket

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

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

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

Step 4 In the **Basic Configurations** area, click **Server-Side Encryption**. The **Server-Side Encryption** dialog box is displayed.

Step 5 Select **Disable**.

Step 6 Click **OK**.

----End

9.1.2 Enabling Server-Side Encryption When Uploading an Object

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

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

Constraints

- The object encryption status cannot be changed.
- A key in use cannot be deleted. Otherwise, the object encrypted with this key cannot be downloaded.
- If an object is server-side encrypted and does not have any IAM agency, other accounts and users cannot access the object even if they can read this object.

Prerequisites

In the region where OBS is deployed, the **KMS Administrator** permission has been added to the user group. For details about how to add the permission, see **Assigning Permissions to an IAM User**. If the current account or user is the grantee, it also requires the **KMS Administrator** permission.

Procedure

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

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

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

Step 4 Add the files to be uploaded.

Step 5 Enable **Inherit from bucket** if your bucket has server-side encryption enabled and the objects to be uploaded need to inherit the encryption configuration of the bucket.

Figure 9-3 Inherit from bucket

If your bucket has server-side encryption disabled or it has server-side encryption enabled but you need a different encryption configuration for its objects, you can select **SSE-KMS** or **SSE-OBS** based on service requirements. If SSE-KMS is selected, you need to select an encryption key type.

- If **Default** is selected, the default key of the current region will be used to encrypt your objects. If there is no such a default key, OBS creates one the first time you upload an object.

- If **Custom** is used, you can click **View KMS Keys** to switch to the KMS console to create a custom key. Then go back to OBS Console and select the key from the drop-down list.

Figure 9-4 Configuring encryption for an object in a bucket with server-side encryption disabled
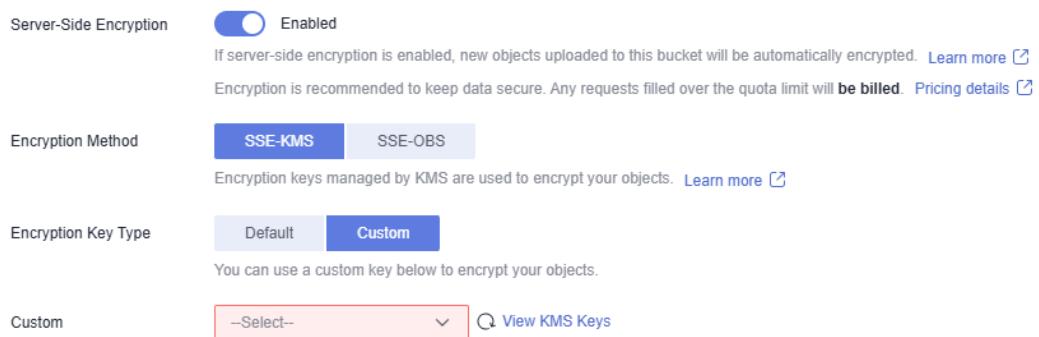
Figure 9-5 Configuring encryption for an object in a bucket with server-side encryption enabled
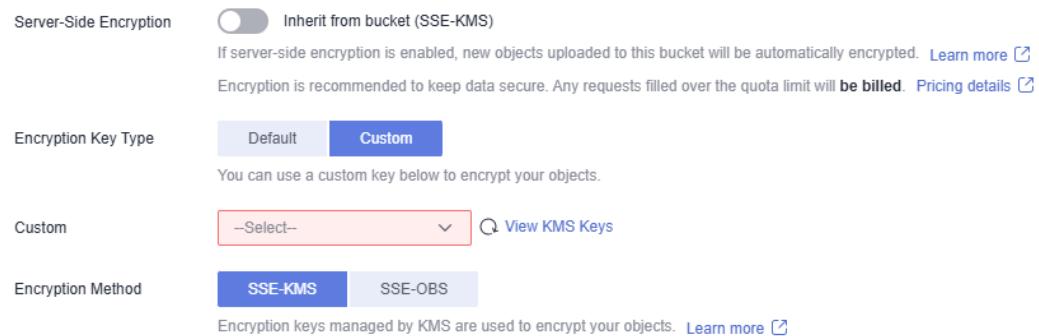
Step 6 Click **Upload**.

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

----End

9.2 Configuring WORM Retention

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

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

Prerequisites

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

Configuring WORM for a Bucket

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

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

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

Step 4 In the **Basic Configurations** area, click **WORM Retention**. The **Configure WORM Retention** dialog box is displayed.

Step 5 Choose **Configure**. Keep the default **Compliance** retention mode and specify a default retention period.

📖 NOTE

- Only the compliance retention mode is currently supported. In this mode, no users can delete protected object versions or change their retention mode during the specified retention period.

- During the specified default retention period, OBS prevents WORM-protected object versions from being deleted. You can configure a retention period in either days (from **1** to **36500**) or years (from **1** to **100**). The upper limit is 100 years.

- When you upload an object to a WORM-protected bucket, the object inherits the WORM retention from the bucket by default. You can also configure a different WORM retention for the object under advanced settings. If both a bucket-level and object-level WORM retention policy are applied to an object, the object-level retention policy will be used.

