

Object Storage Service

User Guide (Paris Regions)

Issue 06
Date 2024-02-29



Copyright © Huawei Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:

<https://www.huawei.com/en/psirt/vul-response-process>

For vulnerability information, enterprise customers can visit the following web page:

<https://securitybulletin.huawei.com/enterprise/en/security-advisory>

Contents

1 Service Overview.....	1
1.1 About OBS.....	1
1.2 Advantages.....	4
1.3 Application Scenarios.....	5
1.4 Permissions Management.....	9
1.5 Restrictions and Limitations.....	13
1.6 Related Services.....	16
1.7 Basic Concepts.....	16
1.7.1 Objects.....	17
1.7.2 Buckets.....	18
1.7.3 Parallel File System.....	19
1.7.4 Access Keys (AK/SK).....	19
1.7.5 Endpoints and Domain Names.....	20
1.7.6 Region and AZ.....	22
2 OBS Console Operation Guide.....	24
2.1 Console Function Overview.....	24
2.2 Restrictions.....	25
2.3 Getting Started.....	26
2.3.1 Process Description.....	26
2.3.2 Configuring User Permissions.....	27
2.3.3 Creating a Bucket.....	29
2.3.4 Uploading an Object.....	30
2.3.5 Downloading an Object.....	32
2.3.6 Deleting an Object.....	32
2.3.7 Deleting a Bucket.....	33
2.4 Storage Classes Overview.....	33
2.5 Managing Buckets.....	34
2.5.1 Creating a Bucket.....	34
2.5.2 Viewing Basic Information of a Bucket.....	36
2.5.3 Searching for a Bucket.....	37
2.5.4 Deleting a Bucket.....	38
2.6 Managing Objects.....	38
2.6.1 Creating a Folder.....	39

2.6.2 Uploading an Object.....	39
2.6.3 Downloading an Object.....	41
2.6.4 Sharing an Object.....	42
2.6.5 Searching for an Object or Folder.....	43
2.6.6 Accessing an Object Using Its URL.....	43
2.6.7 Restoring an Object from Cold Storage.....	44
2.6.8 Deleting an Object or Folder.....	45
2.6.9 Undeleting an Object.....	47
2.6.10 Managing Fragments.....	49
2.7 Server-Side Encryption.....	50
2.7.1 Server-Side Encryption Overview.....	50
2.7.2 Bucket Default Encryption.....	50
2.7.3 Uploading an Object in Server-Side Encryption Mode.....	51
2.8 Object Metadata.....	52
2.8.1 Object Metadata Overview.....	52
2.8.2 Configuring Object Metadata.....	53
2.9 Permissions Control.....	54
2.9.1 Overview.....	54
2.9.2 Permission Control Mechanisms.....	54
2.9.2.1 IAM Policies.....	54
2.9.2.2 Bucket Policies and Object Policies.....	56
2.9.2.3 Bucket ACLs and Object ACLs.....	59
2.9.2.4 Relationship Between a Bucket ACL and a Bucket Policy.....	63
2.9.2.5 How Does Authorization Work When Multiple Access Control Mechanisms Co-Exist?.....	64
2.9.3 Bucket Policy Parameters.....	65
2.9.3.1 Effect.....	65
2.9.3.2 Principals.....	66
2.9.3.3 Resources.....	66
2.9.3.4 Actions.....	67
2.9.3.5 Conditions.....	69
2.9.4 Configuring IAM Policies.....	74
2.9.4.1 Creating an IAM User and Granting OBS Permissions.....	74
2.9.5 Configuring a Bucket Policy.....	75
2.9.5.1 Configuring a Standard Bucket Policy.....	75
2.9.5.2 Configuring a Custom Bucket Policy.....	76
2.9.6 Configuring an Object Policy.....	79
2.9.7 Configuring a Bucket ACL.....	81
2.9.8 Configuring an Object ACL.....	82
2.9.9 Application Cases.....	82
2.9.9.1 Granting an IAM User Permissions to Operate a Specific Bucket.....	82
2.9.9.2 Granting Other Accounts Permissions to Operate a Specific Bucket.....	85
2.9.9.3 Restricting Access to a Bucket for Specific Addresses.....	88

2.9.9.4 Limiting the Time When Objects in a Bucket Are Accessible.....	89
2.9.9.5 Granting Anonymous Users Permission to Access Objects.....	91
2.9.9.6 Granting Anonymous Users Permission to Access Folders.....	91
2.10 Versioning.....	92
2.10.1 Versioning Overview.....	93
2.10.2 Configuring Versioning.....	96
2.11 Logging.....	96
2.11.1 Logging Overview.....	96
2.11.2 Configuring Access Logging for a Bucket.....	98
2.12 Event Notifications.....	99
2.12.1 SMN-Enabled Event Notifications.....	99
2.12.2 Configuring SMN-Enabled Event Notification.....	100
2.12.3 Application Example: Configuring SMN-Enabled Event Notification.....	103
2.13 Cross-Region Replication.....	104
2.13.1 Cross-Region Replication Overview.....	104
2.13.2 Configuring Cross-Region Replication.....	107
2.14 Lifecycle Management.....	109
2.14.1 Lifecycle Management Overview.....	109
2.14.2 Configuring a Lifecycle Rule.....	111
2.15 Configuring User-Defined Domain Names.....	113
2.15.1 Overview.....	113
2.15.2 Configuring a User-Defined Domain Name.....	114
2.16 Static Website Hosting.....	114
2.16.1 Static Website Hosting Overview.....	114
2.16.2 Redirection Overview.....	115
2.16.3 Configuring Static Website Hosting.....	115
2.16.4 Configuring Redirection.....	119
2.16.5 Using a User-Defined Domain Name to Configure Static Website Hosting.....	120
2.17 Cross-Origin Resource Sharing.....	126
2.17.1 CORS Overview.....	126
2.17.2 Configuring CORS.....	126
2.18 URL Validation.....	128
2.18.1 URL Validation Overview.....	128
2.18.2 Configuring URL Validation.....	129
2.19 Monitoring.....	130
2.19.1 Monitoring OBS.....	130
2.19.2 OBS Monitoring Metrics.....	131
2.20 Related Operations.....	132
2.20.1 Creating an IAM Agency.....	132
2.21 Troubleshooting.....	133
2.21.1 An Object Fails to Be Downloaded Using Internet Explorer 11.....	133
2.21.2 OBS Console Cannot Be Opened in Internet Explorer 9.....	134

2.21.3 The Object Name Changes After an Object with a Long Name Is Downloaded to a Local Computer	135
2.21.4 Failed to Configure Event Notifications.....	135
2.21.5 Time Difference Is Longer Than 15 Minutes Between the Client and Server.....	135
2.22 Error Code List.....	136
3 FAQ.....	138
3.1 OBS Basics.....	138
3.1.1 How Can I Get Started with OBS?.....	138
3.1.2 How Do I Obtain an OBS Endpoint?.....	138
3.1.3 What Are the Advantages of Object Storage over SAN and NAS Storage?.....	139
3.1.4 Which Types of Data Can Be Stored in OBS?.....	139
3.1.5 How Much Data Can I Store in OBS?.....	139
3.1.6 Does OBS Support Traffic Monitoring?.....	139
3.1.7 Can Folders in OBS Be Used the Same Way as in a File System?.....	141
3.1.8 Where Is Data Stored in OBS?.....	141
3.1.9 Does OBS Support Access over HTTPS?.....	141
3.1.10 Can Other Users Access My Data Stored in OBS?.....	141
3.1.11 Does OBS Support Resumable Transfer?.....	141
3.1.12 Does OBS Support Batch Upload?.....	142
3.1.13 Does OBS Support Batch Download?.....	142
3.1.14 Does OBS Support Batch Deletion of Objects?.....	143
3.1.15 What Are the Factors That Affect Upload and Download Speed of OBS?.....	143
3.1.16 Why Did Some of My Data Stored on OBS Get Lost?.....	144
3.1.17 Can Deleted Data Be Recovered?.....	144
3.1.18 Will There Be Data Left Over in OBS After I Delete an Object?.....	144
3.1.19 Will My Bucket Performance Be Affected by Other Users' Services?.....	144
3.2 Access Control.....	144
3.2.1 How Can I Control Access to OBS?.....	144
3.2.2 What Are the Differences Between Using an IAM Policy and a Bucket Policy in Access Control?....	145
3.2.3 What Is the Relationship Between a Bucket Policy and an Object Policy?.....	145
3.3 Buckets and Objects.....	145
3.3.1 Why Am I Unable to Create a Bucket?.....	145
3.3.2 Why Am I Unable to Upload an Object?.....	145
3.3.3 Why Am I Unable to Download an Object?.....	146
3.3.4 Why Can't I Delete a Bucket?.....	146
3.3.5 What Is the Relationship Between Bucket Storage Classes and Object Storage Classes?.....	146
3.3.6 Can I Modify the Region of a Bucket?.....	147
3.3.7 How Do I Obtain the Access Path to an Object?.....	147
3.3.8 Why Can't I Search for Certain Objects in My Bucket?.....	147
3.3.9 What Should I Do If an Error Message Is Displayed When I Use Internet Explorer to Access an Object URL That Contains Chinese Characters?.....	148
3.4 Tools.....	149

3.4.1 When Downloading a Folder Using obsutil, the Download Speed Slows After the Folder Download Progress Reaches 90%.....	149
3.4.2 With obsutil, Downloading a File Fails After the Download Progress Reaches 99%.....	150
3.4.3 How Do I Use the obsutil cp Command to Enable Incremental Upload, Download, or Replication?	150
3.5 APIs and SDKs.....	150
3.5.1 What Are the Differences Between PUT and POST Upload Methods?.....	150
3.5.2 Failure with OBS SDK in Uploading a File Greater than 5 GB.....	151
3.5.3 Why Don't the Signatures Match?.....	151
3.6 Security.....	153
3.6.1 How Is Data Security Ensured in OBS?.....	153
3.6.2 Does OBS Scan My Data for Other Purposes?.....	153
3.6.3 Can Engineers Export My Data from the Background of OBS?.....	153
3.6.4 How Does OBS Protect My Data from Being Stolen?.....	153
3.6.5 Can a Pair of AK and SK Be Replaced When It Is Being Used to Access OBS?.....	153
3.6.6 Can Multiple Users Share One Pair of AK and SK to Access OBS?.....	153
3.7 How Do I Use Fragment Management?.....	153
3.7.1 Why Are Fragments Generated?.....	153
3.7.2 How Do I Manage Fragments?.....	154
3.8 How Do I Use Versioning?.....	154
3.8.1 Can I Upload an Object to a Folder Where a Namesake Object Already Exists?.....	154
3.8.2 Can I Recover a Deleted Object?.....	154
3.9 Event Notification.....	154
3.9.1 Which Events Can Trigger Event Notifications?.....	154
3.10 How Do I Use Lifecycle Management?.....	155
3.10.1 What Are the Application Scenarios of Lifecycle Management?.....	155
3.11 How Do I Use Static Website Hosting?.....	155
3.11.1 Can OBS Host My Static Websites?.....	156
3.11.2 Which Types of Websites Can I Use OBS to Host?.....	156
3.11.3 How Do I Obtain the Static Website Hosting Address of a Bucket?.....	156
3.12 How Do I Use Cross-Region Replication?.....	156
3.12.1 What Are the Application Scenarios of Cross-Region Replication?.....	156
3.12.2 Will an Object Deletion in a Source Bucket Be Synchronized to the Destination Bucket?.....	156
3.12.3 Why Objects Are Not Copied to the Destination Bucket After the Cross-Region Replication Rule Has Been Created?.....	157
3.13 Server-Side Encryption.....	157
3.13.1 Does OBS Support Encrypted Upload?.....	157
3.13.2 What Encryption Technologies Can I Use to Encrypt Data on OBS?.....	158
A Change History.....	159

1 Service Overview

1.1 About OBS

OBS Overview

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

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

Product Architecture

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

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

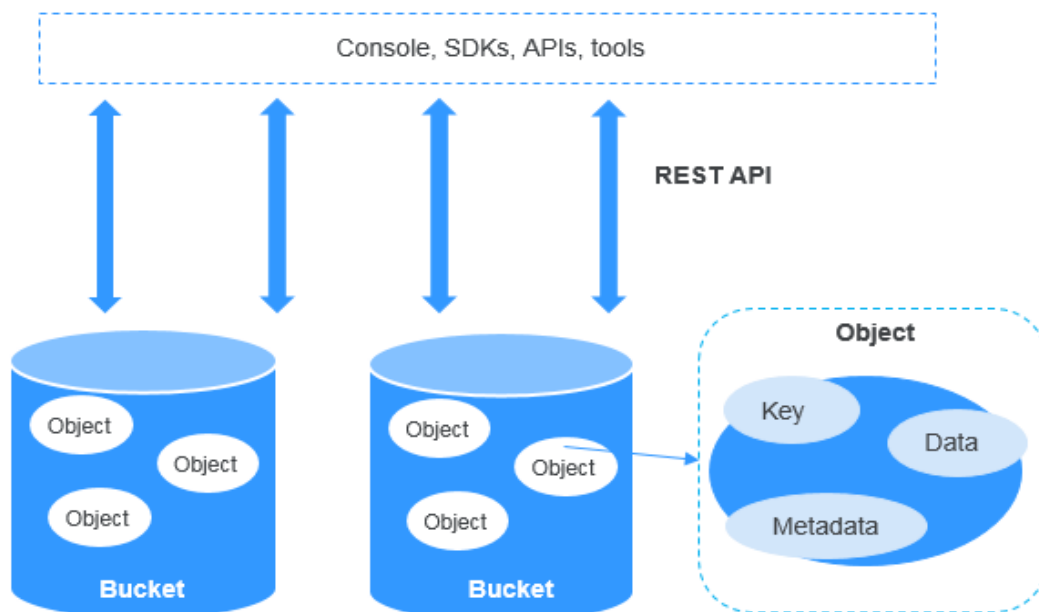
An object is the fundamental storage unit in OBS. An object consists of the following:

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

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

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

Figure 1-1 Product architecture



Storage Classes

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

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

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

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

Table 1-1 Comparison between storage classes

Compared Item	Standard	Warm	Cold
Feature	Top-notch performance, high reliability and availability	Reliable, inexpensive storage with real-time access	Long-term retention of archived data at a low cost
Application scenarios	Cloud application, data sharing, content sharing, and hot data storage	Web disk applications, enterprise backup, active archiving, and data monitoring	Archive, medical image storage, video material storage, and replacement of tape libraries
Minimum storage duration	N/A	30 days	90 days
Minimum measurement unit ^a	64 KB	64 KB	64 KB
Data restore	N/A	Billed for each GB restored.	Data can be restored at a standard or an expedited speed. Billed for each GB restored.

How to Access OBS

OBS provides various resource management tools. You can use any of the tools listed in [Table 1-2](#) to access and manage resources in OBS.

Table 1-2 OBS resource management tools

Tool	Description
OBS Console	OBS Console is a web-based GUI for you to easily manage OBS resources.
OBS Browser+	OBS Browser+ is a Windows client that lets you easily manage OBS resources from your desktop.

Tool	Description
obsutil	obsutil is a command line tool for you to perform common configuration and management operations on OBS. If you are comfortable using the command line interface (CLI), obsutil is recommended for batch processing and automated tasks.
obsfs	obsfs is an OBS tool based on Filesystem in Userspace (FUSE). It helps you mount parallel file systems to Linux, so that you can easily access virtually unlimited storage space of OBS the same way as you would use a regular local file system.
SDKs	OBS SDKs encapsulate the REST API provided by OBS to simplify development. You can call API functions provided by the OBS SDKs to enjoy OBS capabilities.

1.2 Advantages

Comparison Between OBS and On-Premises Storage Servers

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

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

Item	OBS	On-Premises Storage Server
Storage capacity	OBS provides unlimited storage capacity. All services and storage nodes are deployed in distributed clusters. You can expand each node or cluster separately, and you never have to worry about running out of space.	Such servers provide confined storage space due to the limited capacity of the hardware devices they use. When the storage space is not sufficient, you need to buy extra disks for manual expansion.

Item	OBS	On-Premises Storage Server
Security	OBS uses HTTPS and SSL protocols and encrypts data during uploads. To keep data in transit and at rest safe, OBS uses access key IDs (AKs) and secret access keys (SKs) to authenticate user identities and adopts a range of approaches including IAM policies, bucket policies, access control lists (ACLs), and uniform resource locator (URL) validation.	The owner and users are exposed to security risks from cyber attacks, technical vulnerabilities, and accidental operations.
Costs	OBS is an out-of-the-box service that has no initial capital investment or time or labor costs and frees you from O&M.	The initial deployment of on-premises servers requires high investments and a long construction period, but it quickly lags behind as enterprise businesses change so fast. Additional expenditures are required to ensure security.

OBS Advantages

- **Data durability and service continuity:** OBS supports access of hundreds of millions of users.
- **Multi-level protection and authorization management:** Measures, including versioning, server-side encryption, URL validation, virtual private cloud (VPC)-based network isolation, access log audit, and fine-grained access control are provided to keep data secure and trusted.
- **Highly concurrent access for hundreds of billions of objects:** With intelligent scheduling and response, optimized access paths, and technologies such as transmission acceleration, event notifications, and big data vertical optimization, you can store hundreds of billions of objects in OBS and still experience smooth concurrent access with ultra-high bandwidth and low latency.
- **Easy use and management:** OBS provides standard REST APIs and data migration tools to help you quickly move your workloads to cloud. Storage resources are linearly, infinitely scalable, without compromising performance. You do not have to plan storage capacity beforehand or worry about expansion or reduction.