Figure 9-6 Configuring a WORM retention policy
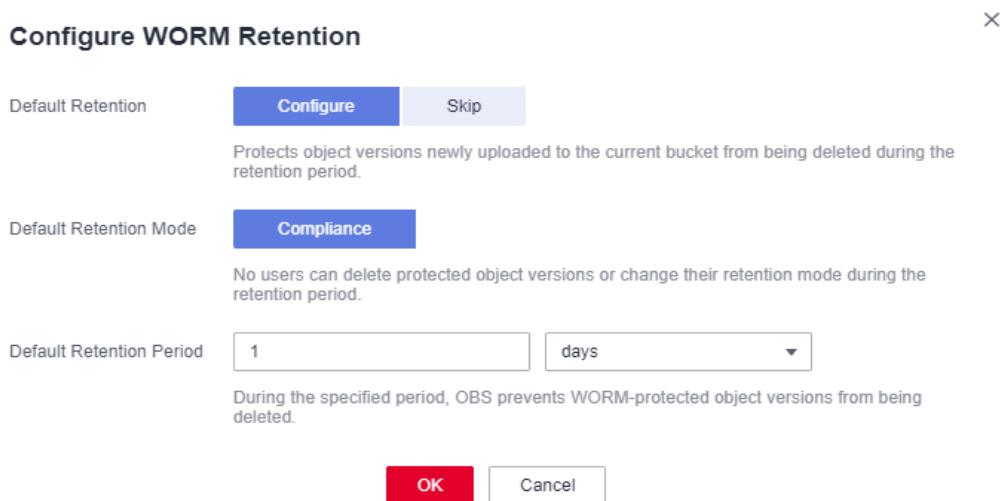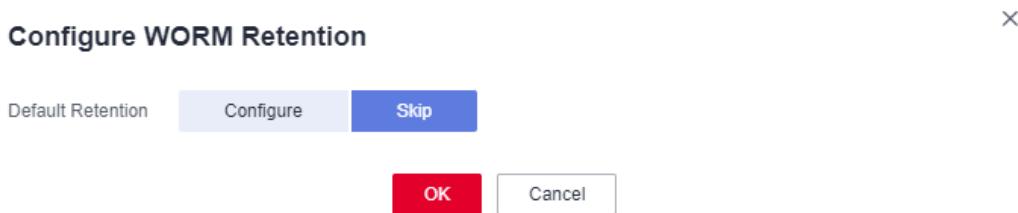

Step 6 Click **OK**.

----End

Skipping the WORM Retention Configuration

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

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

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

Step 4 In the **Basic Configurations** area, click **WORM Retention**. The **Configure WORM Retention** dialog box is displayed.

Step 5 Choose **Skip**.

Figure 9-7 Skipping the WORM retention configuration

Step 6 Click **OK**.

----End

Extending the Retention Period

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

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

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

Step 3 Locate the object version for which you want to extend the retention period, choose **More** > **Extend Retention Period**, and select a date.

📖 **NOTE**

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

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

----End

Related Operations

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

9.3 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 navigation pane of **OBS Console**, choose **Object Storage**.

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

Step 3 In the navigation pane, choose **Data Security** > **CORS Rules**.

Step 4 Click **Create**. The **Create CORS Rule** dialog box is displayed. See **Figure 9-8** for details.

📖 **NOTE**

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

Figure 9-8 Creating a CORS rule
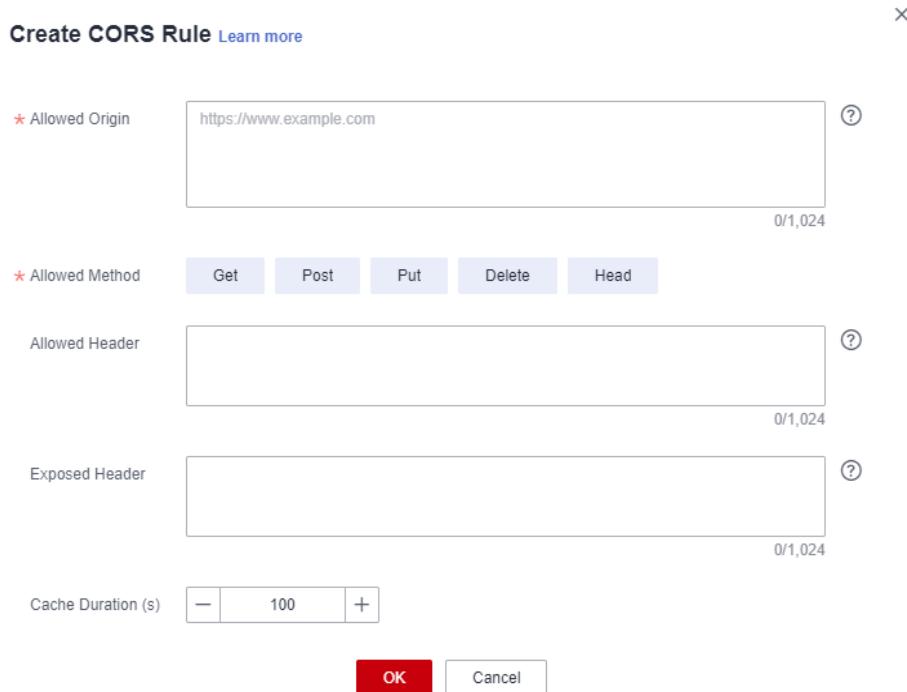
Step 5 In the **CORS Rule** dialog box, configure **Allowed Origin**, **Allowed Method**, **Allowed Header**, **Exposed Header**, and **Cache Duration (s)**.