1.3 Application Scenarios

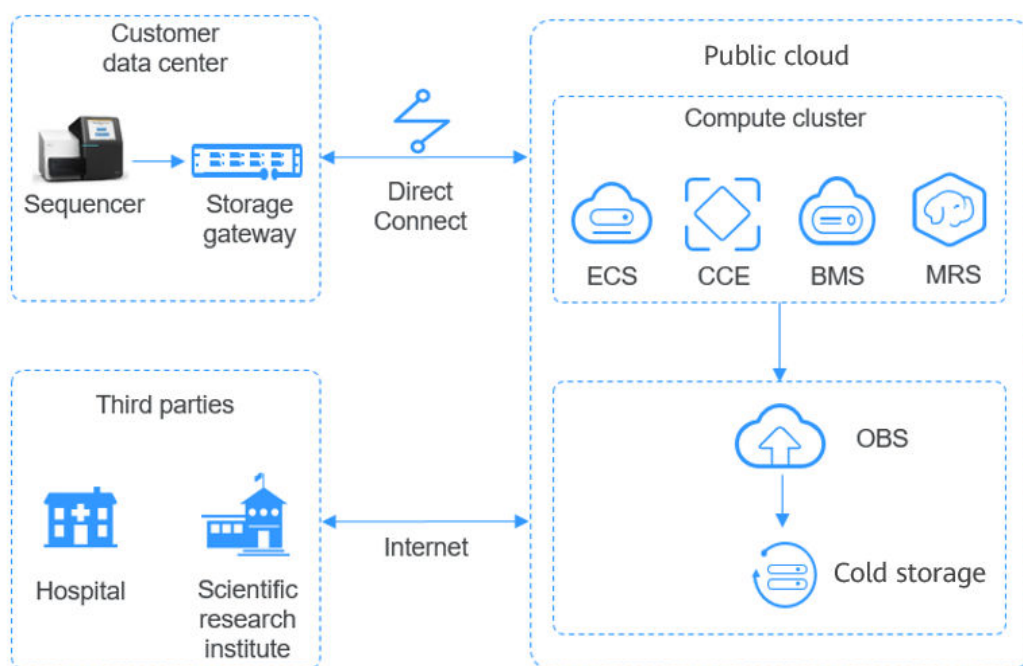
DNA Sequencing

Scenario Description

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

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

Figure 1-2 DNA sequencing



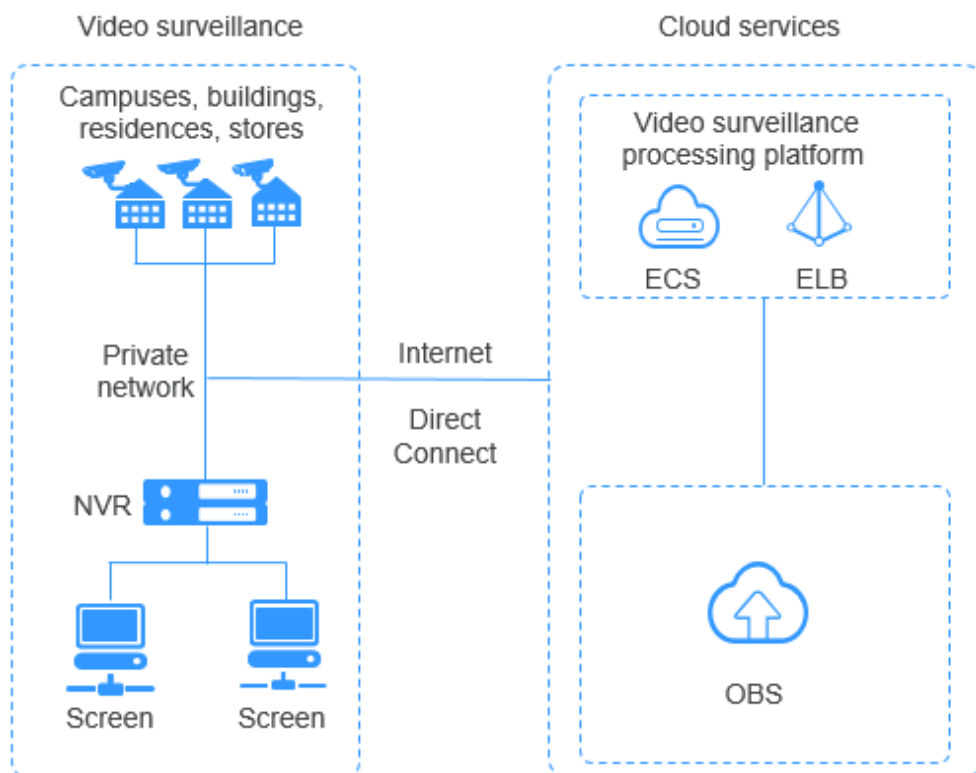
Intelligent Video Surveillance

Scenario Description

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

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

Figure 1-3 Video surveillance

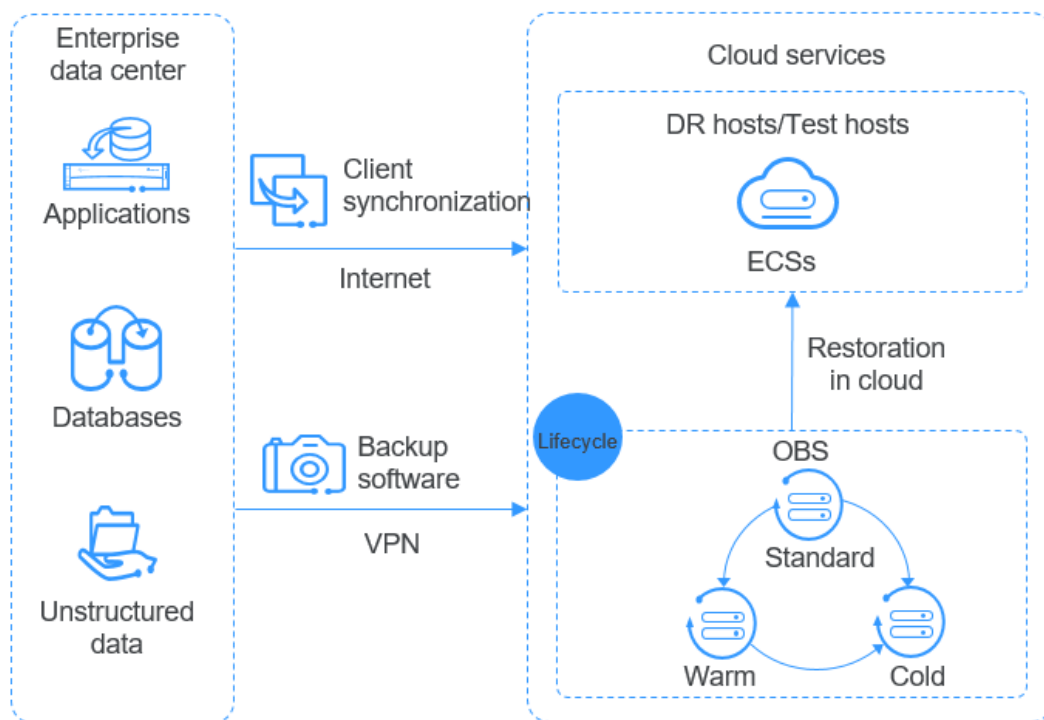


Backup and Archiving

Scenario Description

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

Figure 1-4 Backup and archiving



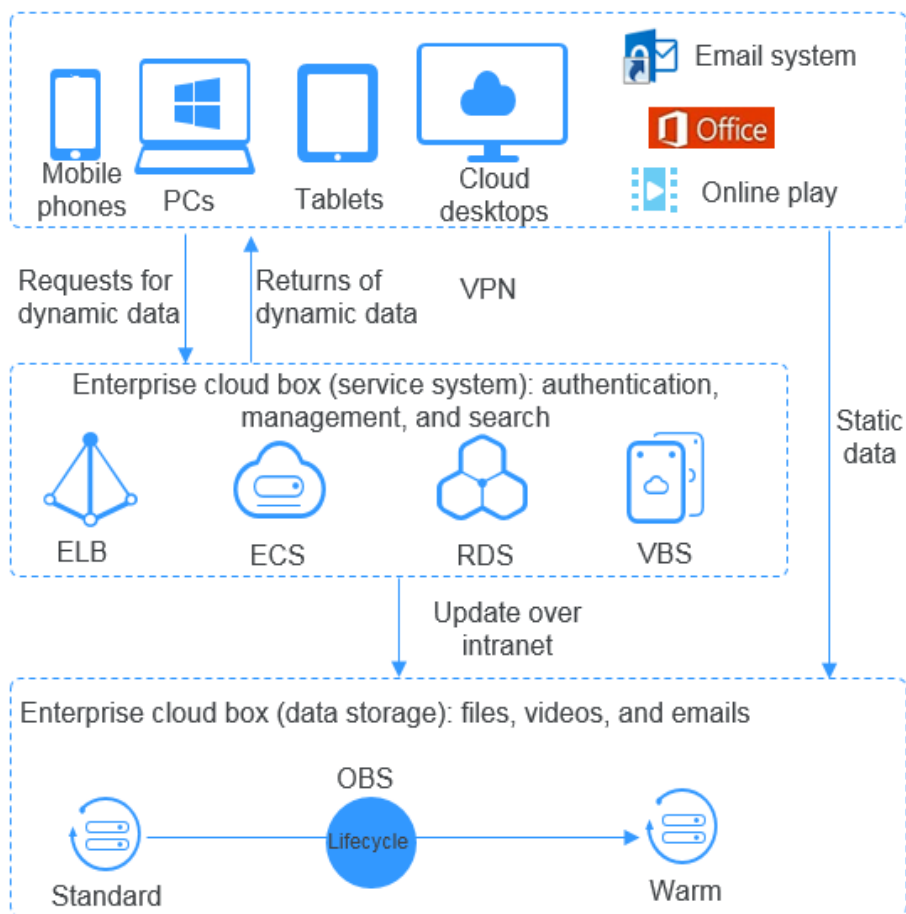
Enterprise Cloud Boxes (Web Disks)

Scenario Description

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

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

Figure 1-5 Enterprise cloud boxes (web disks)



1.4 Permissions Management

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

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

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

IAM is a free service. You only pay for the resources in your account. For more information about IAM, see section "Service Overview" in the *Identity and Access Management User Guide*.

OBS Permissions

By default, new IAM users do not have any permissions assigned. You can assign permissions to these users by adding them to one or more groups and attaching

policies to the groups. IAM provides preset system policies that define common permissions for different services, such as full control access and read-only. You can directly use these preset policies.

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

RBAC policy: An RBAC policy consists of permissions for an entire service. Users in a group with such a policy assigned are granted all the required permissions, including permissions for accessing and managing that service. RBAC policies do not support operation-specific permission control.

 **NOTE**

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

Table 1-4 lists all system policies of OBS.

Table 1-4 OBS system policies

Policy	Description	Policy Type
Tenant Administrator	Allows you to perform any operation on all cloud resources under the account. OBS policies are configured under Global service > OBS .	RBAC policy
Tenant Guest	Allows you to perform read-only operations on all cloud resources under the account. OBS policies are configured under Global service > OBS .	RBAC policy
OBS Buckets Viewer	Allows you to list buckets, and obtain basic bucket information and bucket metadata. OBS policies are configured under Global service > OBS .	RBAC policy

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

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

Operation	Tenant Administrator	Tenant Guest	OBS Buckets Viewer
Listing buckets	Yes	Yes	Yes
Creating buckets	Yes	No	No

Operation	Tenant Administrator	Tenant Guest	OBS Buckets Viewer
Deleting buckets	Yes	No	No
Obtaining basic bucket information	Yes	Yes	Yes
Controlling bucket access	Yes	No	No
Managing bucket policies	Yes	No	No
Modifying bucket storage classes	Yes	No	No
Listing objects	Yes	Yes	No
Listing objects with multiple versions	Yes	Yes	No
Uploading files	Yes	No	No
Creating folders	Yes	No	No
Deleting files	Yes	No	No
Deleting folders	Yes	No	No
Downloading files	Yes	Yes	No
Deleting files with multiple versions	Yes	No	No
Downloading files with multiple versions	Yes	Yes	No
Modifying object storage classes	Yes	No	No
Restoring files	Yes	No	No
Canceling the deletion of files	Yes	No	No
Deleting fragments	Yes	No	No
Controlling object access	Yes	No	No
Configuring object metadata	Yes	No	No

Operation	Tenant Administrator	Tenant Guest	OBS Buckets Viewer
Obtaining object metadata	Yes	Yes	No
Managing versioning	Yes	No	No
Managing logging	Yes	No	No
Managing event notifications	Yes	No	No
Managing lifecycle rules	Yes	No	No
Managing static website hosting	Yes	No	No
Managing CORS rules	Yes	No	No
Managing URL validation	Yes	No	No
Managing domain names	Yes	No	No
Managing cross-region replication	Yes	No	No
Configuring object ACL	Yes	No	No
Configuring the ACL for an object of a specified version	Yes	No	No
Obtaining object ACL information	Yes	Yes	No
Obtaining the ACL information of a specified object version	Yes	Yes	No
Uploading in the multipart mode	Yes	No	No
Listing uploaded parts	Yes	Yes	No
Canceling multipart uploads	Yes	No	No

OBS Resource Permissions Management

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

For more information, see [Overview](#).

1.5 Restrictions and Limitations

This section describes the restrictions on using OBS features.

Table 1-6 OBS use restrictions and limitations

Item	Description
Bandwidth	By default, the maximum bandwidth for read/write (GET/PUT) requests of a single account is 16 Gbit/s. If the actual bandwidth reaches the threshold, flow control will be triggered.
Queries per second (QPS)	<p>Default maximum QPS allowed by a single account:</p> <ul style="list-style-type: none"> • 6,000 write requests (PUT Object) per second • 10,000 read requests (GET Object) per second • 1,000 listing requests (LIST) per second <p>NOTE If you use sequential prefixes (such as timestamps or alphabetical order) for object naming, object access requests may be concentrated in a specific partition, resulting in access hotspots. This limits the request rate in a hotspot partition and increases access delay. Random prefixes are recommended for naming objects so that requests are evenly distributed across partitions, achieving horizontal expansion.</p>
Access rules	<p>In consideration of the DNS resolution performance and reliability, OBS requires that the bucket name must precede the domain when a request carrying a bucket name is constructed to form a three-level domain name, also mentioned as virtual-hosted-style access domain name.</p> <p>For example, you have a bucket named test-bucket in the eu-west-0 region, and you want to access the ACL of an object named test-object in the bucket. The correct URL is https://test-bucket.oss.eu-west-0.prod-cloud-ocb.orange-business.com/test-object?acl.</p>

Item	Description
Buckets	<ul style="list-style-type: none"> ● On OBS, each bucket name must be unique and cannot be changed. ● After you create a bucket, its name, storage redundancy policy, and region cannot be changed. ● An account (including all IAM users under this account) can create a maximum of 100 buckets and parallel file systems. You can use the fine-grained access control of OBS to properly plan and use buckets. ● By default, there is no limit on the storage capacity of the entire OBS system or a single bucket, and any number of objects can be stored. ● A bucket can be deleted only after all objects in the bucket have been deleted. ● The name of a deleted bucket can be reused for another bucket or a parallel file system at least 30 minutes after the deletion.
Uploading objects	<ul style="list-style-type: none"> ● OBS Console supports uploading files in a batch. A maximum of 100 files can be uploaded in a batch with the total size of no more than 5 GB. If you upload only one file in a batch upload, it cannot exceed 5 GB in size. ● If you use OBS Browser+, obsutil, an SDK, or an API, you can upload a single object of up to 48.8 TB. ● 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 file 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 namesake files in the previous folder. ● After versioning is enabled for your bucket, if the new file you upload has the same name as the one you previously uploaded to the bucket, a new file version will be added in the bucket. ● Though any UTF-8 characters can be used in object keys (object names), it is recommended that object keys be named according to the object key naming guidelines. These guidelines help object key names substantially meet the requirements of DNS, web security characters, XML analyzers, and other APIs.
Deleting objects	<p>If versioning is not enabled for a bucket, deleted objects cannot be recovered.</p>

Item	Description
Restoring Cold objects	<ul style="list-style-type: none"> ● If a Cold object is being restored, you cannot suspend or delete the restore task. ● You cannot restore an object in the Restoring state. ● After an object is restored, an object copy in the Standard storage class will be generated. This way, there is a Cold object and a Standard object copy in the bucket at the same time. The Standard object copy will be automatically deleted upon its expiration.
Lifecycle management	There is no limit on the number of lifecycle rules in a bucket, but the total size of XML descriptions about all lifecycle rules in a bucket cannot exceed 20 KB.
Cross-region replication	See Cross-Region Replication Overview .
User-defined domain name binding	<ul style="list-style-type: none"> ● Only buckets in version 3.0 support user-defined domain name binding. ● Currently, user domain names bound to OBS only allow access requests over HTTP. ● A user-defined domain name can be bound to only one bucket. ● Currently, the suffix of a user-defined domain name can contain 2 to 6 uppercase or lowercase letters.
ACLs	<ul style="list-style-type: none"> ● A bucket ACL can have up to 100 grants. The total bucket ACL size cannot exceed 50 KB. ● An object ACL can have up to 100 grants. The total object ACL size cannot exceed 50 KB.
Bucket policies	There is no limit on the number of bucket policies (statements) for a bucket, but the JSON descriptions of all bucket policies in a bucket cannot exceed 20 KB in total.
Parallel file systems	See the <i>Object Storage Service Parallel File System Feature Guide</i> .

1.6 Related Services

Table 1-7 Related services

Function	Related Service	Reference
IAM provides the following functions: <ul style="list-style-type: none"> • User identity authentication • IAM user permission control • IAM agency configuration 	Identity and Access Management (IAM)	Permissions Management Configuring User Permissions Creating an IAM Agency
Cloud Eye monitors OBS buckets, to collect statistics about the upload traffic, download traffic, the number of GET and PUT requests, the average Time to First Byte (TTFB) of GET requests, and the number of 4xx and 5xx errors.	Cloud Eye	OBS Monitoring Metrics
SMN sends OBS related alarms and event notifications, and triggers workflows.	Simple Message Notification (SMN)	SMN-Enabled Event Notifications
KMS encrypts files uploaded to the OBS.	Key Management Service (KMS)	Server-Side Encryption Overview
DNS resolves domain names configured for static website hosting in OBS.	Domain Name Service (DNS)	Using a User-Defined Domain Name to Configure Static Website Hosting

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

1.7 Basic Concepts

1.7.1 Objects

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

An object consists of the following:

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

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

When uploading an object, you can set a storage class for the object. If no storage class is specified, the object is stored in the same storage class as the bucket in which it resides. You can also change the storage class of an existing object in a bucket.

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

Object Key Naming Guidelines

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

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

Alphanumeric characters (also known as unreserved characters)	0-9, a-z, and A-Z
---	-------------------

Special characters (also known as reserved characters)	Exclamation mark (!) Hyphen (-) Underscore (_) Period (.) Asterisk (*) Single quote (') Left parenthesis (Right parenthesis (>)
--	---

The following are examples of valid object key names:

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

1.7.2 Buckets

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

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

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

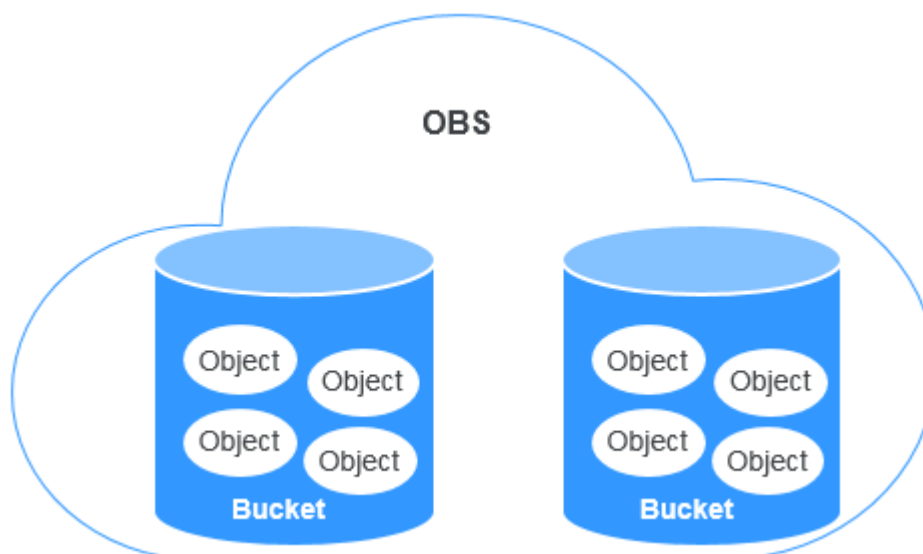
Each bucket name in OBS is globally unique and cannot be changed after the bucket has been created. The region where a bucket resides cannot be changed once the bucket is created. When you create a bucket, OBS creates a default access control list (ACL) that grants users permissions (such as read and write permissions) on the bucket. Only authorized users can perform operations such as creating, deleting, viewing, and configuring buckets.

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

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

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

Figure 1-6 Relationship between objects and buckets



1.7.3 Parallel File System

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

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

1.7.4 Access Keys (AK/SK)

OBS uses an access key ID (AK) and secret access key (SK) to authenticate the identity of a requester. When you use OBS APIs for secondary development and use the AK and SK for authentication, the signature must be calculated based on the algorithm defined by OBS and added to the request.

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

Permanent AK/SK Pair

NOTE

To access OBS in the EU-Paris region, contact the administrator to obtain the AK and SK by referring to the [access key obtaining method](#).

- Access key ID (AK): indicates the ID of the access key. It is the unique ID associated with the SK. The AK and SK are used together to obtain an encrypted signature for a request.
- Secret access key (SK): indicates the private key used together with its associated AK to cryptographically sign requests. The AK and SK are used together to identify a request sender to prevent the request from being modified.

Temporary AK/SK Pair

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

- **Temporary AK:** indicates the ID of a temporary access key. It is the unique ID associated with the SK. The AK and SK are used together to obtain an encrypted signature for a request.
- **Temporary SK:** indicates the temporary private key used together with its associated temporary AK. The AK and SK are used together to identify a request sender to prevent the request from being modified.
- **Security token:** indicates the token used together with the temporary AK and SK to access all resources of a specified account.

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

Table 1-8 OBS resource management tools

Tool	AK/SK Configuration
OBS Browser+	Configure the AK and SK during account configuration.
obsutil	Configure the AK and SK during initial configuration.
obsfs	Configure the AK and SK during initial configuration.
SDKs	Configure the AK and SK in the initialization phase.

1.7.5 Endpoints and Domain Names

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

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

Table 1-9 OBS endpoints

Region Name	Region	Endpoint	Protocol
EU-Paris	eu-west-0	oss.eu-west-0.prod-cloud-ocb.orange-business.com	HTTPS/HTTP

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

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

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

Table 1-10 OBS domain names

Type	Structure	Description	Protocol
Regional domain name	Endpoint	Each region has an endpoint, which is the domain name of the region. For more information about OBS endpoints, see Table 1-9 .	HTTPS HTTP
Bucket domain name	BucketName.Endpoint	After a bucket is created, you can use the domain name to access the bucket. You can compose the domain name according to the structure of bucket domain names, or you can obtain it from basic information of the bucket on OBS Console or OBS Browser.	HTTPS HTTP
Object domain name	BucketName.Endpoint/ObjectName	After an object is uploaded to a bucket, you can use the object domain name to access the object. You can spell out the domain name according to the structure of object domain names, or you can obtain it from the object details on OBS Console or OBS Browser. Alternatively, you can call the <code>GetObjectUrl</code> API through the SDK to obtain the object domain name.	HTTPS HTTP

Type	Structure	Description	Protocol
Static website domain name	BucketName.obs-website.Endpoint	A static website domain name is a bucket domain name when the bucket is configured to host a static website.	HTTP HTTPS
User-defined domain name	Self-owned domain name registered with a domain name provider	You can bind a user domain name to a bucket so that you can access the bucket through the user domain name.	HTTP

1.7.6 Region and AZ

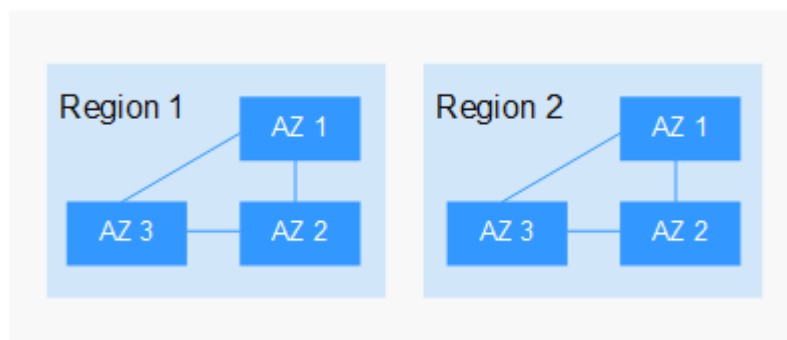
Concept

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

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

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

Figure 1-7 Regions and AZs



How Do I Select a Region?

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

How Do I Select an AZ?

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

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

Regions and Endpoints

Before using an API to call resources, you must specify its region and endpoint. For more information about OBS endpoints, see [Table 1-9](#).

2 OBS Console Operation Guide

2.1 Console Function Overview

[Table 2-1](#) lists functions provided by OBS Console.

Table 2-1 OBS Console functions

Function	Description
Basic bucket operations	Allow you to create and delete buckets of different storage classes in specified regions (service areas), as well as change bucket storage classes.
Basic object operations	Allow you to manage objects, including uploading (multipart uploads included), downloading, and deleting objects, as well as changing object storage classes and restoring Cold objects.
Server-side encryption	Encrypts objects on the server side to enhance security of objects stored on OBS.
Object metadata	Allows you to set properties for objects.
Monitoring	<ul style="list-style-type: none">• Cloud Eye can monitor the following OBS metrics:<ul style="list-style-type: none">- Download Traffic- Upload Traffic- GET Requests- PUT Requests- First Byte Download Delay- 4xx Errors- 5xx Errors
Fragment management	Manages and clears fragments generated due to object upload failures.

Function	Description
Versioning	Stores multiple versions of an object in the same bucket.
Logging	Logs bucket access requests for analysis and auditing.
Event notification	Allows you to receive messages and emails from OBS.
Permission control	Controls access to OBS using IAM policies, bucket/object policies, and bucket/object access control lists (ACLs).
Lifecycle management	Allows you to configure lifecycle rules to periodically expire and delete objects or transition objects between storage classes.
Cross-region replication	Implements object replication across regions under the same account. A cross-region replication rule enables OBS to automatically, asynchronously copy data from a source bucket in one region to a destination bucket in a different region. This provides disaster recovery across regions, catering to your needs for remote backup.
Static website hosting	Supports the hosting of static websites in buckets and the redirection of access requests for buckets.
User-defined domain name configuration	Enables you to bind your website domain name to a bucket domain name. If you want to migrate files from your website to OBS while keeping the website address unchanged, you can use this function.
URL validation	Prevents object links in OBS from being stolen by other websites.
Cross origin resource sharing (CORS)	Allows a web client in one origin to interact with resources in another one. Cross origin resource sharing (CORS) is a browser-standard mechanism defined by the World Wide Web Consortium (W3C). For general web page requests, website scripts and contents in one origin cannot interact with those in another because of Same Origin Policies (SOPs).

2.2 Restrictions

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

Table 2-2 Supported web browser versions

Web Browser	Version
Internet Explorer	<ul style="list-style-type: none">• Internet Explorer 9 (IE9)• Internet Explorer 10 (IE10)• Internet Explorer 11 (IE11)
Firefox	Firefox 55 and later
Chrome	Chrome 60 and later

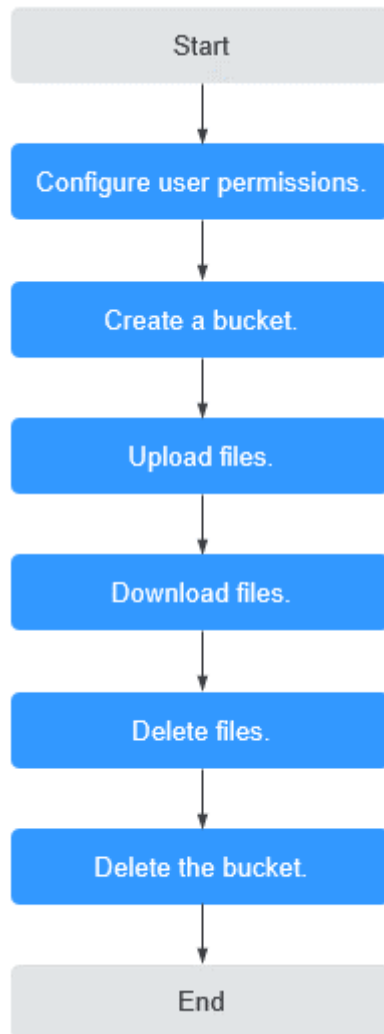
2.3 Getting Started

2.3.1 Process Description

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

The follow-up sections describe how to complete the tasks illustrated in [Figure 2-1](#).

Figure 2-1 OBS Console quick start



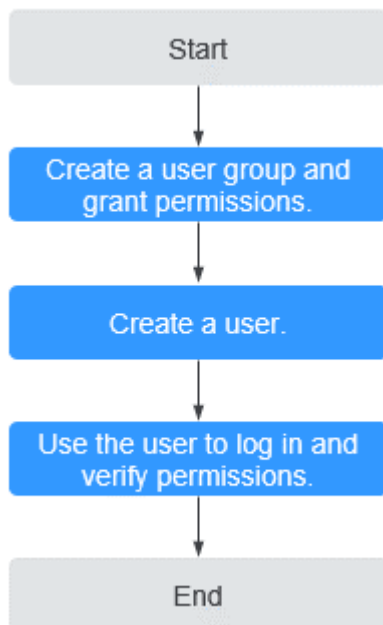
2.3.2 Configuring User Permissions

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

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

Process

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



Procedure

- Step 1** Log in to the management console with your account.
- Step 2** On the top menu bar, choose **Service List > Management & Deployment > Identity and Access Management**. The IAM console is displayed.
- Step 3** Create a user group and assign OBS permissions to it.

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

- In the navigation pane, choose **User Groups**. The **User Groups** page is displayed.
- Click **Create User Group**.
- Enter a user group name and click **OK**.
The user group is displayed in the user group list once the creation is complete.
- Locate the user group you created and click **Modify** in the **Operation** column of the row.
- In the **Group Permissions** area, locate the row that displays **Global service > OBS**, click **Attach Policy** in the **Operation** column, select the policy name, and click **OK**.

NOTE

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

- Step 4** Create an IAM user. For details, see section "Creating an IAM User" in the *Identity and Access Management User Guide*.
- Step 5** Use the created IAM user to log in to OBS Console and verify the user permissions.
- End

2.3.3 Creating a Bucket

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

 **NOTE**

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

Procedure

- Step 1** In the upper right corner of the OBS Console homepage, click **Create Bucket**.
- Step 2** Configure bucket parameters.

Table 2-3 Bucket parameters

Parameter	Description
Region	Geographic area where a bucket resides. For low latency and faster access, select the region nearest to you. Once the bucket is created, its region cannot be changed.
Bucket Name	<p>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.</p> <p>According to the globally applied DNS naming rules, an OBS bucket name:</p> <ul style="list-style-type: none"> • 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 later 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. <p>NOTE</p> <p>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.</p>

Parameter	Description
Storage Class	<p>Storage classes of a bucket. Different storage classes meet different requirements for storage performance and costs.</p> <ul style="list-style-type: none"> • The Standard storage class is for storing a large number of hot files or small files that are frequently accessed (multiple times per month on average) and require quick retrieval. • The Warm storage class is for storing data that is less frequently accessed (less than 12 times per year on average) and requires quick retrieval. • The Cold storage class is for archiving data that is rarely accessed (once a year on average) and has no requirements for quick retrieval. <p>For details, see Storage Classes Overview.</p>
Bucket Policy	<p>Controls read and write permissions for buckets.</p> <ul style="list-style-type: none"> • Private: No access beyond the bucket ACL settings is granted. • Public Read: Anyone can read objects in the bucket. • Public Read and Write: Anyone can read, write, or delete objects in the bucket.
Default Encryption	<p>After default encryption is enabled for a bucket, all objects uploaded to the bucket are automatically encrypted. The obs/default key is used by default. You can also click Create KMS Key to create a key on the KMS console. Then select the created key on OBS Console for encryption.</p> <p>If the default encryption has been enabled for the bucket, uploaded objects are automatically encrypted.</p>
Multi-AZ Mode	<p>If the multi-AZ mode is enabled, data is stored in multiple AZs.</p> <ul style="list-style-type: none"> • Once a bucket is created, its multi-AZ status cannot be changed. So, plan in advance and determine whether to enable the multi-AZ function during bucket creation. • Multi-AZ storage is not available for buckets in the Cold storage class.

Step 3 Click **Create Now**.

----End

2.3.4 Uploading an Object

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

 **NOTE**

OBS Console allows you to upload files in a batch. Up to 100 files can be uploaded at a time, with the total size of no more than 5 GB. If the file size exceeds 5 GB, but no larger than 48.8 TB, use tools (such as OBS Browser+ and obsutil) or the multipart upload of OBS SDKs and APIs for upload.

If versioning is disabled for your bucket and you upload a new file with the same name as the one you previously uploaded to your bucket, the new file automatically overwrites the previous file 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 namesake files in the previous folder.

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

Prerequisites

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

Procedure

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

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

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

 **NOTE**

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

Step 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 file** in the **Upload Object** area to select files.

Step 6 (Optional) Select **KMS encryption** to encrypt the uploaded file. For details, see [Uploading an Object in Server-Side Encryption Mode](#).

 **NOTE**

If the default encryption has been enabled for the bucket, uploaded objects are automatically encrypted.

Step 7 Click **Upload**.

----End

2.3.5 Downloading an Object

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

Limitations and Constraints

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

Procedure

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

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

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

NOTE

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

----End

2.3.6 Deleting an Object

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

Procedure

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

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

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

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

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

----End

Important Notes

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

advised to configure a [lifecycle rule](#) for directories so that they can be deleted in background based on the preset lifecycle rule.

2.3.7 Deleting a Bucket

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

Prerequisites

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

NOTICE

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

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

Procedure

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

 **NOTE**

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

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

----End

2.4 Storage Classes Overview

OBS supports tiered storage classes at the bucket level and object level.

OBS provides the following storage classes: Standard, Warm, and Cold.

These storage classes can meet different needs for storage performance and costs.

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

Bucket Storage Classes vs. Object Storage Classes

When an object is uploaded, it inherits the storage class of the bucket by default, but you can change the default storage class when you upload the object.

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

Comparison of Storage Classes

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

2.5 Managing Buckets

2.5.1 Creating a Bucket

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

NOTE

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

Procedure

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

Step 2 Configure bucket parameters.

Table 2-4 Bucket parameters

Parameter	Description
Region	Geographic area where a bucket resides. For low latency and faster access, select the region nearest to you. Once the bucket is created, its region cannot be changed.
Bucket Name	<p>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.</p> <p>According to the globally applied DNS naming rules, an OBS bucket name:</p> <ul style="list-style-type: none"> • 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 later 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. <p>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.</p>
Storage Class	<p>Storage classes of a bucket. Different storage classes meet different requirements for storage performance and costs.</p> <ul style="list-style-type: none"> • The Standard storage class is for storing a large number of hot files or small files that are frequently accessed (multiple times per month on average) and require quick retrieval. • The Warm storage class is for storing data that is less frequently accessed (less than 12 times per year on average) and requires quick retrieval. • The Cold storage class is for archiving data that is rarely accessed (once a year on average) and has no requirements for quick retrieval. <p>For details, see Storage Classes Overview.</p>
Bucket Policy	<p>Controls read and write permissions for buckets.</p> <ul style="list-style-type: none"> • 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
Default Encryption	<p>After default encryption is enabled for a bucket, all objects uploaded to the bucket are automatically encrypted. The obs/default key is used by default. You can also click Create KMS Key to create a key on the KMS console. Then select the created key on OBS Console for encryption.</p> <p>If the default encryption has been enabled for the bucket, uploaded objects are automatically encrypted.</p>
Multi-AZ Mode	<p>If the multi-AZ mode is enabled, data is stored in multiple AZs.</p> <ul style="list-style-type: none"> Once a bucket is created, its multi-AZ status cannot be changed. So, plan in advance and determine whether to enable the multi-AZ function during bucket creation. Multi-AZ storage is not available for buckets in the Cold storage class.

Step 3 Click **Create Now**.

----End

Related Operations

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

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

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

NOTE

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

----End

2.5.2 Viewing Basic Information of a Bucket

On OBS Console, you can view details about a bucket.

Procedure

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

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

Table 2-5 Parameter description

Parameter	Description
Bucket Name	Name of the bucket.
Storage Class	Storage class of the bucket, which can be Standard , Warm , or Cold .
Bucket Version	Version ID of a bucket.
Region	Region where the bucket resides.
Used Capacity	Total storage space occupied by objects of all versions in a bucket.
Objects	The total number of stored folders and objects of all versions in a bucket.
Owner	Owner refers to the account that created the bucket.
Account ID	Unique identity of the bucket owner. It is the same as Domain ID on the My Credential page.
Created	Time when the creation of a bucket is completed.
Versioning	Versioning status
Endpoint	This parameter specifies the endpoint of the region where the bucket is located. OBS provides an endpoint for each region. An endpoint is a domain name to access OBS in a region and is used to process access requests of that region.
Access Domain Name	OBS assigns each bucket with a default domain name. A domain name is the address of a bucket on the Internet. It can be used to access a bucket over the Internet in scenarios such as cloud application development and data sharing. Structure: <i>BucketName.Endpoint</i>
Multi-AZ Mode	OBS provides the feature of multiple AZs. If the multi-AZ mode is enabled, data is stored in multiple AZs.

 **NOTE**

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

----End

2.5.3 Searching for a Bucket

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


Procedure

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

Step 2 Click  .


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

For example, if you want to search for buckets whose names contain **test**, you

only need to enter **test** in the search box and click  . Then, all buckets that contain **test** in their names are displayed.

----End

Related Operations

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

2.5.4 Deleting a Bucket

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

Prerequisites

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

NOTICE

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

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

Procedure

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

 **NOTE**

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

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

----End

2.6 Managing Objects

2.6.1 Creating a Folder

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

Background Information

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

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
 - Step 2** In the navigation pane, choose **Objects**.
 - Step 3** Click **Create Folder**, or click a folder in the object list to open it and click **Create Folder**.
 - Step 4** In the **Folder Name** text box, enter a name for the folder.
 - You can create single-level or multi-level folders.
 - The name cannot contain the following special characters: \:*?"<>|
 - The name cannot start or end with a period (.) or slash (/).
 - The folder's absolute path cannot exceed 1,023 characters.
 - Any single slash (/) separates and creates multiple levels of folders at once.
 - The name cannot contain two or more consecutive slashes (/).
 - Step 5** Click **OK**.
- End

Follow-up Procedure

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

2.6.2 Uploading an Object

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

Limitations and Constraints

- OBS Console allows you to upload files in a batch. Up to 100 files can be uploaded at a time, with the total size of no more than 5 GB. If the file size exceeds 5 GB, but no larger than 48.8 TB, use tools (such as OBS Browser+ and obsutil) or the multipart upload of OBS SDKs and APIs for upload.
- If versioning is disabled for your bucket and you upload a new file with the same name as the one you previously uploaded to your bucket, the new file automatically overwrites the previous file 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 namesake files in the previous folder.
- After versioning is enabled for your bucket, if the new file you upload has the same name as the one you previously uploaded to the bucket, a new file version will be added in the bucket. For details, see [Versioning Overview](#).

Prerequisites

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

Procedure

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

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

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

NOTE

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

Step 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 file** in the **Upload Object** area to select files.

Step 6 (Optional) Select **KMS encryption** to encrypt the uploaded file. For details, see [Uploading an Object in Server-Side Encryption Mode](#).

NOTE

If the default encryption has been enabled for the bucket, uploaded objects are automatically encrypted.

Step 7 Click **Upload**.

----End

Related Operations

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

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

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

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

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

----End

NOTE

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

Follow-up Procedure

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

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

2.6.3 Downloading an Object

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

Limitations and Constraints

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

Procedure

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

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

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

 **NOTE**

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

----End

2.6.4 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.

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.

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.

Limitations and 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 from one minute to 30 days. Or, you can configure a [bucket policy or object policy](#) to grant other users access to the object permanently.
- Only buckets of version 3.0 support object sharing. You can view the bucket version in the **Basic Information** area on the **Overview** page of a bucket.
- Encrypted objects cannot be shared.
- To share a cold object, restore it first.

Related Operations

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

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

Step 3 In the **Operation** column of the file to be shared, choose **More > Copy Object URL**.

Once a temporary URL is generated, it takes effect and has a validity period of 900s. Within the validity period, anyone can use this temporary URL to access the shared file.

----End

2.6.5 Searching for an Object or Folder

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

Searching by Prefixes of Object Names

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

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


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

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

NOTE


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

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

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

----End

Related Operations

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

2.6.6 Accessing an Object Using Its URL

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

Prerequisites

Anonymous users have the read permission for the object. For details about permission granting, see [Granting Anonymous Users Permission to Access Objects](#).

 **NOTE**

Encrypted objects cannot be shared.

Procedure

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

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

Step 3 Click the object to be shared. The object information is displayed on the top part of the page. You can find the link for accessing the object in the **Link** area.

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

 **NOTE**

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

----End

2.6.7 Restoring an Object from Cold Storage

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

Limitations and Constraints

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

Procedure

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

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

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

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

 **NOTE**

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


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

Table 2-6 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. <ul style="list-style-type: none">● Expedited: Cold objects can be restored within 1 to 5 minutes.● Standard: Cold objects can be restored within 3 to 5 hours.● Bulk: Large amounts, even gigabytes, of data can be restored within 5 to 12 hours at a low cost.

Step 5 Click **OK**.

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

You can click  to manually refresh the restore status.

 **NOTE**

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

----End

Related Operations

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

 **NOTE**

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

2.6.8 Deleting an Object or Folder

Scenarios

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

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

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

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

Background Information

Object Deletion with Versioning Enabled

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

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

Procedure

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

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

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

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

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

 **CAUTION**

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

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

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

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

----End

Related Operations

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

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

2.6.9 Undeleting an Object

Scenarios

If a bucket has [versioning](#) enabled, you can recover a deleted object by undeleting it.

Background Information

Object Deletion with Versioning Enabled

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

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

Object Recovery with Versioning Enabled

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

Note the following points when you undelete objects:

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

Prerequisites

- Versioning has been enabled for the bucket. For details, see [Configuring Versioning](#).
- The file to be recovered is in the **Deleted Objects** list, and has at least one version without the **Delete Marker**.

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
- Step 2** In the navigation pane, choose **Objects**.
- Step 3** Click **Deleted Objects**.
- Step 4** In the row of the deleted object that you want to recover, click **Undelete** on the right.

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

----End

Related Operations

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

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

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

Step 3 Click **Deleted Objects**.

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

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

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

----End

2.6.10 Managing Fragments

Background Information

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

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

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

NOTICE

Generated fragments take up storage space that is billable.

Procedure

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

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

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

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

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

----End

2.7 Server-Side Encryption

2.7.1 Server-Side Encryption Overview

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

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

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

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

2.7.2 Bucket Default Encryption

OBS allows you to configure default encryption for a bucket. After the default encryption is enabled for the bucket, objects uploaded to this bucket are automatically encrypted using the specified KMS key, making data storage more secure.

You can enable default encryption when creating a bucket (see [Creating a Bucket](#)), or enable or disable default encryption after the bucket is created.

OBS only encrypts the objects uploaded after the default encryption is enabled for the bucket, and does not encrypt those uploaded before. After you disable a bucket's default encryption, the encryption status of existing objects keeps unchanged, and you can separately encrypt objects when uploading them to the bucket.

Enabling Default Encryption for a Bucket

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
- Step 2** In the **Basic Configurations** area, click **Default Encryption**. The **Default Encryption** dialog box is displayed.
- Step 3** Select **Enable**.

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

Step 4 Click **OK**.

----End

Disabling Default Encryption for a Bucket

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

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

Step 3 Select **Disable**.

Step 4 Click **OK**.

----End

2.7.3 Uploading an Object in Server-Side Encryption Mode

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

In a bucket with server-side encryption disabled, objects uploaded to it are not encrypted by default, but you can configure server-side encryption for the objects when uploading them. In a bucket with server-side encryption enabled, objects uploaded to it can inherit the encryption settings of the bucket, and you can also separately configure encryption for the objects.

Limitations and 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.
- Objects encrypted on the server side cannot be shared.

Prerequisites

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

Procedure

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

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

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

Step 4 Add the files to be uploaded.

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

 NOTE

If the default encryption has been enabled for the bucket, uploaded objects are automatically encrypted.

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

Step 6 Click **Upload**.

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

----End

2.8 Object Metadata

2.8.1 Object Metadata Overview

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

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

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

Table 2-7 OBS metadata

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

Name	Description
ContentLanguage	Indicates the language or languages intended for the audience. Therefore, a user can differentiate according to the user's preferred language. For details, see the definition about ContentLanguage in HTTP.
WebsiteRedirectLocation	<p>Redirects an object to another object or an external URL. The redirection function is implemented using static website hosting.</p> <p>For example, you can perform the following operations to implement object redirection:</p> <ol style="list-style-type: none"> 1. Set metadata of object testobject.html in the root directory of bucket testbucket. Select WebsiteRedirectLocation for Name and enter http://www.example.com for Value. <p>NOTE OBS only supports redirection for objects in the root directory of a bucket. Redirection for objects located in folders of a bucket is not supported.</p> <ol style="list-style-type: none"> 2. Configure static website hosting for bucket testbucket, and set the object testobject.html in the bucket as the default home page of the hosted static website. 3. If you access object testobject.html through the URL link provided on the Configure Static Website Hosting page, the access request is redirected to http://www.example.com.

 **NOTE**

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

2.8.2 Configuring Object Metadata

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
 - Step 2** In the navigation pane, choose **Objects**.
 - Step 3** Click the object to be operated, and then click the **Metadata** tab.
 - Step 4** Click **Add** and specify the metadata information.
 - Step 5** Click **Save**.
- End

2.9 Permissions Control

2.9.1 Overview

OBS supports the following permission control mechanisms:

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

- Bucket policies and object policies:

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

NOTE

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

An object policy applies to specified objects in a bucket.

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

2.9.2 Permission Control Mechanisms

2.9.2.1 IAM Policies

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

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

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

For details about OBS permissions controlled by IAM policies, see [Permissions Management](#).