Table 9-1 Parameters in CORS rules

Parameter	Description
Allowed Origin	Mandatory Specifies the origins from which requests can access the bucket. Multiple matching rules are allowed. One rule occupies one line, and allows one wildcard character (*****) at most. An example is given as follows: http://rds.example.com https://*.vbs.example.com
Allowed Method	Mandatory Specifies the allowed request methods for buckets and objects. The methods include Get, Post, Put, Delete, and Head.
Allowed Header	Optional Specifies the allowed headers in cross-origin requests. Only CORS requests matching the allowed headers are valid. You can enter multiple allowed headers (one per line) and each line can contain one wildcard character (*) at most. Spaces and special characters including **&:<** are not allowed.
Exposed Header	Optional Specifies the exposed headers in CORS responses, providing additional information for clients. By default, a browser can access only headers **Content-Length** and **Content-Type**. If the browser wants to access other headers, you need to configure them in this parameter. For the configuration of other headers, see **Configuring CORS for a Bucket**. You can enter multiple exposed headers (one per line). Spaces and special characters including ***&:<** are not allowed.
Cache Duration (s)	Mandatory Specifies the duration that your browser can cache CORS responses, expressed in seconds. The default value is **100**.

Step 6 Click **OK**.

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

Then, only the addresses specified in **Allowed Origin** can access the OBS bucket over the methods specified in **Allowed Method**. Suppose you are configuring a

CORS rule for bucket **testbucket** by setting **Allowed Origin** to **https://www.example.com**, **Allowed Method** to **GET**, **Allowed Header** to *****, **Exposed Header** to **ETag**, and **Cache Duration (s)** to **100**. Then, only GET requests from **https://www.example.com** are allowed to access bucket **testbucket**. In addition, there are no limits on headers in a request, the ETag value can be returned in the response, and the client which the requests are from can cache the CORS response for 100 seconds.

----End

9.4 Configuring Versioning

Procedure

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

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

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

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

Step 5 Select **Enable**.

Figure 9-9 Configuring versioning

Step 6 Click **OK** to enable versioning for the bucket.

Step 7 In the navigation pane, choose **Objects**. The object list page is displayed.

Step 8 Above the search box, enable **Historical Versions** to view multiple versions of an object.

----End

Related Operations

After versioning is configured for a bucket, you can delete, share, and download object versions, and extend the retention period of an object version.

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

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

Step 3 In the object list, 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.

   ☐ NOTE

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

2. Share a version of the object by clicking **Share** in the **Operation** column.

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

   ☐ NOTE

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

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

   ----End

9.5 Configuring URL Validation

OBS blocks access requests from blacklisted URLs and allows those from whitelisted URLs.
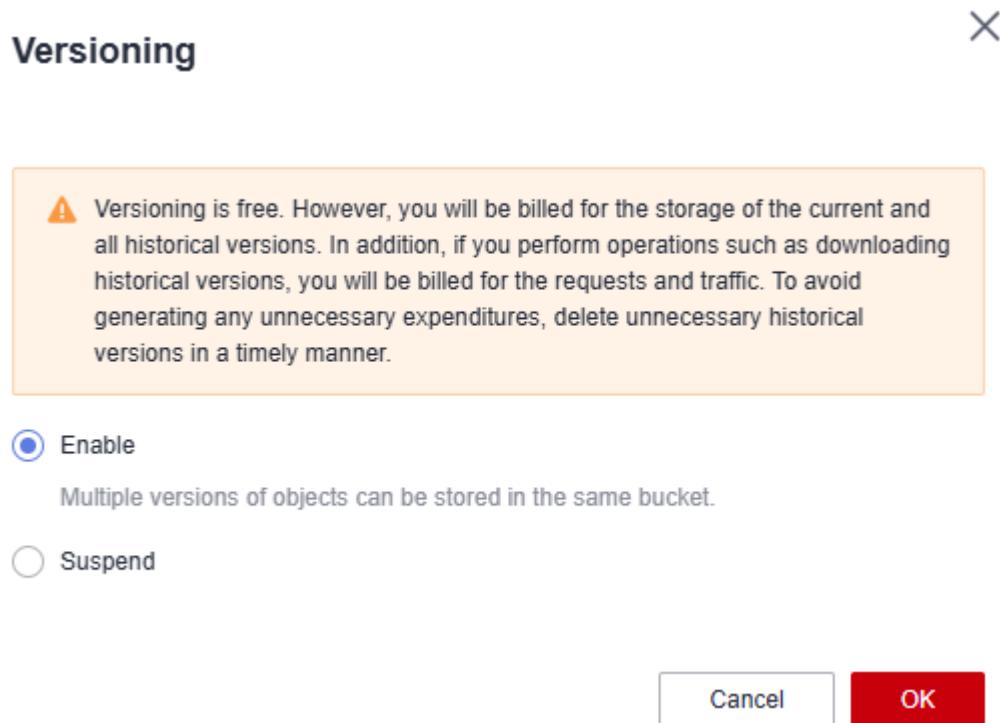
Prerequisites

Static website hosting has been enabled.