IAM policies Application Scenarios

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

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

Policy Structure and Syntax

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

Figure 2-3 Policy structure

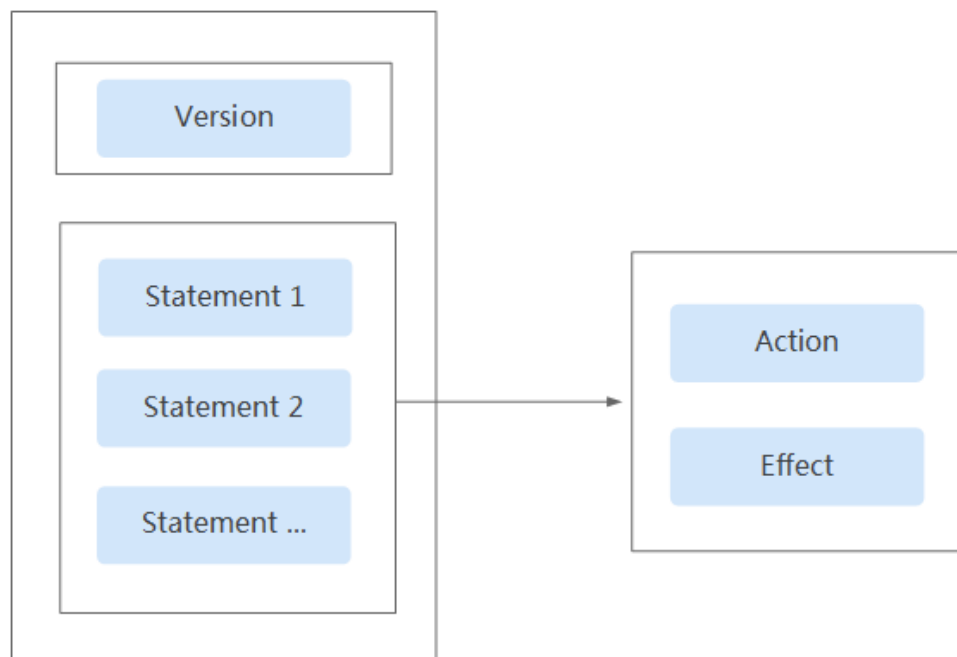


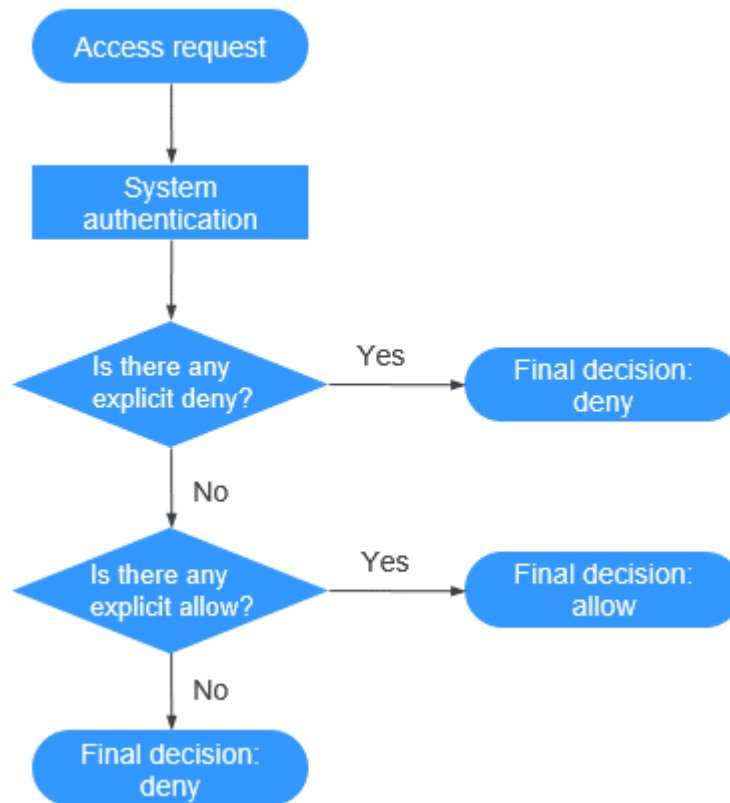
Table 2-8 Policy syntax parameters

Parameter	Description
Version	<p>The version number of a policy.</p> <ul style="list-style-type: none"> • 1.0: RBAC policies. An RBAC policy consists of permissions for an entire service. Users in a group with such a policy assigned are granted all of the permissions required for that service.
Statement	<p>Permissions defined by a policy, including Effect and Action.</p> <ul style="list-style-type: none"> • Effect The valid values for Effect are Allow and Deny. System policies contain only Allow statements. • Action Permissions of specific operations on resources. A policy can contain one or more permissions. The wildcard (*) is allowed to indicate all of the services, resource types, or operations depending on its location in the action.

Authentication of IAM policies

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

Figure 2-4 Authentication logic



NOTE

The actions in each policy are in the OR relationship.

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

2.9.2.2 Bucket Policies and Object Policies

Bucket Owner and Object Owner

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

The owner of an object is the account that uploads the object, who may not be the owner of the bucket to which the object belongs. For example, account **B** is

granted the permission to access a bucket of account **A**, and account **B** uploads a file to the bucket. In that case, instead of the bucket owner account **A**, account **B** is the owner of the object.

Bucket Policies

A bucket policy is attached to a bucket and objects in the bucket. By leveraging bucket policies, the owner of a bucket can grant IAM users or other accounts the permissions to operate the bucket and objects in the bucket.

NOTE

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

Application Scenarios

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

Standard Bucket Policies

There are three options for standard bucket policies.

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

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

Table 2-9 Standard bucket policies

Parameter	Private	Public Read	Public Read and Write
Effect	N/A	Allow	Allow
Principal	N/A	* (Any user)	* (Any user)
Resources	N/A	* (All objects in a bucket)	* (All objects in a bucket)

Parameter	Private	Public Read	Public Read and Write
Actions	N/A	<ul style="list-style-type: none"> GetObject GetObjectVersion ListBucket 	<ul style="list-style-type: none"> GetObject GetObjectVersion PutObject DeleteObject DeleteObjectVersion ListBucket
Conditions	N/A	N/A	N/A

 **NOTE**

For buckets whose version is 3.0, the default permissions of **Public Read** and **Public Read and Write** are updated to solve the problem where external buckets fail to be added to OBS Browser due to insufficient permissions.

- Added the ListBucket permission to the **Public Read** policy.
- Added the ListBucket permission to the **Public Read and Write** policy.
- If you want to add an external bucket to OBS Browser, manually update the configuration of standard bucket policies.

Custom Bucket Policies

The following three modes are provided to facilitate quick configuration:

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

 **NOTE**

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

Object Policies

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

2.9.2.3 Bucket ACLs and Object ACLs

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

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

An ACL supports up to 100 grants.

Who Is a Principal?

A principal can be an account or one of the predefined OBS groups. For details, see [Table 2-10](#).

Table 2-10 Users supported by OBS

Principal	Description
Specific User	<p>You can grant accounts access permissions to a bucket or an object using ACLs. Once a specific account is granted the access permissions, all IAM users who have OBS resource permissions under this account can have the same access permissions to operate the bucket or object.</p> <p>If you need to grant different access permissions to different IAM users, configure bucket policies. For details, see Granting an IAM User Permissions to Operate a Specific Bucket.</p>
Owner	<p>The owner of a bucket is the account that created the bucket. The bucket owner has all bucket access permissions by default. The read and write permissions for the bucket ACL are permanently available to the bucket owner, and cannot be modified.</p> <p>The owner of an object is the account that uploads the object, who may not be the owner of the bucket to which the object belongs. The object owner has the read access to the object, as well as the read and write access to the object ACL, and such access permissions cannot be modified.</p> <p>NOTICE Do not modify the bucket owner's read and write access permissions for the bucket.</p>
Anonymous User	<p>Unregistered common users group of cloud services. If anonymous users are granted access to a bucket or an object, anyone can access the object or bucket without identity authentication.</p>

Principal	Description
Registered User	A registered user refers to any account registered with the cloud services, excluding IAM users or user groups created by any account. To obtain access permissions, a registered user must be authenticated (AK and SK are used for the identity authentication). If the registered user group is granted with the write permission for a bucket, any registered and authenticated cloud service account can upload objects to the bucket, overwrite objects in the bucket, and delete objects from the bucket.
Log Delivery User NOTE Only the bucket ACL supports authorizing permissions to the log delivery user.	A log delivery user only delivers access logs of buckets and objects to the specified target bucket. OBS does not create or upload any file to a bucket automatically. Therefore, if you want to record bucket access logs, you need to grant the permission to the log delivery user who will deliver the access logs to your specified target bucket. The user only delivers logs within the service scope of OBS. NOTICE After logging is enabled, the log delivery user group will be automatically granted the permission to read the bucket ACL and write the bucket where logs are saved. If you manually disable such permissions, bucket logging fails.

What Permissions Can I Grant Using an ACL?

[Table 2-11](#) lists the permissions you can grant using a bucket ACL.

Table 2-11 Access permissions controlled by a bucket ACL

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

[Table 2-12](#) lists the permissions you can grant using an object ACL.

Table 2-12 Access permissions controlled by an object ACL

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

 **NOTE**

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

You can also set an ACL through a header when invoking the API for creating a bucket or uploading an object. Six types of predefined permissions can be set. Even with the predefined permissions configured, the bucket or object owner still has the full control over the resource. [Table 2-13](#) lists the predefined permissions.

Table 2-13 Predefined access permissions in OBS

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

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

Bucket ACL Application Scenarios

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

You can configure bucket ACLs to:

- Grant the log delivery user write access to the target bucket that stores access logs.
- Grant an account read and write access to a bucket, so that data in the bucket can be shared or the bucket can be mounted.

Object ACL Application Scenarios

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

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

- Object-level access control is required. A bucket policy can control access permissions for an object or a set of objects. If you want to further specify an access permission for an object in the set of objects for which a bucket policy has been configured, then the object ACL is recommended for easier access control over single objects.
- An object is accessed through a URL. Generally, if you want to grant anonymous users the permission to read an object through a URL, use the object ACL.

2.9.2.4 Relationship Between a Bucket ACL and a Bucket Policy

Mapping Between Bucket ACLs and Bucket Policies

Bucket ACLs are used to control basic read and write access to buckets. Custom settings of bucket policies support more actions that can be performed on buckets. Bucket policies supplement bucket ACLs. In most cases (granting permissions to log delivery user groups excluded), you can use bucket policies to manage access to buckets. [Table 2-14](#) shows the mapping between bucket ACL access permissions and bucket policy actions.

Table 2-14 Mapping between bucket ACL access permissions and bucket policy actions

ACL Permission	Option	Mapped Action in a Custom Bucket Policy
Access to bucket	Read	<ul style="list-style-type: none"> ListBucket ListBucketVersions ListBucketMultipartUploads
	Write	<ul style="list-style-type: none"> PutObject DeleteObject DeleteObjectVersion
Access to ACL	Read	GetBucketAcl
	Write	PutBucketAcl

Mapping Relationship Between Object ACLs and Bucket Policies

Object ACLs are used to control basic read and write access permissions for objects. The custom settings of bucket policies support more actions that can be performed on objects. [Table 2-15](#) describes the mapping relationship between object ACL access permissions and bucket policy actions.

Table 2-15 Mapping relationship between object ACLs and bucket policies

Object ACL	Option	Mapped Action in a Custom Bucket Policy
Access to Object	Read	<ul style="list-style-type: none"> GetObject GetObjectVersion
Access to ACL	Read	<ul style="list-style-type: none"> GetObjectAcl GetObjectVersionAcl
	Write	<ul style="list-style-type: none"> PutObjectAcl PutObjectVersionAcl

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

Based on the principle of least privilege, the default access control result is always deny, and an explicit deny statement always take precedence over an allow statement. Suppose that IAM policies grant a user the access to an object, a bucket policy denies the user's access to that object, and there is no ACL. Then user's access to the object will be denied.

If no method specifies an allow statement, then the request will be denied by default. Only if no method specifies a deny statement and one or more methods specify an allow statement, will the request be allowed. For example, if a bucket has multiple bucket policies with allow statements, the adding of a new bucket policy with an allow statement will simply add the allowed permissions to the bucket, but the adding of a new bucket policy with a deny statement will result in a re-arrangement of the permissions. The deny statement will take precedence over allowed statements, even the denied permissions are allowed in other bucket policies.

Figure 2-5 Authorization process

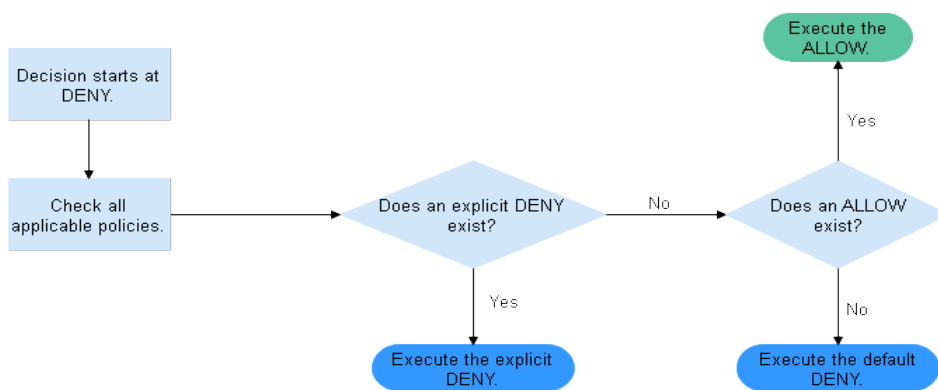


Figure 2-6 is a matrix of the IAM policies, bucket policies, and ACLs (allow and deny effects).

Figure 2-6 Matrix of the IAM policies, bucket policies, and ACLs (allow and deny effects)

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

2.9.3 Bucket Policy Parameters

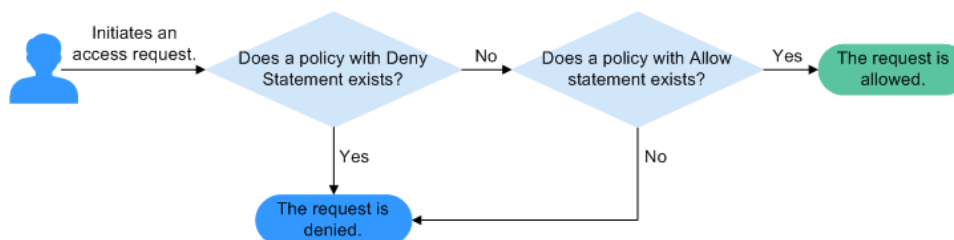
2.9.3.1 Effect

A bucket policy can either allow or deny requests.

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

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

Figure 2-7 Determining a bucket policy when the allow and deny statements conflict



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

2.9.3.2 Principals

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

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

NOTE

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

2.9.3.3 Resources

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

Resources can be specified in either of the following ways:

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

Applying a Bucket Policy to a Bucket

To specify the current bucket as the resource, keep the resource text box empty. When configuring actions for the policy, select bucket related actions.

Applying a Bucket Policy to Specified Objects

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

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

imgs-folder/example.jpg

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

NOTE

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

2.9.3.4 Actions

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

Actions can be specified in either of the following ways:

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

Actions Related to Buckets

Table 2-16 Actions related to buckets

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

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

Actions Related to Objects

Table 2-17 Actions related to objects

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

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

2.9.3.5 Conditions

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

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

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

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

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

Table 2-18 describes the predefined condition operators provided by OBS.

Table 2-18 Condition operators

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

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

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

Table 2-19 General keys

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

Table 2-20 Keys related to bucket actions

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

Action	Optional Key	Description	Description
PutBucketAcl	acl	Type: String. Configures the bucket ACL. When modifying a bucket ACL, you can use the request that contains a canned ACL setting in its header. Value options of a canned ACL setting: private public-read public-read-write authenticated-read bucket-owner-read bucket-owner-full-control log-delivery-write	None

Table 2-21 Keys related to object actions

Action	Optional Key	Description
PutObject	acl	Type: String. Configures the object ACL. When uploading an object, you can use the request that contains a canned ACL setting in its header. Value options of a canned ACL setting: private public-read public-read-write authenticated-read bucketowner-read bucket-owner-full-control log-delivery-write.
	copysource	Type: String. Specifies names of the source bucket and the source object. Format: /bucketname/keyname
	metadata-directive	Type: String. Specifies whether to copy the metadata from the source object or replace with the metadata in the request. Values: COPY REPLACE
PutObjectAcl	acl	Type: String. Configures the object ACL. When uploading an object, you can use the request that contains a canned ACL setting in its header. Value options of a canned ACL setting: private public-read public-read-write authenticated-read bucketowner-read bucket-owner-full-control log-delivery-write.
GetObjectVersion	VersionId	Type: String. Obtains the object with the specified version ID.

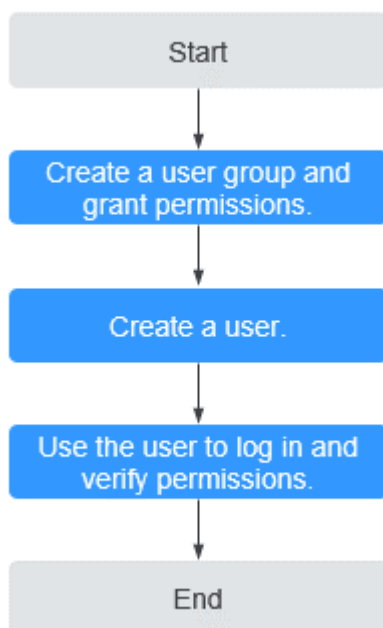
Action	Optional Key	Description
GetObjectVersionAcl	VersionId	Type: String. Obtains the ACL of the object with specified version ID.
PutObjectVersionAcl	VersionId	Type: String. Specifies a version ID.
	acl	Type: String. Configures the ACL of the object with the specified version ID. When uploading an object, you can use the request that contains a canned ACL setting in its header. Value options of a canned ACL setting: private public-read public-read-write authenticated-read bucketowner-read bucket-owner-full-control log-delivery-write.
DeleteObjectVersion	VersionId	Type: String. Deletes the object with the specified version ID.

2.9.4 Configuring IAM Policies

2.9.4.1 Creating an IAM User and Granting OBS Permissions

Process

Figure 2-8 Process of granting an IAM user the OBS permissions



Procedure

- Step 1** Log in to the management console with your account.
- Step 2** On the top menu bar, choose **Service List > Management & Deployment > Identity and Access Management**. The IAM console is displayed.
- Step 3** Create a user group and assign OBS permissions to it.

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

1. In the navigation pane, choose **User Groups**. The **User Groups** page is displayed.
2. Click **Create User Group**.
3. Enter a user group name and click **OK**.
The user group is displayed in the user group list once the creation is complete.
4. Locate the user group you created and click **Modify** in the **Operation** column of the row.
5. In the **Group Permissions** area, locate the row that displays **Global service > OBS**, click **Attach Policy** in the **Operation** column, select the policy name, and click **OK**.

NOTE

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

- Step 4** Create an IAM user. For details, see section "Creating an IAM User" in the *Identity and Access Management User Guide*.
- Step 5** Use the created IAM user to log in to OBS Console and verify the user permissions.
----End

2.9.5 Configuring a Bucket Policy

2.9.5.1 Configuring a Standard Bucket Policy

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

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
- Step 2** In the navigation pane, choose **Permissions**.
- Step 3** On the **Bucket Policies** tab page, select a policy from the **Standard Bucket Policies** area.
 - **Private**: No access beyond the bucket ACL settings is granted.

- **Public Read:** Anyone can read objects in the bucket.
- **Public Read and Write:** Anyone can read, write, or delete objects in the bucket.

 **NOTE**

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

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

----End

2.9.5.2 Configuring a Custom Bucket Policy

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

Procedure

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

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

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

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

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

 **NOTE**

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

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

Table 2-22 Parameters in bucket policies

Parameter	Value	Description
Principal	<ul style="list-style-type: none"> • Include or Exclude • Current account or Other account 	<p>Specifies users on whom this bucket policy takes effect.</p> <ul style="list-style-type: none"> • Include: The policy takes effect on specified users. • Exclude: The policy takes effect on all users except the specified ones.
Principal	<ul style="list-style-type: none"> • Include or Exclude • Current tenant or Other tenant 	<p>The person the policy is applied to.</p> <ul style="list-style-type: none"> • Include: The policy takes effect on specified users. • Exclude: The policy takes effect on all users except the specified ones.
Resources	<ul style="list-style-type: none"> • Include or Exclude • Input format: Object: <i>Object name</i> Object set: <i>Object name prefix*</i>, <i>*Object name suffix</i>, or <i>*</i> 	<p>Indicates the resource that a bucket policy applies to. With the read-only mode and read and write mode, the policy can only apply to objects.</p> <ul style="list-style-type: none"> • Include: The policy takes effect on the specified OBS resources. • Exclude: The policy takes effect on all OBS resources except the specified ones.

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

[Table 2-23](#) describes each parameter.

Table 2-23 Parameters for configuring a custom bucket policy

Parameter	Value	Description
Effect	Allow or Deny	<p>Effect of a bucket policy.</p> <ul style="list-style-type: none"> • Allow: The policy allows the matched requests. • Deny: The policy denies the matched requests.

Parameter	Value	Description
Principal	<ul style="list-style-type: none"> • Include or Exclude • Current account or Other account 	<p>Specifies users on whom this bucket policy takes effect.</p> <ul style="list-style-type: none"> • Include: The policy takes effect on specified users. • Exclude: The policy takes effect on all users except the specified ones.
Principal	<ul style="list-style-type: none"> • Include or Exclude • Current tenant or Other tenant 	<p>The person the policy is applied to.</p> <ul style="list-style-type: none"> • Include: The policy takes effect on specified users. • Exclude: The policy takes effect on all users except the specified ones.
Resources	<ul style="list-style-type: none"> • Include or Exclude • Resource input format: Object: <i>Object name</i> Object set: <i>Object name prefix*</i>, <i>*Object name suffix</i>, or <i>*</i> Blank: Indicates that the resource is the entire bucket. 	<p>Indicates the resource that a bucket policy applies to.</p> <ul style="list-style-type: none"> • Include: The policy takes effect on the specified OBS resources. • Exclude: The policy takes effect on all OBS resources except the specified ones. <p>Relationship between resource types and actions:</p> <ul style="list-style-type: none"> • When a resource is an object or an object set, only the actions related to the object can be configured. • When the resource is a bucket, only the actions related to the bucket can be configured.
Actions	<ul style="list-style-type: none"> • Include or Exclude • For details, see Actions. 	<p>Operations stated in the bucket policy.</p> <ul style="list-style-type: none"> • Include: The policy takes effect on specified actions. • Exclude: The policy takes effect on all actions except the specified ones.
Conditions	<ul style="list-style-type: none"> • Conditional Operator: See Table 2-18. • Key: See Table 2-19, Table 2-20, and Table 2-21. • Value: The entered value is associated with the key. 	<p>Conditions under which the bucket policy takes effect</p>

----End

2.9.6 Configuring an Object Policy

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

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
- Step 2** In the navigation pane, choose **Objects**.
- Step 3** On the right of the object to be operated, choose **More > Configure Object Policy**. The **Configure Object Policy** dialog box is displayed.
- Step 4** Select a proper policy mode as required. Valid options are as follows:
- **Read-only:** The authorized user has the read permission on the object. For follow-up procedure, see [Step 5](#).
 - **Read and write:** The authorized user has the read and write permissions on the object. For follow-up procedure, see [Step 5](#).
 - **Customized:** The authorized user has the customized permissions on the object. For detailed configuration, see [Step 6](#).

NOTE

You can configure only one object policy at a time.

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

Table 2-24 Object policy parameters in read-only or read and write mode

Parameter	Value	Description
Principal	<ul style="list-style-type: none"> • Include or Exclude • Current account or Other account 	Indicates the user that the object policy applies to. <ul style="list-style-type: none"> • Include: The policy applies to specified users. • Exclude: The policy applies to users except the specified ones.
Principal	<ul style="list-style-type: none"> • Include or Exclude • Current tenant or Other tenant 	The person the object policy is applied to. <ul style="list-style-type: none"> • Include: The policy applies to specified users. • Exclude: The policy applies to users except the specified ones.

Parameter	Value	Description
Resources	Include or Exclude	Resources on which the object policy takes effect. <ul style="list-style-type: none"> • Include: The bucket policy applies to specified OBS resources. • Exclude: The bucket policy applies to OBS resources except the specified ones.

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

Table 2-25 Object policy parameters in the custom mode

Parameter	Value	Description
Effect	Allow or Deny	Effect of the object policy. <ul style="list-style-type: none"> • Allow: The policy allows the matched requests. • Deny: The policy denies the matched requests.
Principal	<ul style="list-style-type: none"> • Include or Exclude • Current account or Other account 	Specifies users on whom this object policy takes effect. <ul style="list-style-type: none"> • Include: The policy applies to specified users. • Exclude: The policy applies to users except the specified ones.
Principal	<ul style="list-style-type: none"> • Include or Exclude • Current tenant or Other tenant 	The person the object policy is applied to. <ul style="list-style-type: none"> • Include: The policy applies to specified users. • Exclude: The policy applies to users except the specified ones.
Resources	<ul style="list-style-type: none"> • Include or Exclude 	Resources on which the object policy takes effect. <ul style="list-style-type: none"> • Include: The bucket policy applies to specified OBS resources. • Exclude: The bucket policy applies to OBS resources except the specified ones.

Parameter	Value	Description
Actions	<ul style="list-style-type: none"> • Include or Exclude • For details about the actions, see Actions Related to Objects. 	<p>Operation stated in the object policy.</p> <ul style="list-style-type: none"> • Include: The bucket policy applies to specified actions. • Exclude: The bucket policy applies to actions except the specified ones.
Conditions	<ul style="list-style-type: none"> • Condition Operator: See Table 2-18. • Key: See Table 2-19 and Table 2-21. • Value: The entered value is associated with the key. 	<p>Condition for an object policy to take effect.</p>

Step 7 Click **OK**.

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

----End

2.9.7 Configuring a Bucket ACL

Prerequisites

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

Procedure

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

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

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

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

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

Step 5 Click **Save**.

----End

2.9.8 Configuring an Object ACL

Prerequisites

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

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

Procedure

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

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

Step 3 Click the object to be operated.

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

NOTE

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

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

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

Step 6 Click **Save**.

----End

2.9.9 Application Cases

2.9.9.1 Granting an IAM User Permissions to Operate a Specific Bucket

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

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

Procedure

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

- Step 2** In the navigation pane, choose **Permissions**.
- Step 3** Choose **Bucket Policies > Custom Bucket Policies**.
- Step 4** Click **Create Bucket Policy**. The **Create Bucket Policy** dialog box is displayed.
- Step 5** Configure parameters listed in the table below to grant an IAM user the permission to access the bucket (to list objects in the bucket).

Table 2-26 Parameters for granting permission to access a bucket

Parameter	Value
Policy Mode	Customized
Effect	Allow
Principal	<ul style="list-style-type: none"> • Include • Select Current account and select the IAM user to be authorized.
Resources	<ul style="list-style-type: none"> • Include • Leave it blank.
Actions	<ul style="list-style-type: none"> • Include • ListBucket

- Step 6** Click **OK**.
- Step 7** Click **Create Bucket Policy**. The **Create Bucket Policy** dialog box is displayed.
- Step 8** Configure parameters in the table below to grant an IAM user the permission to upload objects to a bucket.

 **NOTE**

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

Table 2-27 Parameters for granting permission to upload objects

Parameter	Value
Policy Mode	Customized
Effect	Allow
Principal	<ul style="list-style-type: none"> • Include • Select Current account and select the IAM user to be authorized.
Resources	<ul style="list-style-type: none"> • Include • Resource name: *

Parameter	Value
Actions	<ul style="list-style-type: none"> • Include • PutObject <p>NOTE In this example, only the permission to upload objects is granted. You can also select other object actions to grant corresponding permissions if needed. The asterisk (*) indicates all actions. For details about the supported actions, see Actions.</p>

Step 9 Click **OK**.

----End


Verification

Verify the preceding permissions on OBS Browser.

Step 1 Obtain the AK and SK for the authorized IAM user from OBS Console.

Step 2 Open OBS Browser, enter the obtained AK and SK, and set the **Access Path** to the name of the authorized bucket.

Figure 2-9 Adding a new account - OBS

- Step 3** Access requests from unauthorized users are denied.
- Step 4** After being granted the permission to access the bucket, the user can access the bucket on OBS Browser, with objects in the bucket properly displayed.
- Step 5** Upload an object to the bucket. The upload fails. Click  in the upper right corner of the page. On the task management page displayed, you can see the task status is **Failed** and the failure reason is **Access denied**.
- Step 6** After being granted the permission to upload objects, the user can upload objects to the bucket on OBS Browser, with the uploaded objects properly displayed in the object list.

----End

2.9.9.2 Granting Other Accounts Permissions to Operate a Specific Bucket

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

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

NOTE

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

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

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

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
- Step 2** In the navigation pane, choose **Permissions**.
- Step 3** Choose **Bucket Policies > Custom Bucket Policies**.
- Step 4** Click **Create Bucket Policy**. The **Create Bucket Policy** dialog box is displayed.
- Step 5** Configure the parameters listed in the table below to grant other accounts bucket access permission.

Table 2-28 Parameters for granting bucket access permission

Parameter	Value
Policy Mode	Customized
Effect	Allow

Parameter	Value
Principal	<ul style="list-style-type: none"> • Include • Select Other account. Enter the account ID and user ID. <p>NOTE The account ID and user ID can be obtained on the My Credential page of the account or user to be authorized. If you grant the permission only to the account itself, IAM user IDs are not required. If you grant the permission to one or more IAM users under the account, configure both the account ID and IAM user IDs. Use commas (,) to separate multiple IAM user IDs.</p>
Resources	<ul style="list-style-type: none"> • Include • Leave it blank.
Actions	<ul style="list-style-type: none"> • Include • ListBucket

Step 6 Click **OK**.

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

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

 **NOTE**

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

Table 2-29 Parameters for granting permission to upload objects

Parameter	Value
Policy Mode	Customized
Effect	Allow

Parameter	Value
Principal	<ul style="list-style-type: none">• Include• Select Other account. Enter the account ID and user ID. <p>NOTE The account ID and user ID can be obtained on the My Credential page of the account or user to be authorized. If you grant the permission only to the account itself, IAM user IDs are not required. If you grant the permission to one or more IAM users under the account, configure both the account ID and IAM user IDs. Use commas (,) to separate multiple IAM user IDs.</p>
Resources	<ul style="list-style-type: none">• Include• Resource name: *
Actions	<ul style="list-style-type: none">• Include• PutObject

Step 9 Click **OK**.

----End

Verification

Verify the preceding permissions on OBS Browser.

Step 1 Obtain the AK and SK for the authorized IAM user from OBS Console.

Step 2 Open OBS Browser, enter the obtained AK and SK, and set the **Access Path** to the name of the authorized bucket.

Figure 2-10 Adding a new account - OBS

Add Account ✕

If you have created access keys, obtain the access key ID and its secret access key from the credentials.csv file downloaded from OBS Console. You can also click [here](#) to create a pair of access keys on the Access Keys tab.

Account Name ?


Service ?

Access Key ID ?

Secret Access Key ?

Access Path ?

Remember my secret access key

- Step 3** Access requests from unauthorized users are denied.
 - Step 4** After being granted the permission to access the bucket, the user can access the bucket on OBS Browser, with objects in the bucket properly displayed.
 - Step 5** Upload an object to the bucket. The upload fails. Click  in the upper right corner of the page. On the task management page displayed, you can see the task status is **Failed** and the failure reason is **Access denied**.
 - Step 6** After being granted the permission to upload objects, the user can upload objects to the bucket on OBS Browser, with the uploaded objects properly displayed in the object list.
- End

2.9.9.3 Restricting Access to a Bucket for Specific Addresses

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

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
- Step 2** In the navigation pane, choose **Permissions**.

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

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

Step 5 Configure parameters listed in the table below.

Table 2-30 Restricting access to a bucket for specific addresses

Parameter	Value
Policy Mode	Customized
Effect	Deny
Principal	<ul style="list-style-type: none"> ● Include > Other account ● If the account ID is set to *, the policy setting takes effect on all anonymous users. ● Leave the user ID blank.
Resources	<ul style="list-style-type: none"> ● Include ● Leave the field blank, indicating the policy takes effect on the entire bucket.
Actions	<ul style="list-style-type: none"> ● Include ● Select the asterisk (*), indicating all actions are involved.
Conditions	<ul style="list-style-type: none"> ● Conditional Operator: IpAddress ● Key: SourceIP ● Value: 114.115.1.0/24

Step 6 Click **OK**.

----End

Verification

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

2.9.9.4 Limiting the Time When Objects in a Bucket Are Accessible

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

Procedure

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

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

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

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

Step 5 Configure parameters listed in the table below.

Table 2-31 Parameters for granting permission to access a bucket

Parameter	Value
Policy Mode	Customized
Effect	Allow
Principal	<ul style="list-style-type: none"> • Include • Select Other account, and enter an asterisk (*) as the account ID, indicating all anonymous users.
Resources	<ul style="list-style-type: none"> • Include • Set the resource name to *, indicating all resources in the bucket. <p>NOTE This example only grants permissions for resources in the bucket. If you also want to grant permission for the bucket (for example, the permission to list objects in the bucket), create another custom bucket policy.</p>
Actions	<ul style="list-style-type: none"> • Include • Select * as the action name, which indicates all action permissions. <p>NOTE Selecting * may cause resources to be deleted. To avoid this risk, select Get* that indicates all read permissions.</p>
Conditions	<ul style="list-style-type: none"> • Condition Operator: DateGreaterThan • Key: CurrentTime • Value: 2019-03-26T12:00:00Z (UTC format)
Conditions	<ul style="list-style-type: none"> • Condition Operator: DateLessThan • Key: CurrentTime • Value: 2019-03-26T15:00:00Z (UTC format)

 **NOTE**

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

Step 6 Click **OK**.

----End

Verification

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

2.9.9.5 Granting Anonymous Users Permission to Access Objects

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

Procedure

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

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

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

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

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

----End

Verification

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

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

----End

2.9.9.6 Granting Anonymous Users Permission to Access Folders

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

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
- Step 2** In the navigation pane, choose **Permissions**.
- Step 3** Choose **Bucket Policies > Custom Bucket Policies**.
- Step 4** Click **Create Bucket Policy**. The **Create Bucket Policy** dialog box is displayed.
- Step 5** Configure parameters according to the following table, so that you can grant anonymous users the permission to access the folder and objects in it:

Table 2-32 Parameters for granting permission to access a bucket

Parameter	Value
Policy Mode	Customized
Effect	Allow
Principal	<ul style="list-style-type: none"> • Include • Select Other account, and enter an asterisk (*) as the account ID, indicating all anonymous users.
Resources	<ul style="list-style-type: none"> • Include • Set this parameter to all objects in the selected folder. If the folder name is folder-001, enter the value folder-001/*.
Actions	<ul style="list-style-type: none"> • Include • GetObject

- Step 6** Click **OK**.

----End

Verification

- Step 1** After the permission is successfully configured, select an object in the folder and click the object name to view its details. The object link (URL) is displayed on the details page. Share the URL over the Internet, so that all users can access or download the object through the Internet.
- Step 2** Use the URL to access the object in a browser. An anonymous user can access the object.

----End

2.10 Versioning

2.10.1 Versioning Overview

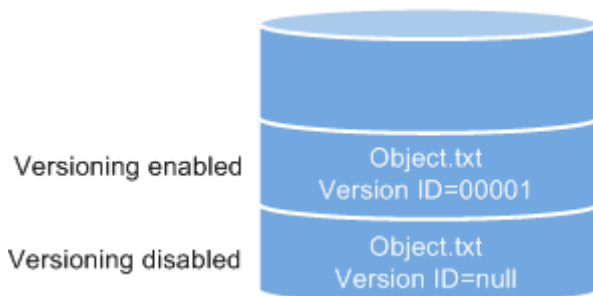
OBS can store multiple versions of an object. You can quickly search for and restore different versions or restore data in the event of accidental deletions or application faults.

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

Enabling Versioning

- Enabling versioning does not change the versions and contents of existing objects in the bucket. The version ID of an object is **null** before versioning is enabled. If a namesake object is uploaded after versioning is enabled, a version ID will be assigned to the object. For details, see [Figure 2-11](#).

Figure 2-11 Versioning (with existing objects)



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

Figure 2-12 Versioning (for new objects)

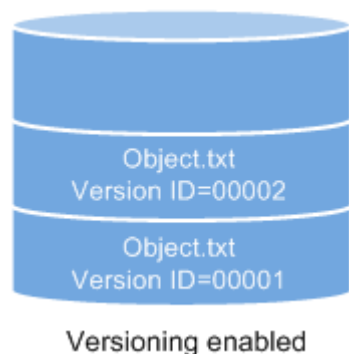


Table 2-33 Version description

Version	Description
Latest version	After versioning is enabled, each operation on an object will result in saving of the object with a new version ID. The version ID generated upon the latest operation is called the latest version.

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

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

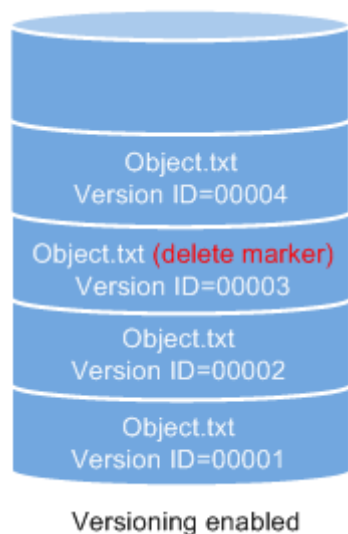
Figure 2-13 Object with a delete marker



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

For example, after object **A** is uploaded and deleted, it will be displayed in the **Deleted Objects** list. If you upload an object named **A** again, the object **A** will be displayed in the **Objects** list, and the previously deleted object **A** will no longer be displayed in the **Deleted Objects** list. For details, see [Figure 2-14](#).

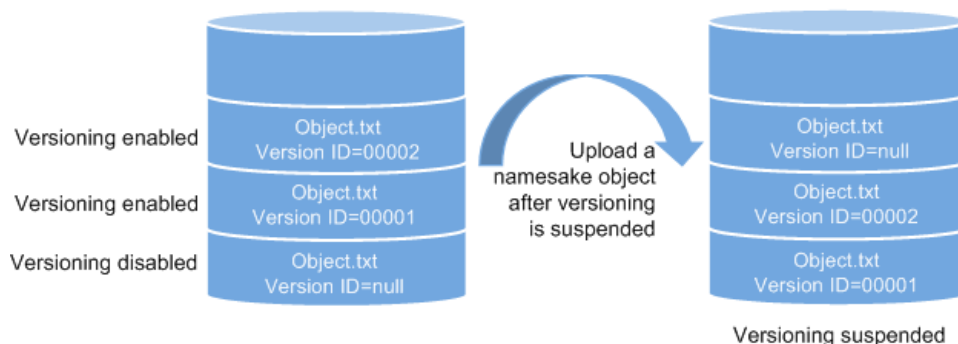
Figure 2-14 Uploading a namesake object after the original one is deleted



Suspending Versioning

Once the versioning function is enabled, it can be suspended but cannot be disabled. Once versioning is suspended, version IDs will no longer be allocated to newly uploaded objects. If an object with the same name already exists and does not have a version ID, the object will be overwritten.

Figure 2-15 Object versions in the scenario when versioning is suspended



If versions of objects in a bucket do not need to be controlled, you can suspend the versioning function.

- Historical versions will be retained in OBS. If you do not need these historical versions, manually delete them.
- Objects can be downloaded by version IDs. By default, the latest object is downloaded if the version ID is not specified.

Differences Between Scenarios When Versioning Is Suspended and Disabled

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

2.10.2 Configuring Versioning

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
- Step 2** In the **Basic Information** area, move the cursor over **Disabled**, **Suspended**, or **Enabled** next to **Versioning**. The **Edit** button is displayed next to the versioning status. Click **Edit**. The dialog box for editing the versioning status is displayed.
- Step 3** Select **Enable**.
- Step 4** Click **OK** to enable versioning for the bucket.
- Step 5** Click an object to go to the object details page. On the **Versions** tab, view all versions of the object.

----End

Related Operations

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

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
- Step 2** In the navigation pane, choose **Objects**.
- Step 3** In the object list, click the object you want to go to the object details page.
- Step 4** On the **Versions** tab, view all versions of the object.
- Step 5** Perform the following operations on object versions:

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

NOTE

- If the version you want to download is in the Cold storage class, restore it first.
2. Delete a version of the object by clicking **Delete** in the **Operation** column. If you delete the latest version, the most recent version becomes the latest version.

----End

2.11 Logging

2.11.1 Logging Overview

You can enable logging to facilitate analysis or audit as required. Access logs enable a bucket owner to analyze the property, type, or trend of requests to the bucket in depth. When the logging function of a bucket is enabled, OBS will log

access requests for the bucket automatically, and write the generated log files to the specified bucket (target bucket).

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

OBS can record bucket access requests in logs for request analysis and log audit.

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

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

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

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

The access log of each bucket contains the following information.

Table 2-34 Bucket log format

Parameter	Value Example	Description
BucketOwner	787f2f92b20943998a4fe2ab75eb09b8	Account ID of the bucket owner
Bucket	bucket	Name of the bucket
Time	[13/Aug/2015:01:43:42 +0000]	Timestamp of the request (UTC)
Remote IP	xx.xx.xx.xx	IP address from where the request is initiated
Requester	787f2f92b20943998a4fe2ab75eb09b8	Requester ID
RequestID	281599BACAD9376ECE141B842B94535B	Request ID
Operation	REST.GET.BUCKET.LOCATION	Name of the operation
Key	-	Object name
Request-URI	GET /bucket?location HTTP/1.1	URI of the request
HTTPStatus	200	Return code
ErrorCode	-	Error code

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

2.11.2 Configuring Access Logging for a Bucket

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

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
- Step 2** In the **Basic Configurations** area, click **Logging**. The **Logging** dialog box is displayed.
- Step 3** Select **Enable**.
- Step 4** Select an existing bucket where you want to store log files. Log delivery users of the selected bucket will be automatically granted the permissions to read the bucket ACL and write logs to the bucket.

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

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

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

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

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

Example:

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

Step 6 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, click **Disable** in the **Logging** dialog box and then click **OK**. After logging is disabled, logs are not recorded, but existing logs in the target bucket will be retained.

2.12 Event Notifications

2.12.1 SMN-Enabled Event Notifications

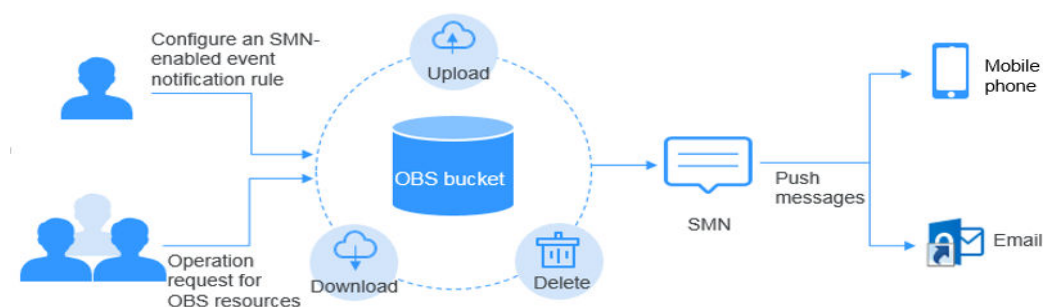
Simple Message Notification (SMN) is a reliable and extensible message notification service that can handle a huge number of messages. It significantly simplifies system coupling and can automatically push messages to endpoints via email or text message.

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

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

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

Figure 2-16 SMN-enabled event notification



2.12.2 Configuring SMN-Enabled Event Notification

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

Background Information

For details, see [SMN-Enabled Event Notifications](#).

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
- Step 2** In the **Basic Configurations** area, click **Event Notification**. The **Event Notification** page is displayed.
Alternatively, you can choose **Basic Configurations > Event Notification** in the navigation pane.
- Step 3** Click **Create**. The **Create Event Notification** dialog box is displayed.
- Step 4** Configure event notification parameters, as described in [Table 2-35](#).