Procedure

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

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

Step 3 In the navigation pane, choose **Data Security** > **URL Validation**.

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

Principles for setting **Referers**:

- The length of a whitelist or blacklist cannot exceed 1024 characters.
- Referer format:
  - You can enter multiple referers, each in a line.
  - The referer parameter supports asterisks (*) and question marks (?). An asterisk works as a wildcard that can replace zero or multiple characters, and a question mark (?) can replace a single character.
  - If the referer header field contains **http** or **https** during download, the referer must contain **http** or **https**.
- If **Whitelisted Referers** is left blank but **Blacklisted Referers** is not, all websites except those specified in the blacklist are allowed to access data in the target bucket.
- If **Whitelisted Referers** is not left blank, only the websites specified in the whitelist are allowed to access the target bucket no matter whether **Blacklisted Referers** is left blank or not.

📖 **NOTE**

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

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

Step 5 Click ✔ to save the settings.

----End

10 Monitoring and Logging

10.1 Monitoring

10.1.1 Monitoring OBS

Scenarios

In the use of OBS, you may send PUT and GET requests that generate upload and download traffic, or receive error responses from the server. To learn the requests, traffic, and error responses in a timely manner, you can use Cloud Eye to perform automatic and real-time monitoring over your buckets.

You do not need to separately subscribe to Cloud Eye. It starts automatically once you create a resource (a bucket, for example) in OBS. For more information about Cloud Eye, see **What Is Cloud Eye?**

Figure 10-1 Cloud Eye monitoring
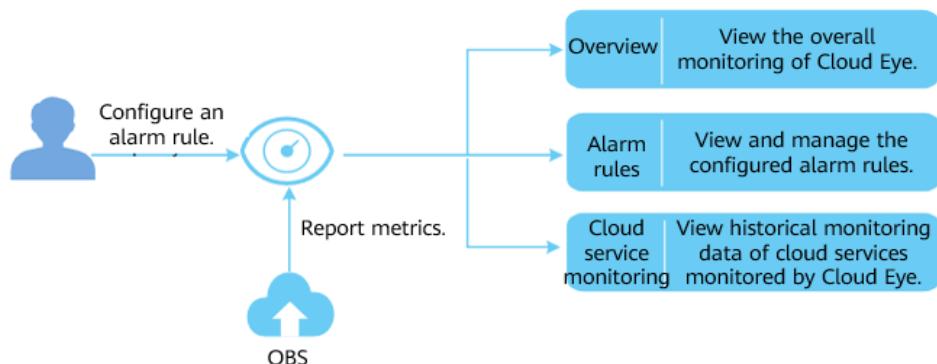
Setting Alarm Rules

In addition to automatic and real-time monitoring, you can configure alarm rules in Cloud Eye to receive alarm notifications when there are exceptions.

For details about how to configure alarm rules for monitoring over OBS, see **Creating an Alarm Rule**.

On Cloud Eye, you can configure alarm rules for events. When specified events happen, you will receive alarm notifications. For details, see **Creating an Alarm Rule and Notification for Event Monitoring**.

Viewing OBS Monitoring Metrics

Cloud Eye monitors **OBS monitoring metrics** in real time. You can view detailed monitoring statistics of each metric on the console of Cloud Eye.

For details about how to view OBS monitoring metrics, see **Querying Metrics of a Cloud Service**.

Cloud Eye monitors **OBS events** in real time. You can view the monitoring data on the Cloud Eye console. For details, see **Viewing Event Monitoring Data**.

10.1.2 Monitored OBS Metrics

Functions

This section defines the namespace, list, and dimensions of monitoring metrics reported by OBS to Cloud Eye. You can use the management console or **API** provided by Cloud Eye to search for monitoring metrics and alarms generated by OBS.

Namespace

SYS.OBS

Monitoring Metrics

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
download_bytes	Bytes Downloaded	Specifies the response bytes of all download requests made to all buckets in a region, including bytes in HTTP entity bodies. Unit: byte	≥ 0 bytes	Bucket	5 min

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
upload_bytes	Bytes Uploaded	Specifies the bytes of all upload requests made to all buckets in a region, including bytes in HTTP entity bodies. Unit: byte	≥ 0 bytes	Bucket	5 min
get_request_count	GET Requests	Specifies the number of GET, HEAD, or OPTIONS requests made to all buckets and objects in the buckets of a region. Unit: count	≥ 0 counts	Bucket	5 min
put_request_count	PUT Requests	Specifies the number of PUT, POST, and DELETE requests made to all buckets and objects in the buckets of a region. Unit: count	≥ 0 counts	Bucket	5 min
first_byte_latency	First Byte Download Delay	Specifies the average time from receiving a GET, HEAD, or OPTIONS request to the time that the system starts to respond in a measurement period. Unit: ms	≥ 0 ms	Bucket	5 min
request_count_4xx	4xx Errors	Specifies the times that the server responds to requests whose error codes are 4xx. Unit: count	≥ 0 counts	Bucket	5 min
request_count_5xx	5xx Errors	Specifies the times that the server responds to requests whose error codes are 5xx. Unit: count	≥ 0 counts	Bucket	5 min

Dimensions

Table 10-1 Dimensions

Key	Value
User	User dimension. The value is an account ID. To obtain the ID, see **Obtaining Account, IAM User, Project, User Group, Region, and Agency Information**.
Bucket_Name	Bucket dimension. The value is the bucket name.