Table 2-35 Event notification parameters

Parameter	Description
Name	Name of the event. If the event name is left blank, the system will automatically assign a globally unique ID.
Events	<p>Various types of events. Currently, OBS supports event notification for the following types of events:</p> <ul style="list-style-type: none"> ● ObjectCreated: Indicates all kinds of object creation operations, including PUT, POST, and COPY of objects, as well as the merging of parts. <ul style="list-style-type: none"> – Put: Creates or overwrites an object using the PUT method. – Post: Creates or overwrites an object using the POST (browser-based upload) method. – Copy: Creates or overwrites an object using the COPY method. – CompleteMultipartUpload: Merges parts of a multipart upload. ● ObjectRemoved: Deletes an object. <ul style="list-style-type: none"> – Delete: Deletes an object with a specified version ID. – DeleteMarkerCreated: Deletes an object without specifying a version ID. <p>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.</p>
Prefix	<p>Object name prefix for which notifications will be triggered.</p> <p>NOTE If neither the Prefix nor the Suffix is configured, the event notification rule applies to all objects in the bucket.</p>

Parameter	Description
Suffix	<p>Object name suffix for which notifications will be triggered.</p> <p>NOTE</p> <ul style="list-style-type: none"> • A folder path ends with a slash (/). Therefore, if you want to configure the event notification for operations on folders and you need to filter folders by suffix, the suffix must also end with a slash (/). • If neither the Prefix nor the Suffix is configured, the event notification rule applies to all objects in the bucket.
SMN Topic	<p>Project: The project that contains the SMN topic you want to select.</p> <p>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.</p>
	<p>Topic: specifies the SMN topic that authorizes OBS to publish messages. You can create such topics on the SMN management console.</p> <p>NOTE</p> <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> a. The subscriber of the topic cannot receive messages. b. Event notifications associated with unavailable topics are automatically cleared. • For details about how to use SMN, see sections "Creating a Topic", "Adding a Subscription", and "Configuring Topic Policies" in the <i>Simple Message Notification User Guide</i>.

Step 5 Click **OK**.

----End

Related Operations

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

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

2.12.3 Application Example: Configuring SMN-Enabled Event Notification

Background Information

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

Procedure


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

Step 2 Create a bucket.

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

Step 3 Create a folder.

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

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

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

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

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

For details, see [Table 2-35](#).

Step 5 Go back to OBS Console.

Step 6 Configure an event notification rule.

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

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

Table 2-36 Event notification parameters

Parameter	Value
Name	test
Events	ObjectCreated, ObjectRemoved
Prefix	SMN/ NOTE <ul style="list-style-type: none"> - A folder path ends with a slash (/). Therefore, if you want to configure the event notification for operations on folders and you need to filter folders by suffix, the suffix must also end with a slash (/). - If neither the Prefix nor the Suffix is configured, the event notification rule applies to all objects in the bucket.
Notification Method	SMN topic <i>Select the project to which the SMN topic belongs.</i> TestTopic

----End

Verification

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

Step 2 Upload the **test.txt** file to the folder created in [Step 3](#).

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

Step 3 Delete the **test.txt** file uploaded in [Step 2](#).

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

----End

2.13 Cross-Region Replication

2.13.1 Cross-Region Replication Overview

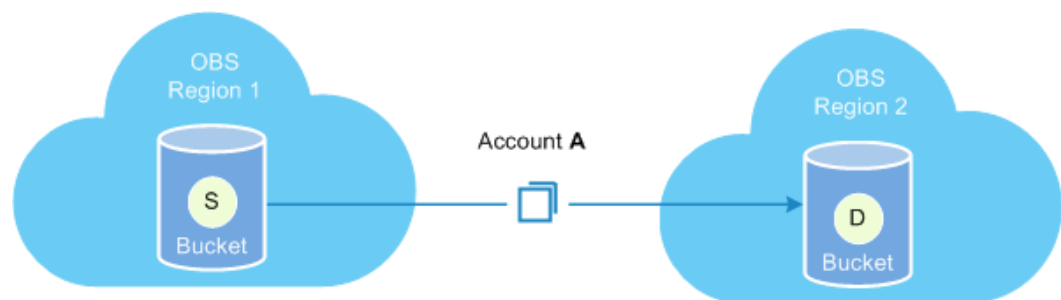
OBS offers disaster recovery across regions, catering to your needs for remote backup.

Cross-region replication refers to the process of automatically and asynchronously replicating data from a source bucket in one region to a destination bucket in another region based on the created replication rule. Source and destination

buckets must belong to the same account. Replication across accounts is currently not supported.

You can configure a rule to replicate only objects with a specified prefix or replicate all objects in a bucket. Replicated objects in the destination bucket are copies of those in the source bucket. Objects in both buckets have the same name and metadata (including the content, size, last modification time, creator, version ID, custom metadata, and ACL). By default, the storage class of the replicated object is the same as that of the source object. You can also specify a storage class for the replicated object.

Figure 2-17 Cross-region replication



Contents Replicated

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

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

Application Scenarios

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

Limitations and Constraints

Cross-region replication has the following limitations and constraints:

- Currently, only buckets of version 3.0 support cross-region replication. To check the bucket version, go to the **Overview** page of the bucket on OBS Console. Then you can view the bucket version in the **Basic Information** area.

- By default, objects uploaded before cross-region replication is enabled are not replicated to the destination bucket unless the function for synchronizing existing objects is enabled.
- The source bucket and the destination bucket must belong to different regions separately. Data cannot be copied between buckets in the same region.
- Objects of the Cold storage class in the source bucket cannot be copied to the destination bucket through the cross-region replication function.
- If the region where the destination bucket resides does not support the storage classes, object copies will be stored in the standard storage class.
- The versioning status of the source bucket must be the same as that of the destination bucket.
- Objects in a source bucket can be copied to only one destination bucket, and cannot be copied again from the destination bucket to another bucket. For example, bucket A and bucket B are in two different regions. You can copy data from bucket A to bucket B or the other way round. However, data copies in either bucket A or bucket B cannot be replicated anymore.
- Object deletion actions made on the source bucket are usually not synchronized to the destination bucket when synchronous deletion of objects is disabled. The object deletion synchronization will happen only when both the source and destination buckets have versioning enabled and you delete an object from the source bucket without specifying a version.

When synchronous deletion of objects is enabled, object deletion actions made on the source bucket will be synchronized to the destination bucket. Deleting an object from the source bucket also deletes the object from the destination bucket.

- If you change the versioning status of the destination bucket when cross-region replication is enabled, the replication of objects will fail. If you want to change the versioning status of the source bucket, disable the cross-region replication first, and then make the change.
- Ensure that owners of the source and destination buckets have the read and write permissions to the two buckets. Otherwise, data cannot be synchronized. If the system does not have the permissions to read the source bucket or write the destination bucket due to read/write permission errors, objects cannot be copied successfully, and such replication will not be resumed even if the permission error is rectified.
- For a source bucket, you can create only one cross-region replication rule that applies to the whole bucket for replication of all objects in the bucket. However, you can create a maximum of 100 cross-region replication rules based on object prefixes for the replication of objects that match the prefixes.
- OBS currently only supports the replication between one source bucket and one destination bucket. Replication from one source bucket to multiple destination buckets is not supported. The destination bucket can be modified. However, modifying the destination bucket will change the destination bucket of all existing rules.
- If you delete the OBS agency when the cross-region replication is enabled, the replication will be in the FAILED status.
- Do not delete, overwrite object replicas in the destination bucket, or modify their ACLs, which may cause inconsistency of latest object versions or

permission control settings between the destination bucket and the source bucket.

- After a replication with **Synchronize Existing Objects** enabled is complete, if the replication policy keeps unchanged, any ACL changes of source objects will be synchronized to object copies. However, ACL changes of source historical objects will not be synchronized to the copies of historical objects.

2.13.2 Configuring Cross-Region Replication

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

NOTE

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

Prerequisites

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

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
- Step 2** In the navigation pane, click **Cross-Region Replication**.
- Step 3** Click **Create Rule**. The **Create Cross-Region Replication Rule** dialog box is displayed.
- Step 4** Configure a cross-region replication rule according to your service needs. For details about the parameters, see [Table 2-37](#).

Table 2-37 Cross-region replication parameters

Parameter		Description
Status		Indicates whether the rule is enabled or disabled after being created. The versioning status of the source and destination buckets must keep the same.
Source bucket	Replicate	Indicates the objects the rule will apply to. <ul style="list-style-type: none"> • All objects: The rule applies to all objects in the bucket. • Match by prefix: The rule applies only to objects with the specified prefix.

Parameter		Description
	Prefix	<ul style="list-style-type: none"> To apply the rule to objects with the specified prefix, you must set Prefix to a value no longer than 1,024 characters. If the specified prefix overlaps with the prefix of an existing rule, OBS regards these two rules as one and the new rule cannot be configured. For example, if there is already a rule with prefix abc in OBS, you cannot configure another rule whose prefix starts with abc. To copy a folder, end the prefix with a slash (/), for example, imgs/.
	Synchronize Existing Objects	Indicates whether to synchronize the objects that were already in the bucket before the rule configuration to the destination bucket. By default, these objects are not synchronized.
	Synchronize Deleting Action	Indicates whether to synchronize the object deletions in the source bucket to the destination bucket. With this function enabled, deleting an object from the source bucket will also delete the object copy from the destination bucket.
	Replicate KMS encrypted objects	<p>OBS will try to copy KMS encrypted objects no matter whether this option is selected or not.</p> <ul style="list-style-type: none"> If this option is selected, only the IAM agencies that have the KMS Administrator permission at both the source and destination ends are displayed in the drop-down list of IAM Agency in the Create Cross-Region Replication Rule dialog box. If this option is not selected, only the IAM agencies that do not have the KMS Administrator permission at either the source or destination end are displayed in the drop-down list of IAM Agency in the Create Cross-Region Replication Rule dialog box. <p>If KMS is not available in the destination region or the agency does not have the KMS Administrator permission in the source and destination regions, KMS encrypted objects will fail to be replicated to the destination bucket, and the object replication status will be failed.</p> <p>After a KMS-encrypted object is replicated to the destination bucket, the key for encrypting the object copy changes to the default key obs/default of the destination region.</p>

Parameter		Description
Destination bucket	Region	Indicates the region of the destination bucket. The destination and source buckets must be in different regions.
	Bucket	Indicates the destination bucket.
	Change storage class for replicated objects	By default, this option is not selected, indicating that the storage class of object copies is the same as that of the source objects. If you need to change the storage class of objects copies, select this parameter, then you can specify a storage class.
Permissions	IAM Agency	<p>Delegates OBS to operate your resources, so that OBS can use this agency to implement cross-region replication.</p> <p>If there is no IAM agency available, click Create IAM agencies to create one. If you have already created IAM agencies, select one from the drop-down list.</p> <p>NOTE Requirements: The IAM agency must be of OBS. The OBS project must have the OBS FullAccess permissions. If Replicate KMS encrypted objects is selected, you also need the KMS Administrator permission in the regions where the source and destination buckets are located.</p>

Step 5 (Optional) Create an IAM Agency. For details, see [Creating an IAM Agency](#).

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

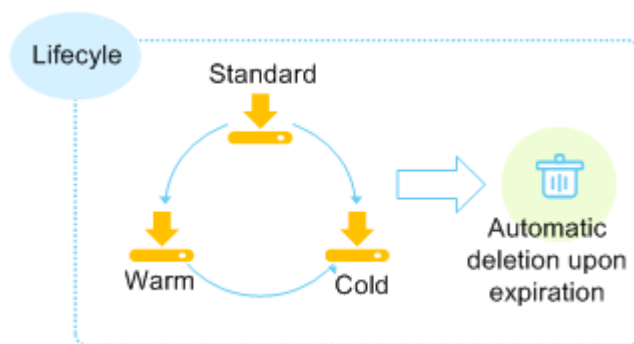
----End

2.14 Lifecycle Management

2.14.1 Lifecycle Management Overview

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

Figure 2-18 Lifecycle management



You may configure lifecycle rules to:

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

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

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

Lifecycle rules have the following key elements:

- Policy
You can specify an object name prefix to apply a lifecycle rule to a set of objects. You can also apply a lifecycle rule to the entire bucket (including the objects in it).
- Time
You can specify the number of days after which objects that have been last updated and meet specified conditions are automatically transitioned to Warm or Cold, or are expired and then deleted.
 - Transition to Warm: This defines the number of days since the last object update after which objects meeting specified conditions are automatically transitioned to the Warm storage class.
 - Transition to Cold: This defines the number of days since the last object update after which objects meeting specified conditions are automatically transitioned to the Cold storage class.
 - Expiration time: This defines the number of days since the last object update after which objects meeting specified conditions are automatically expired and then deleted.

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

can be transitioned to Cold at least 63 days after their last update. If only transition to Cold is used, but transition to Warm is not, there is no limit on the number of days for transition. The number set for expiration time must be larger than that specified for any of the transition operations.

2.14.2 Configuring a Lifecycle Rule

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

- Transition objects from Standard to Warm or Cold.
- Transition objects from Warm to Cold.
- Expire objects and then delete them.

Lifecycle rules do not transition Cold objects to other storage classes.

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
- Step 2** In the **Basic Configurations** area, click **Lifecycle Rules**. The **Lifecycle Rules** page is displayed.

Alternatively, you can choose **Basic Configurations > Lifecycle Rules** in the navigation pane.
- Step 3** Click **Create**.
- Step 4** Configure a lifecycle rule.

Basic Information:

- **Status:**
Select **Enable** to enable the lifecycle rule.
- **Rule Name:**
It identifies a lifecycle rule. A rule name can contain a maximum of 255 characters.
- **Applies To:** Can be set to **Object name prefix** or **Bucket**.
 - **Object name prefix:** Objects with this specified prefix will be managed by the lifecycle rule. The prefix cannot start with a slash (/) or contain two consecutive slashes (//), and cannot contain the following special characters: \:*?"<>|
 - **Bucket:** All objects in the bucket will be managed by the lifecycle rule.

NOTE

- If the specified prefix is overlapping with the prefix set in an existing lifecycle rule, OBS regards these two rules as one and forbids you to configure the one you are configuring. For example, if there is already a rule with prefix **abc** in OBS, you cannot configure another rule whose prefix starts with **abc**.
- If there is already a lifecycle rule based on an object prefix, you are not allowed to configure another rule that is applied to the entire bucket.
- If a lifecycle rule has been configured for the entire bucket, no more rules that apply to object name prefix can be added.

Current Version or Historical Version:

 NOTE

- **Current Version** and **Historical Version** are two concepts for versioning. If versioning is enabled for a bucket, uploading objects with the same name to the bucket creates different object versions. The last uploaded object is called the current version, while those previously uploaded are called historical versions.
- You can configure either the **Current Version** or **Historical Version**, or both of them.
- **Transition to Warm:** After this number of days since the last update, objects meeting specified conditions will be transitioned to Warm. This number must be at least 30.
- **Transition to Cold:** After this number of days since the last update, objects meeting specified conditions will be transitioned to Cold. If you configure to transition objects first to Warm and then Cold, the objects must stay Warm at least 30 days before they can be transitioned to Cold. If only transition to Cold is used, but transition to Warm is not, there is no limit on the number of days for transition.
- **Delete Objects After (Days):** After this number of days since the last update, objects meeting certain conditions will be expired and then deleted. This number must be larger than that specified for any of the transition operations.

For example, on January 7, 2015, you saved the following files in OBS:

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

On January 10, 2015, you saved another four files:

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

On January 10, 2015, you set the objects prefixed with **log** to expire one day later. You might encounter the following situations:

- Objects **log/test1.log** and **log/test2.log** uploaded on January 7, 2015 might be deleted after the last system scan. The deletion could happen on January 10, 2015 or January 11, 2015, depending on the time of the last system scan.
- Objects **log/clientlog.log** and **log/serverlog.log** uploaded on January 10, 2015 might be deleted on January 11, 2015 or January 12, 2015, depending on whether they have been stored for over one day (since their last update) when the system scan happened.

On the day of operation, you can set the objects with the name prefix **log** to be transitioned to **Warm** 30 days later, transitioned to **Cold** 60 days later, and deleted 100 days later, then OBS will transition **log/clientlog.log**, **log/serverlog.log**, **log/test1.log**, and **log/test2.log** to **Warm** when their storage duration exceeds 30 days, transition them to **Cold** when their storage duration exceeds 60 days, and delete them when their storage duration exceeds 100 days, respectively.

 NOTE

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

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

----End

Follow-up Procedure

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

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

2.15 Configuring User-Defined Domain Names

2.15.1 Overview

Application Scenario

After you upload a file to a bucket, you can access this file using the bucket's access domain name by default. If you want to use a custom domain name to access the file, bind the custom domain name to the bucket.

Assume that you have a domain name **www.example.com** and you upload an image **image.png** to an OBS bucket. As long as you bind **www.example.com** to the bucket, you can use **http://www.example.com/image.png** to access **image.png**. The steps below describe the configurations:

1. Create a bucket on OBS and upload file **image.png** to the bucket.
2. On OBS Console, bind **www.example.com** to the created bucket.
3. On the DNS server, add a CNAME record and map **www.example.com** to the domain name of the bucket.
4. Send a request for image **image.png**. After the request for **http://www.example.com/image.png** reaches OBS, OBS finds the mapping between the **www.example.com** and the bucket's domain name, and redirects the request to the **image.png** file stored in the bucket. This way, a request for **http://www.example.com/image.png** actually accesses **http://Bucket domain name/image.png**.

Limitations and Constraints

1. Only buckets with version 3.0 or later support user-defined domain name configuration. The version number of a bucket is displayed in the **Basic Information** area.
2. User-defined domain names currently allow requests over only HTTP, but not HTTPS.

3. A user-defined domain name can be bound to only one bucket.
4. The suffix of a user-defined domain name can contain 2 to 6 uppercase or lowercase letters.

2.15.2 Configuring a User-Defined Domain Name

Procedure

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

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

Step 3 Click **Bind User Domain Name** and enter the domain name to be bound.

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

Step 4 Click **OK**.

Step 5 Configure a CNAME record on the DNS, and map the user-defined domain name (for example, **example.com**) to the domain name of the bucket.

The CNAME configuration varies depending on DNS providers. For details, contact your DNS provider.

----End

2.16 Static Website Hosting

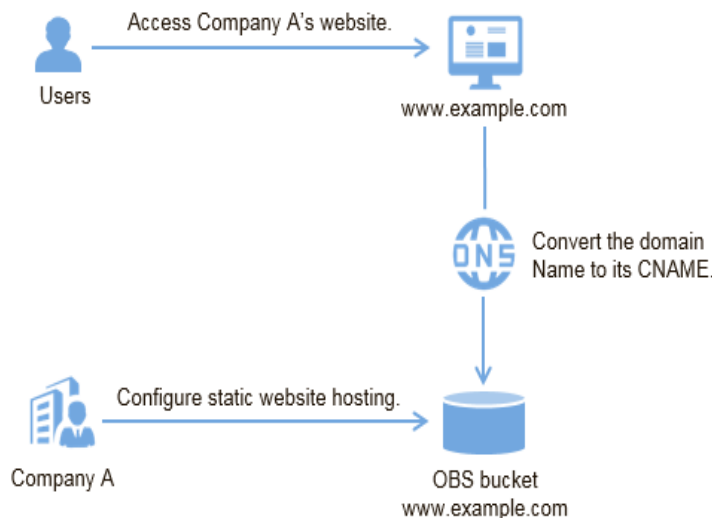
2.16.1 Static Website Hosting Overview

You can upload the content files of static websites to your bucket on OBS, authorize anonymous users the permission to read these files, and configure static website hosting for the bucket to host these files.

Static websites contain static web pages and some scripts that can run on clients, such as JavaScript and Flash. Different from static websites, dynamic websites rely on servers to process scripts, including PHP, JSP, and ASP.NET. OBS does not support scripts running on servers.

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

Figure 2-19 Static website hosting



2.16.2 Redirection Overview

When using static website hosting, you can also configure redirection to redirect specific or all requests.

If the structure, address, or file name extension of a website is changed, users will fail to access the website using the old address (such as the address saved in the folder of favorites), and the 404 error message is returned. In this case, you can configure redirection for the website to redirect user access requests to the specified page instead of returning the 404 error page.

Typical configurations include:

- Redirecting all requests to another website.
- Redirecting specific requests based on redirection rules.

2.16.3 Configuring Static Website Hosting

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

The configuration of static website hosting takes two minutes at most to take effect.

Prerequisites

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

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

Static web page files in the Cold storage class have been restored. For more information, see [Restoring an Object from Cold Storage](#).

Procedure

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

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

Grant the read permission for static website files to anonymous users. For details, see [Granting Anonymous Users Permission to Access Objects](#).

If the bucket contains only static website files, configure the **Public Read** policy for the bucket so that all files in it can be accessed publicly.

1. Choose **Permissions > Bucket Policies**.
2. In the **Standard Bucket Policies** area, select the **Public Read** policy for the bucket.
3. Click **Public Read**. In the confirmation dialog box that is displayed, click **Yes**.

Step 3 In the **Basic Configurations** area, click **Static Website Hosting**. The **Static Website Hosting** page is displayed.

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

Step 4 Click **Configure Static Website Hosting**. The **Configure Static Website Hosting** dialog box is displayed.

Step 5 Enable **Status**.

Step 6 Set the hosting type to the current bucket.

Step 7 Configure the homepage and 404 error page.

- **Home Page:** specifies the default homepage of the static website. When OBS Console is used to configure static website hosting, only HTML web pages are supported. When APIs are used to configure static website hosting, OBS does not have such a restriction, but the object **Content-Type** must be specified.

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

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

Step 8 Optional: In **Redirection Rules**, configure redirection rules. Requests that comply with the redirection rules are redirected to the specific host or page.

A redirection rule is compiled in the JSON or XML format. Each rule contains a **Condition** and a **Redirect**. The parameters are described in [Table 2-38](#).

Table 2-38 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.
	ReplaceKeyPrefix-With	The object name prefix used in the redirection request. OBS replaces the value of KeyPrefixEquals with the value you specified here for ReplaceKeyPrefixWith . 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 ReplaceKeyPrefix-With to documents under Redirect . This way, requests for object docs/a.html will be redirected to documents/a.html .

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

Example of setting a redirection rule

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

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

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

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

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

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

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

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

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

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

Step 9 Click **OK**.

After the static website hosting is effective in OBS, you can access the static website by using the URL provided by OBS.

NOTE

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

----End

2.16.4 Configuring Redirection

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

Prerequisites

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

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

Static web page files in the Cold storage class have been restored. For more information, see [Restoring an Object from Cold Storage](#).

Procedure

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

Step 2 In the **Basic Configurations** area, click **Static Website Hosting**. The **Static Website Hosting** page is displayed.

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

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

Step 4 Enable **Status**.

Step 5 Set **Hosting By** to **Redirection**, and enter the access domain name or URL of the bucket to which requests are redirected.

Step 6 Click **OK**.

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

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

Grant the read permission for static website files to anonymous users. For details, see [Granting Anonymous Users Permission to Access Objects](#).

If the bucket contains only static website files, configure the **Public Read** policy for the bucket so that all files in it can be accessed publicly.

1. Choose **Permissions** > **Bucket Policies**.
2. In the **Standard Bucket Policies** area, select the **Public Read** policy for the bucket.
3. Click **Public Read**. In the confirmation dialog box that is displayed, click **Yes**.

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

 **NOTE**

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

----End

2.16.5 Using a User-Defined Domain Name to Configure Static Website Hosting

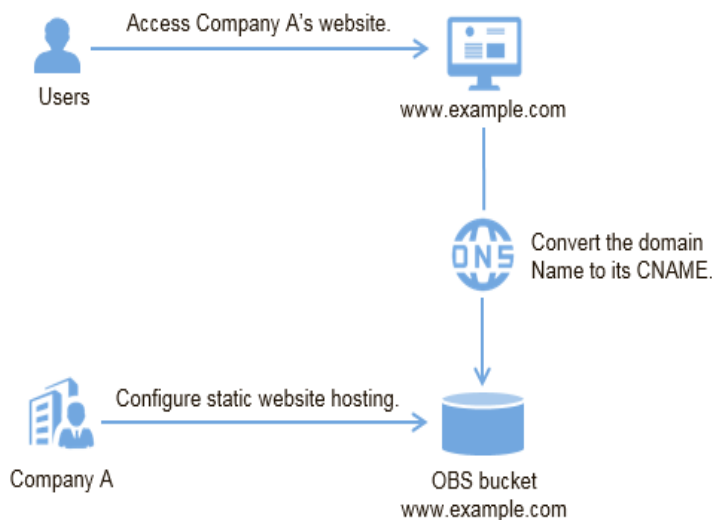
OBS allows you to access static websites hosted by OBS using user-defined domain names. This section uses a specific scenario as an example to describe how to use a user-defined domain name to configure static website hosting. For a basic understanding of the concepts and operations about the static website hosting on OBS, see [Configuring Static Website Hosting](#).

Scenario

Company **A** has a large number of files to archive but it does not want to put the time and effort into its storage resources. Therefore, the company subscribes to

OBS for hosting static websites and expects that the usernames under the company account can access the static resources through a user-defined domain name. See [Figure 2-20](#).

Figure 2-20 Using a user-defined domain name to access hosted static website



Operation Process

Create a bucket on OBS Console first, for storing static website resources, and enable static website hosting for this bucket. Then use DNS to create and configure domain name hosting. The procedure is as follows:

1. [Register a domain name.](#)
2. [Create a bucket.](#)
3. [Upload static website files.](#)
4. [Configure static website hosting on OBS.](#)
5. [Bind a user-defined domain name.](#)
6. [Create and configure domain name hosting.](#)
7. [Verify the configuration.](#)

Data Planning

[Table 2-39](#) describes the data to be planned before this configuration.

Table 2-39 Data planning

Item	Description	Example
User-defined domain name	Indicates user's own domain name.	www.example.com

Item	Description	Example
Static website homepage	Indicates the index page that is returned when you access a static website, that is, the homepage.	index.html
404 error page	When an incorrect static website path is accessed, the 404 error page is returned.	error.html

- For example, the content of the **index.html** file is as follows:

```
<html>
<head>
  <title>Hello OBS!</title>
  <meta charset="utf-8">
</head>
<body>
  <p>Welcome to use OBS static website hosting.</p>
  <p>This is the homepage.</p>
</body>
</html>
```

- For example, the content of the **error.html** file is as follows:

```
<html>
<head>
  <title>Hello OBS!</title>
  <meta charset="utf-8">
</head>
<body>
  <p>Welcome to use OBS static website hosting.</p>
  <p>This is the 404 error page.</p>
</body>
</html>
```

Procedure

Step 1 Register a domain name.

If you have a registered domain name, skip this step.

If you do not have a registered domain name, register one with a registrar of your choice. In this scenario, the example domain name **www.example.com** is used. In practice, you need to replace the domain name with the one you actually planned.

Step 2 Create a bucket.

There are no special requirements on bucket names. Create a bucket for storing static website files as prompted. The following example describes how to create a bucket named **example**:

- Log in to OBS Console.
- Click **Create Bucket** in the upper right corner of the page.
- Configure the following parameters in the dialog box that is displayed:
 - Region:** Select a region closest to you.
 - Bucket Name:** Enter **example**.

- **Storage Class:** It is recommended that you select **Standard**.

 **NOTE**

You can also select the Warm, or Cold storage class based on the website requirements for access frequency and speed. For details about storage classes, see [Storage Classes Overview](#).

- **Bucket Policy:** Select **Public Read** to allow any user to access objects in the bucket.
 - **Default Encryption:** Select **Disable**.
4. Click **Create Now** to complete the creation.

Step 3 Upload static website files to the bucket.

Prepare the static website files to be uploaded and perform the following steps to upload all static website files to bucket **example**.

1. Click the bucket name **example** to go to the **Objects** page.
2. Click **Upload Object**.
3. Drag the prepared static website files to the **Upload Object** area.
You can also click **add file** in the **Upload Object** area to select files.

 **NOTE**

- The static website files cannot be encrypted for upload.
 - The website home page file (**index.html**) and 404 error page (**error.html**) must be stored in the root directory of the bucket.
 - It is recommended that you select **Standard** for the storage class. If the storage class of a static website file is Cold, you need to restore the static website file before you can access it. For details, see [Restoring an Object from Cold Storage](#).
4. Click **Upload** to complete the upload.

Step 4 Configure static website hosting.

After uploading the static website files, you need to configure the static website hosting function for the bucket.

 **NOTE**

You can also redirect the entire static website to another bucket or domain name. For details, see [Configuring Redirection](#).

1. Click the bucket name **example** to go to the **Objects** page.
2. In the navigation pane, choose **Basic Configurations** > **Static Website Hosting**. The **Static Website Hosting** page is displayed.
3. Click **Configure Static Website Hosting** to open the dialog box.
4. Enable **Status**.
5. Set **Hosting Type** to **Host a static website**.

 **NOTE**

You can also configure redirection rules based on service requirements to implement website content redirection. For details, see [Configuring Static Website Hosting](#).

6. Set the **Home Page** to **index.html** as planned, and the **404 Error Page** to **error.html**.

7. Click **OK**.

Step 5 Bind a user-defined domain name.

To bind a user-defined domain name to a bucket, perform the following steps:

1. Click the bucket name **example** to go to the **Objects** page. In the navigation pane, choose **Domain Name Mgmt**.
2. Click **Bind User Domain Name** and set **User Domain Name** to **www.example.com**.
3. Click **OK**. The user-defined domain name is bound to the bucket.

Step 6 Create and configure domain name hosting.

To facilitate unified management of your user-defined domain names and static websites and implement cloud-based services, directly manage your user-defined domain names on DNS. After the hosting is configured, you can perform subsequent management of the domain name on DNS, including managing record sets and PTR records, as well as creating wildcard DNS records.

Alternatively, you can add a CNAME record to the DNS at the DNS registrar, mapping to the static website domain name hosted by the bucket.

To create and configure domain name hosting on DNS, perform the following steps:

1. Add a public zone.

Use the root domain name **example.com** created in [Step 1](#) as the name of the public zone to be created. For details about how to create a public zone, see "Step 1. Create a Public Zone" in section "Routing Internet Traffic to a Website" of the *Domain Name Service User Guide*.

2. Add a CNAME record.

In DNS, add a record set for the sub-domain name **www.example.com** of the hosted domain name, to map the CNAME of the sub-domain name to the static website domain name hosted by OBS. Configure the parameters as follows:

- **Name:** Enter **www**.
- **Type:** Select **CNAME-Canonical name**.
- **Line:** Select **Default**.
- **TTL (s):** Retain the default value.
- **Value:** Domain name to map, that is, the static website domain name hosted by bucket **example**.

For details, see section "Adding a CNAME Record Set" in the *Domain Name Service User Guide*.

3. Change the DNS server address at your domain name registrar.

At your domain name registrar, change the DNS server address in the NS record of the root domain name to the cloud DNS server address. The specific address is the NS value of the public zone in DNS.

For details about how to change the addresses of the DNS servers, see "Step 4. Change DNS Servers of the Domain Name" in section "Routing Internet Traffic to a Website" of the *Domain Name Service User Guide*.

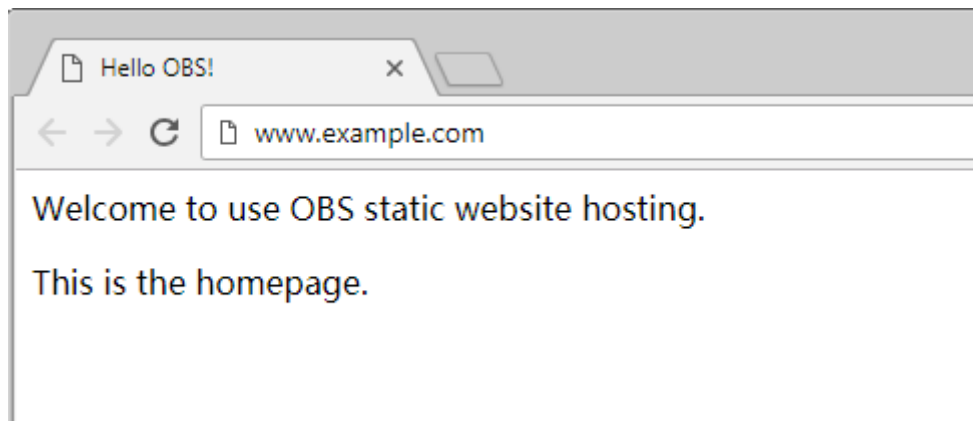
 **NOTE**

The address change will be effective within 48 hours. The actual time taken varies depending on the domain name registrar.

Step 7 Verify that the configuration is successful.

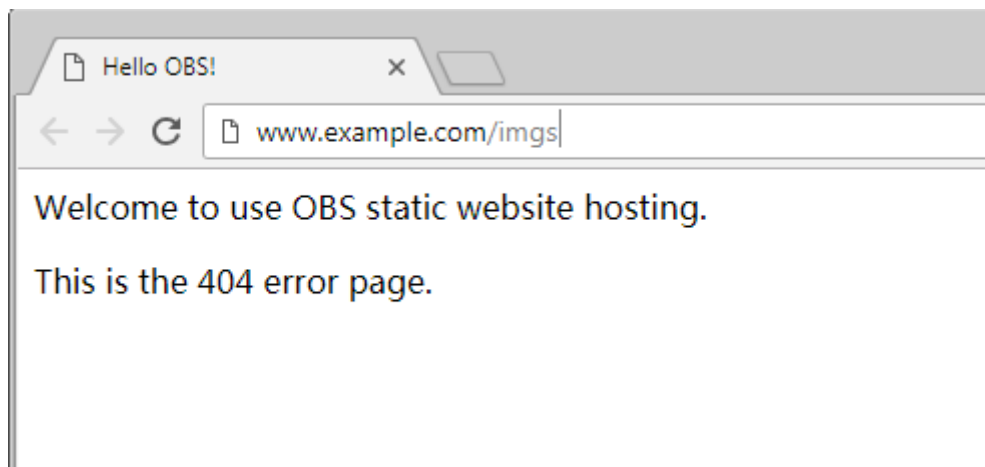
- Enter the following URL in the address box of the browser: **www.example.com**, to check whether the default homepage can be accessed. See [Figure 2-21](#).

Figure 2-21 Default homepage



- In the web browser, enter a static file access address that does not exist in a bucket. For example, enter **www.example.com/imgs** to verify that the 404 error page (error.html) can be returned. [Figure 2-22](#) displays the error page.

Figure 2-22 404 error page



 **NOTE**

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

----**End**

Website Update

If you need to update a static file, such as a picture, a piece of music, an HTML file, or a CSS file, you can re-upload the static file.

By default, if two files in a path share one name, the newly uploaded file overwrites the original one. To prevent files from being overwritten, you can enable the versioning function. Versioning allows you to keep multiple versions of a static file, so that you can retrieve and restore history versions conveniently. With versioning enabled, data can be restored rapidly when accidental operations or application faults occur. For detailed information about versioning, see chapter [Versioning Overview](#).

2.17 Cross-Origin Resource Sharing

2.17.1 CORS Overview

CORS is a browser-standard mechanism provided by the World Wide Web Consortium (W3C). It defines the interaction methods between client-side web applications in one origin and resources in another origin. For general web page requests, website scripts and contents in one origin cannot interact with those in another origin because of Same Origin Policies (SOPs).

The CORS specification is supported to allow cross-origin requests to access OBS resources.

OBS supports static website hosting. Static websites stored in OBS can respond to website requests from another origin only when CORS is configured for the bucket.

Typical application scenarios of CORS are as follows:

- Enables JavaScript and HTML5 to be used for establishing web applications that can directly access resources in OBS. No proxy servers are required for transfer.
- Enables the dragging function of HTML5 to be used to upload files to OBS (with the upload progress displayed) or update OBS contents using web applications.
- Hosts external web pages, style sheets, and HTML5 applications in different origins. Web fonts or pictures in OBS can be shared by multiple websites.

The configuration of CORS takes effect within two minutes.

2.17.2 Configuring CORS

This section describes how to use CORS in HTML5 to implement cross-origin access.

Prerequisites

Static website hosting has been configured. For details, see [Configuring Static Website Hosting](#).

Procedure

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

Step 2 In the **Basic Configurations** area, click **CORS Rules**. The **CORS Rules** page is displayed.

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

Step 3 Click **Create**. The **Create CORS Rule** dialog box is displayed.

NOTE

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

Step 4 In the **CORS Rule** dialog box, configure **Allowed Origin**, **Allowed Method**, **Allowed Header**, **Exposed Header**, and **Cache Duration (s)**.

Table 2-40 Parameters in CORS rules

Parameter	Description
Allowed Origin	<p>Mandatory</p> <p>Specifies the origins from which requests can access the bucket.</p> <p>Multiple matching rules are allowed. One rule occupies one line, and allows one wildcard character (*) at most. An example is given as follows:</p> <pre>http://rds.example.com https://*.vbs.example.com</pre>
Allowed Method	<p>Mandatory</p> <p>Specifies the allowed request methods for buckets and objects.</p> <p>The methods include Get, Post, Put, Delete, and Head.</p>
Allowed Header	<p>Optional</p> <p>Specifies the allowed headers in cross-origin requests. Only CORS requests matching the allowed headers are valid.</p> <p>You can enter multiple allowed headers (one per line) and each line can contain one wildcard character (*) at most. Spaces and special characters including & and < are not allowed.</p>

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

Step 5 Click **OK**.

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

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

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

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

----End

2.18 URL Validation

2.18.1 URL Validation Overview

To reduce costs, some websites steal links from other websites to enrich their own contents. Link stealing not only damages interests of the original websites but also increases workloads on the original websites' servers. Therefore URL is used to resolve this problem.

In HTTP, a website can detect the web page that accesses a target web page using the **Referer** field. As the **Referer** field can trace sources, specific techniques can be

used to block or return to specific web pages if the pages are not from the website. URL validation checks whether the **Referer** field in requests matches the whitelist or blacklist by setting **Referers**. If the field matches the whitelist, the requests are allowed. Otherwise, the requests are blocked or specific pages are displayed.

OBS supports URL validation based on the **Referer** header field in HTTP requests to prevent a user's data in OBS from being stolen by other users. OBS supports both whitelists and blacklists.


2.18.2 Configuring URL Validation

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

Prerequisites

Static website hosting has been enabled.

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page is displayed.
- Step 2** In the **Basic Configurations** area, click **URL Validation**. The **URL Validation** page is displayed.
- Step 3** Click  next to the text box of **Whitelisted Referers** or **Blacklisted Referers**, and enter the referers.

Principles for setting **Referers**:


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

NOTE

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

- If **Whitelisted Referers** and **Blacklisted Referers** are both left blank, all websites are allowed to access data in the target bucket by default.

- Before determining whether a user has the four types of permissions (read, write, ACL read, and ACL write) for a bucket or objects in the bucket, check whether this user complies with the URL validation principles of the **Referer** field.

Step 4 Click  to save the settings.

----End

2.19 Monitoring

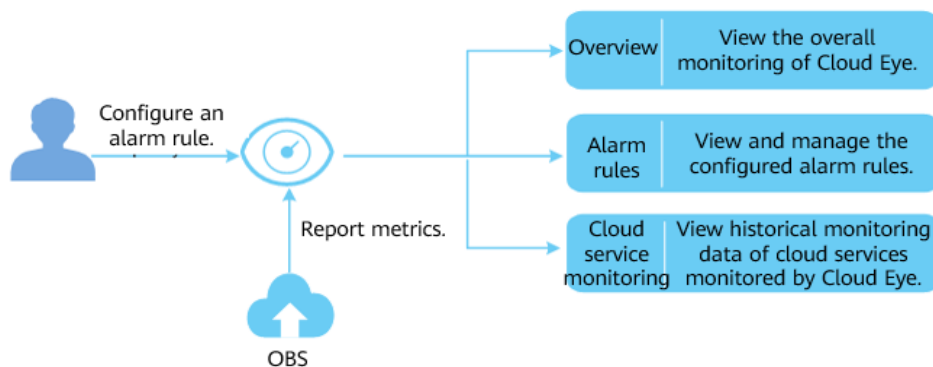
2.19.1 Monitoring OBS

Scenarios

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

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

Figure 2-23 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 specified events happen.

For details, see section "Creating an Alarm Rule" in *Cloud Eye User Guide*.

Viewing OBS Monitoring Metrics

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

For details, see section "Querying Cloud Service Monitoring Metrics" in *Cloud Eye User Guide*.

2.19.2 OBS Monitoring Metrics

Functions

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

Namespace

SYS.OBS

Monitoring Metrics

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
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

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
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 2-41 Dimensions

Key	Value
bucket_name	Bucket dimension. The value is the bucket name.

2.20 Related Operations

2.20.1 Creating an IAM 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 Cross-Region Replication

- Step 1** In the **Create Cross-Region Replication Rule** dialog box on OBS Console, click **Create IAM agencies** to jump to the **Agencies** page on the IAM console.
- Step 2** Click **Create Agency**.
- Step 3** Enter an agency name.
- Step 4** Select **Cloud service** for the **Agency Type**.
- Step 5** Select **Object Storage Service (OBS)** for **Cloud Service**.
- Step 6** Set a validity period.
- Step 7** In the **Permissions** area, choose **Global service > OBS**, and click **Attach Policy** in the **Operation** column. The **Attach Policy** window is displayed.
- Step 8** Choose **Base > Tenant Administrator**, and click **OK**.
- Step 9** (Optional) If **Replicate KMS encrypted objects** is selected when configuring the cross-region replication rule, the **KMS Administrator** policy set must be configured in the regions where the source bucket and destination bucket are located.
1. Click **Modify** in the row of the region where the source/destination bucket resides. The **Attach Policy** dialog box is displayed.
 2. Search for **KMS** and check the box next to the **KMS Administrator** policy set.
 3. Click **OK**.
- Step 10** Click **OK** to complete the agency creation.
- End

2.21 Troubleshooting

2.21.1 An Object Fails to Be Downloaded Using Internet Explorer 11

Symptom

A user logs in to OBS Console using Internet Explorer 11 and uploads an object. When the user attempts to download the object to the original path to replace the original object without closing the browser, a message is displayed indicating a download failure. Why does this happen?

For example, a user uploads object **abc** from the root directory of local drive C to a bucket in OBS Console. When the user attempts to download the object to the root directory of local drive C to replace the original object without closing the browser, a message is displayed indicating a download failure.

Answer

This problem is caused by browser incompatibility. It can be solved by using a different web browser.

If this problem occurs, close the browser and try again.

2.21.2 OBS Console Cannot 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?

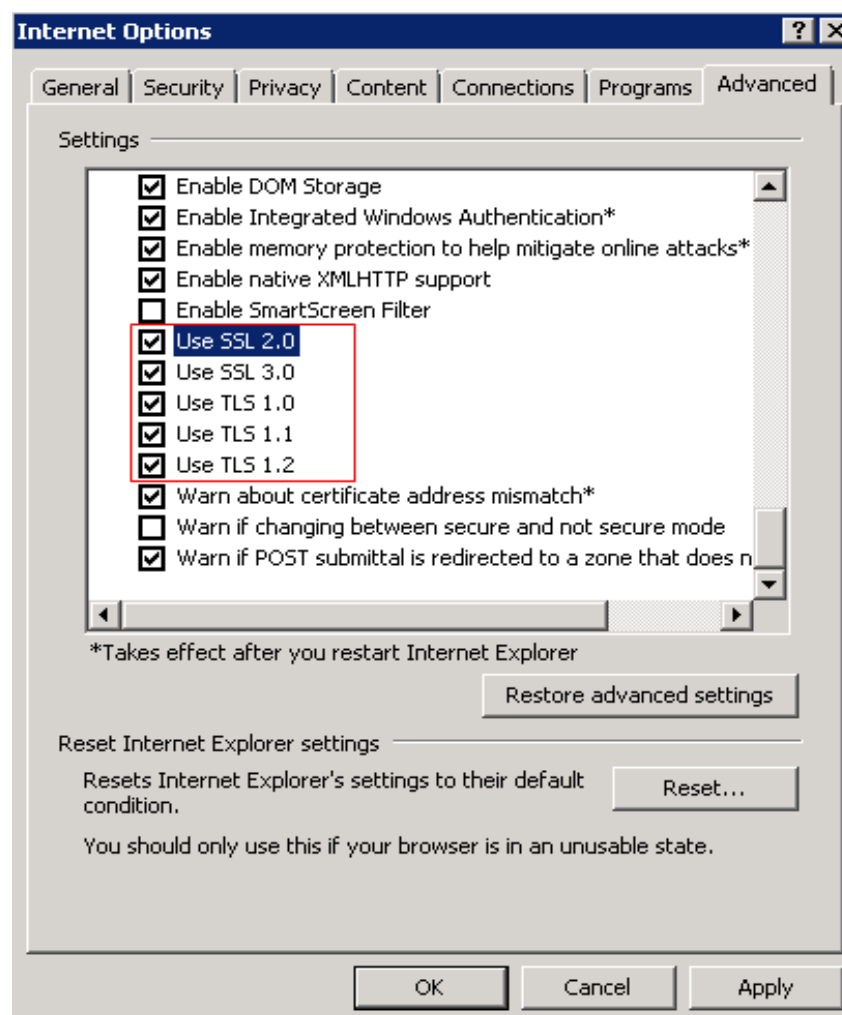
Answer

Confirm whether **Use SSL** and **Use TLS** are selected in **Internet Options**. If not, perform the following procedure and try again:

Step 1 Open Internet Explorer 9.