10.1.3 Monitored OBS Events

Functions

You can configure alarm rules on Cloud Eye to monitor critical events or operation events related to your cloud resources. Once configured, Cloud Eye will notify you when the specified events happen.

Namespace

SYS.OBS

Monitored Events

Table 10-2 OBS events that can be monitored

| Event Name	Event ID	Severity	Description	Handling Suggestion	Impact
Deleting buckets	deleteBucket	Major	An event is reported when a bucket deletion takes place.	Deleted buckets cannot be recovered. If you still need to use buckets, **create a bucket** again. **CAUTION** If you want to reuse the name of a deleted bucket for a new bucket, wait at least 30 minutes after the bucket is deleted.	Deleting buckets may affect your services. Before deleting a bucket, make sure that your services do not depend on it.

Event Name	Event ID	Severity	Description	Handling Suggestion	Impact
Deleting bucket policies	deleteBucketPolicy	Major	An event is reported when a bucket policy deletion takes place.	• You can delete unwanted bucket policies. • If you delete a bucket policy by mistake, **create a bucket policy** again.	After a bucket policy is deleted, some users may fail to access the associated bucket and the objects in it.
Configuring bucket ACLs	setBucketAcl	Minor	An event is reported when a bucket ACL configuration takes place.	If you do not want an account to access a bucket or the objects in it, you can delete the bucket ACL.	A bucket ACL grants an account the access to the bucket and the objects in it.
Configuring bucket policies	setBucketPolicy	Minor	An event is reported when a bucket policy configuration takes place.	If you do not need a bucket policy to perform fine-grained access control over a bucket and the objects in it, you can delete the bucket policy.	A bucket policy grants an account some operation permissions for the bucket or the objects in it under certain conditions.

10.2 Cloud Trace Service

Cloud Trace Service (CTS) records operations on cloud resources in your account. You can use the logs to perform security analysis, track resource changes, audit compliance, and locate faults.

After you enable CTS and configure a tracker, CTS can record management and data traces of OBS for auditing.

Figure 10-2 CTS
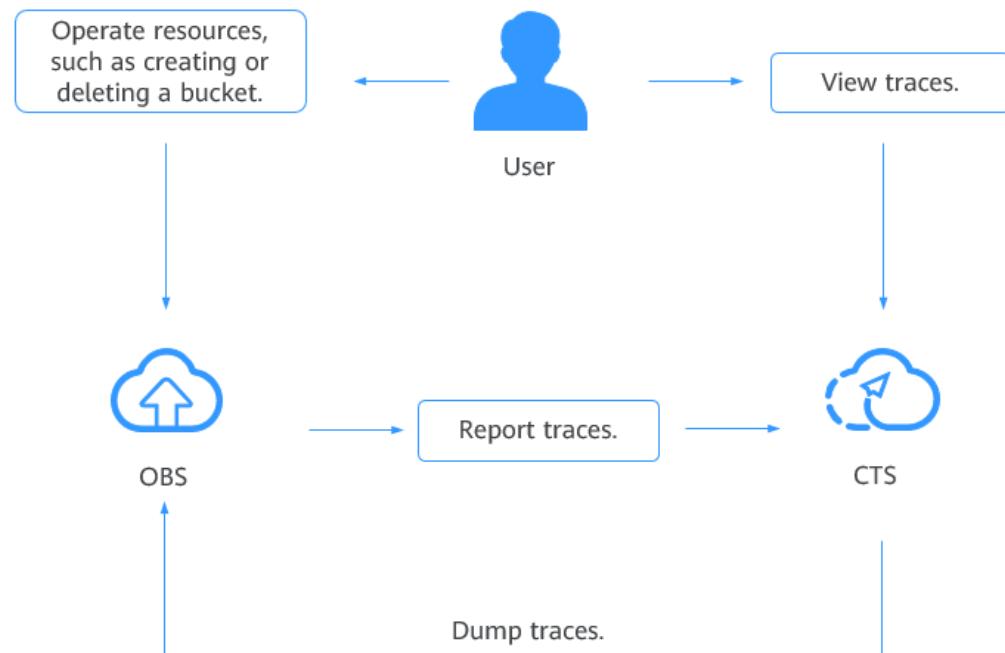
Procedure

Step 1 Log in to the management console.

Step 2 In the upper left corner of the top navigation menu, click [icon] to select a region.

Step 3 Choose **Service List** > **Management & Governance** > **Cloud Trace Service**. The **Trace List** page is displayed.

Step 4 Configure the cloud audit for OBS by referring to the tracker configuration section in the *Cloud Trace Service User Guide*.

----End

Table 10-3 OBS management operations logged by CTS

Tracker Type	Operation	Resource	Trace Name
Management	Deleting a bucket	bucket	deleteBucket
Management	Deleting the CORS configuration of a bucket	bucket	deleteBucketCors
Management	Deleting the lifecycle configuration of a bucket	bucket	deleteBucketLifecycle
Management	Deleting a bucket policy	bucket	deleteBucketPolicy

Tracker Type	Operation	Resource	Trace Name
Management	Deleting the tag configuration of a bucket	bucket	deleteBucketTagging
Management	Deleting the static website hosting configuration of a bucket	bucket	deleteBucketWebsite
Management	Creating a bucket	bucket	createBucket
Management	Configuring the bucket ACL	bucket	setBucketAcl
Management	Configuring the CORS rule for a bucket	bucket	setBucketCors
Management	Configuring the bucket lifecycle rules	bucket	setBucketLifecycle
Management	Configuring the bucket logging function	bucket	setBucketLogging
Management	Configuring the event notification function for buckets	bucket	setBucketNotification
Management	Configuring the bucket policy	bucket	setBucketPolicy
Management	Configuring the bucket quota	bucket	setBucketQuota
Management	Configuring the bucket storage class	bucket	setBucketStorageclass
Management	Configuring the bucket tag	bucket	setBucketTagging
Management	Configuring the versioning function for buckets	bucket	setBucketVersioning
Management	Configuring the static domain name for buckets	bucket	setBucketWebsite

Table 10-4 OBS data operations logged by CTS

Tracker Type	Operation	Resource	Trace Name
Data_Read	Downloading an object	object	GET.OBJECT
Data_Read	Querying the object ACL	object	GET.OBJECT.ACL
Data_Read	Querying the bucket website configuration	object	GET.OBJECT.WEBSITE
Data_Read	Accessing an object through the website	object	HEAD.OBJECT.WEBSITE
Data_Read	Querying the object metadata	object	HEAD.OBJECT
Data_Read	Listing part data	object	LIST.OBJECT.UPLOAD
Data_Write	Deleting an object	object	DELETE.OBJECT
Data_Write	Canceling a part	object	DELETE.UPLOAD
Data_Write	Queries the cross-domain requests for objects	object	OPTIONS.OBJECT
Data_Write	Uploading an object	object	POST.OBJECT
Data_Write	Deleting objects in batches	object	POST.OBJECT.MULTIDELETE
Data_Write	Restoring Archive objects	object	POST.OBJECT.RESTORE
Data_Write	Merging parts	object	POST.UPLOAD.COMPLETE
Data_Write	Initializing multipart tasks	object	POST.UPLOAD.INIT
Data_Write	Uploading an object	object	PUT.OBJECT
Data_Write	Configuring the object ACL	object	PUT.OBJECT.ACL
Data_Write	Copying an object	object	PUT.OBJECT.COPY
Data_Write	Configuring the object storage class	object	PUT.OBJECT.STORAGECLASS

Tracker Type	Operation	Resource	Trace Name
Data_Write	Uploading a part	object	PUT.PART
Data_Write	Copying a part	object	PUT.PART.COPY

Follow-up Procedure

You can click **Disable** under the **Operation** column on the right of a tracker to disable the tracker. After the tracker is disabled, the system will stop recording operations, but you can still view existing operation records.

You can click **Delete** under the **Operation** column on the right of a tracker to delete the tracker. Deleting a tracker has no impact on existing operation records. When you enable CTS again, you can view operation records that have been generated.

10.3 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.

Uploading bucket logs to the target bucket incurs billable PUT requests. For details about the pricing, see **Requests**.

Procedure

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

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

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

Step 4 In the **Basic Configurations** area, click **Logging**. The **Logging** dialog box is displayed.

Step 5 Select **Enable**. For details, see **Figure 10-3**.

Figure 10-3 Logging
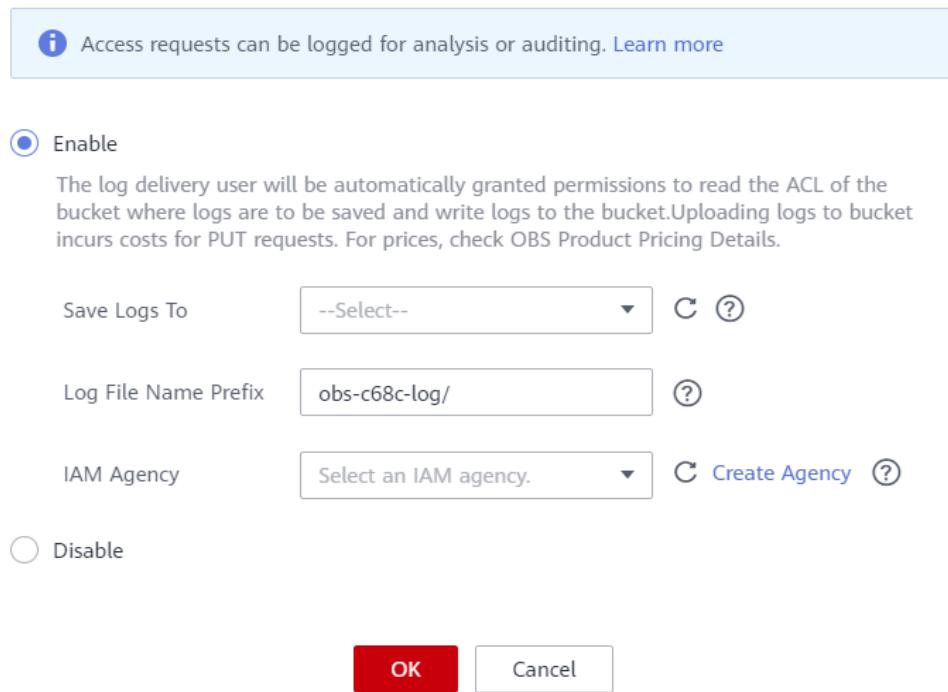
Step 6 Select an existing bucket where you want to store log files. Log delivery users of the selected bucket will be automatically granted the permissions to read the bucket ACL and write logs to the bucket.

Step 7 Enter a prefix for the **Log File Name Prefix**.

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

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

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

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

Example:

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

bucket. A log file is named as follows: **bucket-log2015-06-29-12-22-07-N7MXLAF1BDG7MPDV**.

Step 8 Select an IAM agency to grant OBS the permission to upload log files to the specified bucket.

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

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

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 9 Click **OK**.

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

----End

Related Operations

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

11 Task Center

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, restoring objects in batches, changing storage classes in batches, 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

12 Related Operations

12.1 Creating an Agency

To use some OBS features, you need to use IAM agencies to grant required permissions to OBS for processing your data.

Creating an Agency for Uploading Logs

Step 1 In the **Logging** dialog box, click **Create Agency** to jump to the **Agencies** page on the **Identity and Access Management** console.

Step 2 In the navigation pane, choose **Policy Management** > **Agency Policies**.

Step 3 Click **Create Agency**.

Step 4 Enter an agency name.

Step 5 Select **Cloud service** for the **Agency Type**.

Step 6 Select **OBS** for **Cloud Service**.

Step 7 Set a validity period.

Step 8 Click **Next**.

Step 9 On the **Select Policy/Role** page, select a custom policy that has the permission to upload data to the log storage bucket and click **Next**.

If no custom policy is available, create one by referring to **Creating a Custom Policy**.

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

☐ **NOTE**

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

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

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

Step 10 On the **Select Scope** page, select **Global services** for **Scope** and click **OK**.

Step 11 (Optional) If the log storage bucket has server-side encryption enabled, the agency also requires the **KMS Administrator** permission for the region where the bucket is located.

1. Go to the **Agencies** page of the IAM console and click the name of the agency created in the previous step.

2. Choose the **Permissions** tab and click **Authorize**.

3. On the **Select Policy/Role** page, search for and select **KMS Administrator**. Then, click **Next**.

4. On the **Select Scope** page, select **Region-specific projects** for **Scope**. Then, select the project in the region where the log storage bucket is located.

----End

Creating an Agency for Back to Source

Step 1 In the **Create Back-to-Source Rule** dialog box on OBS Console, click **View Agencies** to jump to the **Agencies** page on the IAM console.

Step 2 Click **Create Agency**.

Step 3 On the **Create Agency** page, specify an agency name, select **Cloud service** for **Agency Type**, and choose **Object Storage Service (OBS)** for **Cloud Service**.

Step 4 Select a validity period and enter a description.

Step 5 Click **Done**. In the displayed dialog box, click **Authorize**.

Step 6 If there is already a policy meeting your requirements, go to **9**. Otherwise, click **Create Policy** in the upper right corner.

Step 7 Specify a policy name and choose **Visual editor** for **Policy View**.

Step 8 Configure **Policy Content** as follows:

1. Select **Allow**.

2. For the service, select **Object Storage Service (OBS)**.

3. For actions, under **ReadOnly**, select **obs:object:GetObject**, under **Read/Write**, select **obs:object:PutObject** and **obs:object:AbortMultipartUpload**, and under **ListOnly**, select **obs:bucket:ListBucket**.

4. For resources, select **All**.

5. Click **Next**.

Step 9 Select the policy and click **Next**. Then, set **Scope** to **All resources** and click **OK**.

 NOTE

The **All resources** option means that OBS can use all resources, including those in enterprise projects, region-specific projects, and global services under the account based on assigned permissions.

----End

13 Troubleshooting

13.1 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.

13.2 OBS Console Couldn't Be Opened in Internet Explorer 9

Question

Why OBS Console cannot be opened in Internet Explorer 9, even if the address of OBS Console can be pinged?
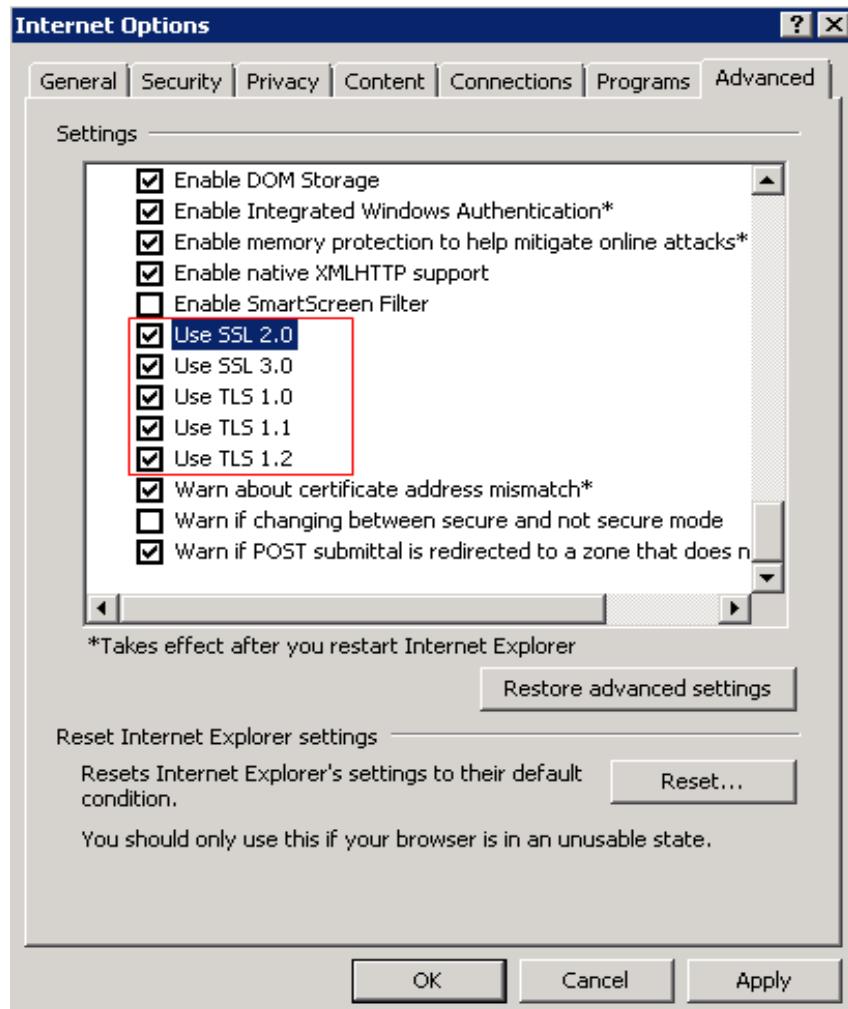
Answer

Confirm whether **Use SSL** and **Use TLS** are selected in **Internet Options**. If not, do as follows and try again:

Step 1 Open Internet Explorer 9.

Step 2 Click **Tools** in the upper right corner and choose **Internet Options** > **Advanced**. Then select **Use SSL 2.0**, **Use SSL 3.0**, **Use TLS 1.0**, **Use TLS 1.1**, and **Use TLS 1.2**, as shown in **Figure 13-1**.

Figure 13-1 Internet Options

Step 3 Click **OK**.

----End

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

Symptom

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

Cause

For Windows, a file name, including the file name extension, can contain a maximum of 255 characters.

When an object with a name containing more than 255 characters is downloaded to a local computer, the system keeps only the first 255 characters automatically.

Solution

Change the object name to a string of no more than 255 characters.

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

Symptom

Error message "Time difference is longer than 15 minutes between the client and server" or "The difference between the request time and the current time is too large" is displayed during the use of OBS.

Cause

For security purposes, OBS verifies the time offset between the client and server. If the offset is longer than 15 minutes, the OBS server will reject your requests and this error message is reported.

Solution

To resolve this problem, adjust your local time (UTC) and try again.

14 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 14-1 lists some common error codes in OBS error responses.

Table 14-1 OBS error codes

Error Code	Description
Obs.0000	Invalid parameter.
Obs.0001	All access requests to this object are invalid.
Obs.0002	The absolute path of a file cannot exceed 1023 characters. Please retry.
Obs.0003	The connection timed out.
Obs.0004	Time difference is longer than 15 minutes between the client and server. Correctly set the local time. For security purposes, OBS verifies the time offset between the client and server. If the offset is longer than 15 minutes, the OBS server will reject your requests and this error message is reported. To resolve this problem, adjust your local time (UTC) and try again.
Obs.0005	The server load is too heavy. Try again later.
Obs.0006	The number of buckets has reached the upper limit. An account (including all IAM users under this account) can create a maximum of 100 buckets and parallel file systems. You can use the fine-grained access control of OBS to properly plan and use buckets.
Obs.0007	The target bucket does not exist or is not in the same region with the current bucket.
Obs.0008	The account has not been registered with the system. Only a registered account can be used.

Error Code	Description
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. 3. Your account is in arrears or has insufficient balance. 4. Your account has been frozen.
Obs.0017	The end time of the new validity period must be later than that of the old validity period.
Obs.0018	The validity period cannot be shorter than the remaining period.
Obs.0019	Cannot determine whether the bucket has objects or fragments. Check whether you have the read permission for this bucket.
Obs.0020	TMS system error. Try again later.
Obs.0021	You do not have permissions to access TMS. Configure the required permissions in IAM.
Obs.0022	The TMS system is busy. Try again later.