Step 2 Click **Tools** in the upper right corner and choose **Internet Options > Advanced**. Then select **Use SSL 2.0**, **Use SSL 3.0**, **Use TLS 1.0**, **Use TLS 1.1**, and **Use TLS 1.2**, as shown in [Figure 2-24](#).

Figure 2-24 Internet Options



Step 3 Click **OK**.

----End

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

Question

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

Answer

For Windows, a file name, including the file name extension, can contain a maximum of 255 characters. When an object with a name containing more than 255 characters is downloaded to a local computer, the system keeps only the first 255 characters automatically.

2.21.4 Failed to Configure Event Notifications

Question

During the configuration of event notifications on OBS, message "OBS is not authorized to use this topic. Go to SMN to authorize OBS to use this topic." is displayed.

Answer

Go to the SMN console. On the **Configure Topic Policy** page, select **OBS** under **Services that can publish messages to this topic**.

For details about how to use the SMN service, see "Topic Policy" in the *SMN User Guide*.

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

Question

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

Answer

For security purposes, OBS verifies the time offset between the client and server. If the offset is longer than 15 minutes, the OBS server will reject your requests and this error message is reported. To resolve this problem, adjust your local time (UTC) and try again.

2.22 Error Code List

If a request fails to be processed due to errors, an error response is returned. An error response contains an error code and error details. [Table 2-42](#) lists some common error codes in OBS error responses.

Table 2-42 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.
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.

Error Code	Description
Obs.0012	The requested bucket name already exists. Bucket namespace is shared by all users in the system. Enter a different name and try again.
Obs.0013	The requested folder name already exists. Enter a different name and try again.
Obs.0014	The file size has exceeded 50 MB. Use OBS Browser to upload it.
Obs.0015	The absolute path in the search criteria cannot exceed 1023 characters. Please retry.
Obs.0016	Upload failed. Possible causes: 1. The network is abnormal. 2. You have incorrect or no permissions to write the bucket.
Obs.0017	The end time of the new validity period must be later than that of the old validity period.
Obs.0018	The validity period cannot be shorter than the remaining period.
Obs.0019	Cannot determine whether the bucket has objects or fragments. Check whether you have the read permission for this bucket.
Obs.0020	TMS system error. Try again later.
Obs.0021	You do not have permissions to access TMS. Configure the required permissions in IAM.
Obs.0022	The TMS system is busy. Try again later.

3 FAQ

3.1 OBS Basics

3.1.1 How Can I Get Started with OBS?

Create an account, add a payment method, and you can start using OBS.

If you use an IAM user, ensure that the user has been added to a user group that has the permissions required to use OBS.

3.1.2 How Do I Obtain an OBS Endpoint?

You can access OBS through domain names. When you are using the API, third-party tools, or other methods to access OBS, you can use domain names to conveniently locate resources in OBS.

Before using OBS, ensure that the DNS server address has been correctly configured on the client.

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

Table 3-1 OBS endpoints

Region Name	Region	Endpoint	Protocol
EU-Paris	eu-west-0	oss.eu-west-0.prod-cloud-ocb.orange-business.com	HTTPS/HTTP

3.1.3 What Are the Advantages of Object Storage over SAN and NAS Storage?

- SAN storage provides LUNs or volumes for applications. LUNs and volumes are forms of disk storage. Upper-layer applications use Fibre Channel or iSCSI protocols to access SAN storage. SAN storage focuses on disk management. For other purposes, SAN storage must rely on upper-layer applications.
- NAS storage provides file systems or folders for applications. Upper-layer applications use NFS or CIFS protocols to access NAS storage. Directory trees of file systems must be maintained.
- Object storage is suitable for web applications. A massive bucket storage space is provided based on a URL address to store a wide range of file objects. Object storage adopts a flat architecture. Users do not need to maintain complex file directories. There is no need to worry about running out of storage because the storage a bucket can provide is practically unlimited.

3.1.4 Which Types of Data Can Be Stored in OBS?

OBS can store all types of data.

3.1.5 How Much Data Can I Store in OBS?

There are no restrictions on the total capacity or number of objects or files that can be stored by the OBS system or in any single bucket. However, there are limitations on what you can upload to your bucket at a time.

- OBS Console allows you to upload files in a batch. Up to 100 files can be uploaded at a time, with the total size of no more than 5 GB. If you upload only one file in a batch upload, it cannot exceed 5 GB in size.
- If you use OBS Browser+, obsutil, an API, or an SDK, you can upload a single object of up to 48.8 TB.

3.1.6 Does OBS Support Traffic Monitoring?

Yes.

On Cloud Eye, you can monitor the OBS metrics described in the following table.

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
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

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
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

3.1.7 Can Folders in OBS Be Used the Same Way as in a File System?

No.

OBS does not involve files or folders like in a file system. For your convenience, OBS provides a way to simulate folders. On OBS Console, you can simulate a folder by adding a slash (/) to the name of an object, which is then displayed as a folder.

3.1.8 Where Is Data Stored in OBS?

When creating a bucket on OBS, you can specify a region for the bucket. Then your data on OBS is stored on multiple storage devices in this region.

3.1.9 Does OBS Support Access over HTTPS?

Yes, OBS can be accessed over HTTPS.

- When accessing OBS using the allocated domain name, just replace **http** in the URL of the bucket or object with **https** in the browser.

3.1.10 Can Other Users Access My Data Stored in OBS?

Yes.

- Bucket ACLs and bucket policies can be used to grant other users read access to your buckets.
- You can grant other users read permissions for objects in your bucket by configuring object ACLs, object policies, or bucket policies. Alternatively, you can configure object sharing.

3.1.11 Does OBS Support Resumable Transfer?

Resumable transfer is supported for all transfer methods except API.

Table 3-2 Support for resumable transfer by different OBS tools

OBS Tool	Resumable Data Transfer
OBS Console	Not supported
OBS Browser+	Supported
obsutil	Supported
SDKs	Supported Before using SDK for resumable transfer, you must enable the resumable transfer option. Only in this way, can the progress of the last upload be read when you continue the transfer process again. For the setting details, see the corresponding SDK documentation.
APIs	Not supported

3.1.12 Does OBS Support Batch Upload?

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

Table 3-3 Support for batch upload by different OBS tools

Tool	Batch Upload
OBS Console	Not supported
OBS Browser+	Supports batch upload of files and folders. A maximum of 500 files or folders can be uploaded at a time.
obsutil	Supports upload of a single folder with the maximum size of 48.8 TB.
SDKs	Not supported
APIs	Not supported

3.1.13 Does OBS Support Batch Download?

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

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

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

Tool	Batch Download
SDKs	Not supported
APIs	Not supported

3.1.14 Does OBS Support Batch Deletion of Objects?

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

Table 3-5 Support for batch deletion by different OBS tools

Tool	Batch Deletion
OBS Console	Supported. A maximum of 100 objects can be deleted at a time. If a folder is selected, only one folder can be deleted at a time.
OBS Browser+	Supported. Files and folders can be deleted in a batch, and the number of files and folders to be deleted is not limited.
obsutil	You can delete objects in batches by prefix.
SDKs	Supported. A maximum of 1,000 objects can be deleted at a time.
APIs	Supported. A maximum of 1,000 objects can be deleted at a time.

NOTE

The batch deletion performance is negatively correlated with the number of objects in a single request. When it comes to QPS, deleting N objects is counted as N operations. If a large number of objects named with prefixes in lexicographic order are deleted, lots of requests may be concentrated in a specific partition, which results in hot access. This limits the request rate in the hot partition and increases access delay.

To address this problem, you can reduce the number of objects in a single batch deletion request, initiate more concurrent requests, and name objects with random prefixes.

3.1.15 What Are the Factors That Affect Upload and Download Speed of OBS?

The OBS upload and download speed may be affected by:

- The default upper limit of the OBS read/write bandwidth allowed for a single account: 16 Gbit/s (which means the total GET and PUT bandwidths over both public and private networks)

If the actual bandwidth reaches this upper limit, flow control will be triggered.

- Bandwidth of the purchased VM NIC

If the NIC bandwidth is lower than 16 Gbit/s, the node bandwidth will be limited by the VM bandwidth. You need to purchase multiple VMs to run concurrently to reach 16 Gbit/s.

- Disk I/O and resources consumed by other processes

3.1.16 Why Did Some of My Data Stored on OBS Get Lost?

- Check whether there is a lifecycle rule configured to automatically delete objects after a certain date.
- Check whether the write permission to the bucket has been granted to other users. If it was, those other users can delete objects from the bucket. If you have enabled logging, you can check the logs to find out who deleted the objects.

3.1.17 Can Deleted Data Be Recovered?

- If versioning is enabled for a bucket, deleted objects are saved to the **Deleted Objects** list. You can recover objects from the **Deleted Objects** list. For details, see [Undeleting an Object](#).
- If versioning is not enabled, deleted objects cannot be recovered.

3.1.18 Will There Be Data Left Over in OBS After I Delete an Object?

After you select the objects that you want to delete, OBS will delete the data completely, with nothing remaining. This protects against data leaks.

3.1.19 Will My Bucket Performance Be Affected by Other Users' Services?

No. OBS isolates the access from different accounts, so there is no performance interference or impact between different accounts.

3.2 Access Control

3.2.1 How Can I Control Access to OBS?

You can use the following mechanisms to control access to OBS.

- IAM policies

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

IAM policies can be used to grant access to various IAM users under the same parent account.

The process is as follows:

- a. Create a user group and select an IAM permission set for it.
- b. Create an IAM user and add it to the user group, and it will inherit the permissions of the user group you added it to.

- **Bucket policies**
A bucket policy applies to the configured OBS bucket and all the objects in the bucket. An OBS bucket owner can use a bucket policy to grant permissions on buckets and objects in the buckets to IAM users or other accounts.
- **Access Control List (ACL)**
ACLs control read and write permissions for accounts. ACL control is not as fine-grained as bucket policies and IAM policies, so IAM policies and bucket policies are recommended instead.

3.2.2 What Are the Differences Between Using an IAM Policy and a Bucket Policy in Access Control?

IAM policies apply to cloud resources. With the OBS permissions, an IAM policy can be applied to all buckets and objects in OBS.

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

3.2.3 What Is the Relationship Between a Bucket Policy and an Object Policy?

An object policy takes effect on only one object in a bucket. A bucket policy can be applied to multiple or all objects in a bucket.

3.3 Buckets and Objects

3.3.1 Why Am I Unable to Create a Bucket?

- If the number of buckets created by the current user reaches 100, delete some unneeded buckets first.
- If the name for the new bucket already exists, use another name and try again. Each OBS bucket name must be globally unique. Specifically, it must be different from that of buckets created by its owner or by any other users.
- The name of a deleted bucket cannot be reused immediately after the deletion. It can be reused for a bucket or a parallel file system at least 30 minutes later after the deletion.
- Check whether the account has required permissions. If the account does not have the permissions, grant them.
- Check whether the account is in arrears or the account balance is insufficient. If this is the case, pay off the outstanding balance or top up the account.
- Check whether the network connectivity between the local computer and OBS is normal. If the network is down, restore the network connection.
- If the failure is not caused by any of the preceding reasons, check the returned error code and find the reason.

3.3.2 Why Am I Unable to Upload an Object?

- Check whether the network connectivity between the local computer and OBS is normal. If the network is down, restore the network connection.

- If a message indicating "service unavailable" is displayed when objects are being uploaded, try again later.
- Check whether the account is in arrears or the account balance is insufficient. If this is the case, pay off the outstanding balance or top up the account.
- Check whether the account has the permissions required to upload objects. This check should cover the IAM policies, bucket policies, and bucket ACLs. If the account does not have the required permissions, grant the permissions first.
- If the fault persists, contact customer service.

3.3.3 Why Am I Unable to Download an Object?

- Check whether the network connectivity between the local computer and OBS is normal. If the network is down, restore the network connection.
- Check whether the account is in arrears or the account balance is insufficient. If this is the case, pay off the outstanding balance or top up the account.
- Check whether the account has the permissions needed to download objects from the bucket. This check should cover IAM policies, bucket policies, object policies, bucket ACLs, and object ACLs. If the account does not have the required permissions, grant the permissions first.
- Check whether the current object is encrypted with KMS. If yes, downloading the object from OBS Console or OBS Browser will fail. To download an encrypted object using the SDK or API, a decryption key is required.
- Check whether the object is in the Cold storage class. If it is and the status is **Unrestored**, restore the object first.
- If the fault persists, contact customer service.

3.3.4 Why Can't I Delete a Bucket?

- Check whether the network connectivity between the local computer and OBS is normal. If the network is down, restore the network connection.
- Check whether all objects in the bucket have been deleted. If not, delete all objects from the bucket.
- Check whether all fragments in the bucket have been deleted. If not, delete all fragments from the bucket.
- If versioning is enabled, check whether there are deleted objects remaining in the bucket. If yes, permanently delete all deleted objects from the bucket.
- Check whether the account that deletes the bucket is the owner of the bucket.
- If the fault persists, contact customer service.

3.3.5 What Is the Relationship Between Bucket Storage Classes and Object Storage Classes?

When an object is uploaded, it inherits the storage class of the bucket by default, but you can change the default storage class when you upload the object.

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

3.3.6 Can I Modify the Region of a Bucket?

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

3.3.7 How Do I Obtain the Access Path to an Object?

Object access paths use the following format: **https://{bucket name}.{domain name}/{object name}**.

You can combine a path manually or use the tools in the following table to obtain it.

Table 3-6 How to obtain an object URL

Tool	Object URL
OBS Console	Click the object and copy the URL for the detailed information of the object.
OBS Browser+	Click the Attribute button of the object and then you can copy the URL displayed in the detailed information about the object.
obsutil	Not supported
SDKs	You can get the URL of an object by calling the getObjectUrl interface. NOTE When uploading an object, you can obtain its URL from the returned value. The URL of an existing object in the bucket cannot be obtained.
APIs	Not supported

 **NOTE**

If the object access path is user-assembled, you need to escape the object name by referring to the URL encoding rules.

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

On OBS Console and OBS Browser, you can search for objects by object name prefix. For example, if you search for **test**, you will find all objects whose names start with **test**. However, if the keyword entered is in the middle or at the end of the object name, the search will not return those results. For example, the name of the object to be searched for is **testabc** and you enter **abc** in the search box, **testabc** will not be found. Only objects whose names start with the prefix **abc** are found.

3.3.9 What Should I Do If an Error Message Is Displayed When I Use Internet Explorer to Access an Object URL That Contains Chinese Characters?

Description

HTTP 400 error is returned when using the Internet Explorer to access an object URL that contains Chinese characters?

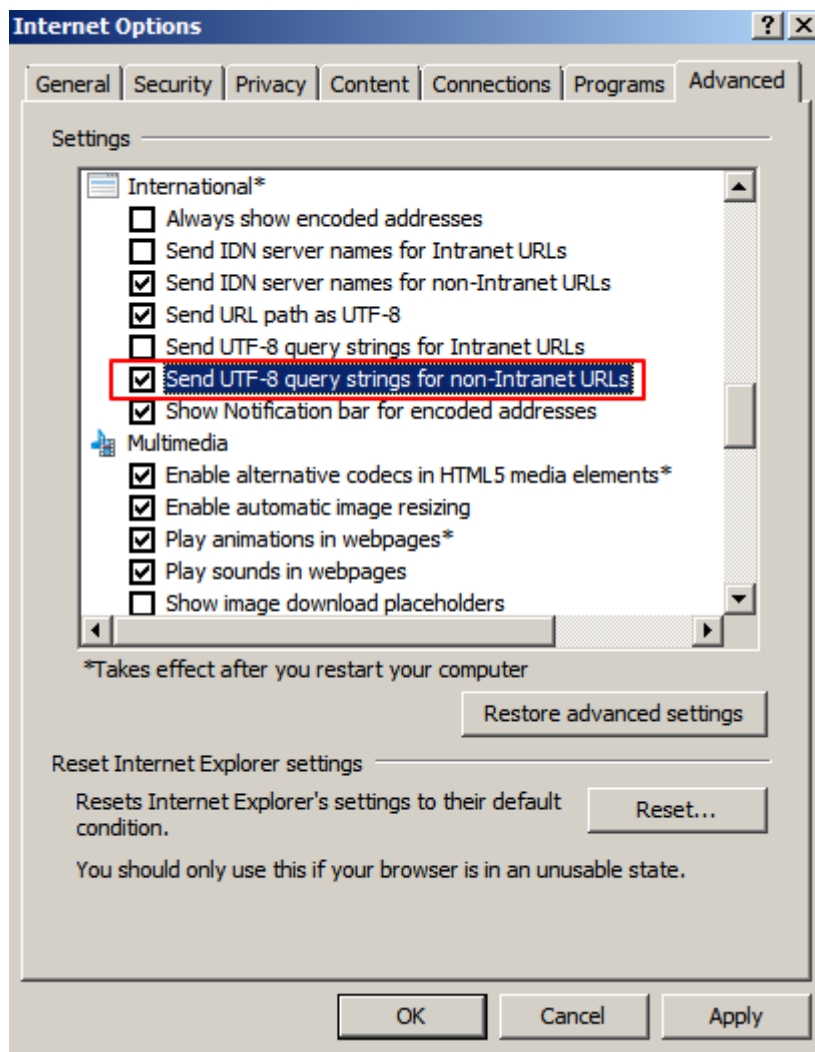
Handling Method

By default, the Internet Explorer does not use the UTF-8 to send query strings. To solve this problem, change the default configuration of the Internet Explorer.

Procedure

- Step 1** Open Internet Explorer, for example, IE 11.
- Step 2** Click **Settings** in the upper right corner of the browser and choose **Internet Options > Advanced**.
- Step 3** Select **Send UTF-8 query string for non-Internet URLs**, as shown in the following figure.

Figure 3-1 Changing IE default settings



Step 4 Click **Apply**, and then click **OK**.

Step 5 Restart Internet Explorer.

Then, you can properly access the object URL.

----End

3.4 Tools

3.4.1 When Downloading a Folder Using obsutil, the Download Speed Slows After the Folder Download Progress Reaches 90%

This problem may occur in the following scenarios:

- Scenario 1: The folder contains a few large objects among a large number of small objects. Large objects are downloaded at fast speed. But the download

speed of small objects in large quantity is closely related to the TPS performance. Therefore, if the remaining 10% are mostly small objects, the download speed may decrease.

- Scenario 2: The folder contains same-size objects. It is possible that all objects have been downloaded but are queuing to be written to disks, which may be reflected as a slowdown in the download progressing. In this case, check the writing speed of your clients.

3.4.2 With `obsutil`, Downloading a File Fails After the Download Progress Reaches 99%

Possible causes:

1. Network fluctuation
2. Failure in caching the file to the target folder due to disk I/O freezes.

Solution:

1. Run the download command again.
The resumable download function is enabled by default for `obsutil` download tasks. You only need to run the same download command again, the failed file download will be resumed and the file will be downloaded to your local path.
2. If the problem persists, upgrade `obsutil` to the latest version and try again.

3.4.3 How Do I Use the `obsutil cp` Command to Enable Incremental Upload, Download, or Replication?

When running the `obsutil cp` command to upload or download data, you can add the `-u` parameter to enable the incremental upload/download function.

This parameter indicates that the system will compare the source path with the target path when uploading, downloading, or replicating an object. The system uploads, downloads, or replicates an object only when the target object does not exist, the object size is inconsistent, or the last modification time of the target object is earlier than that of the source object.

3.5 APIs and SDKs

3.5.1 What Are the Differences Between PUT and POST Upload Methods?

Parameters are passed through the request header if the PUT method is used to upload objects; if the POST method is used to upload objects, parameters are passed through the form field in the message body.

With the PUT method, you need to specify the object name in the URL, but object name is not required with the POST method, which uses the bucket domain name as the URL. Request lines of these two methods are given as follows:

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

POST / HTTP/1.1

Either PUT or POST method allows the object size of [0, 5 GB] for each upload. If you need to upload an object greater than 5 GB, use the multipart upload method.

For details about PUT and POST APIs, see the *Object Storage Service API Reference*.

3.5.2 Failure with OBS SDK in Uploading a File Greater than 5 GB

OBS server has a restriction on the object upload API, which only allows a maximum of 5 GB for an upload. If you want to upload a file greater than 5 GB, use the multipart upload API. Operations are detailed in the following procedure:

1. Call the OBS API for initializing a multipart upload task to generate a multipart upload ID (Upload ID).
2. Call the OBS API for uploading parts one by one or in parallel. The size of each part can be up to 5 GB.
3. After parts are uploaded, call the OBS API to merge parts to generate the complete object.

OBS SDKs support atomic operations. In the section "Multipart Upload" of *OBS SDK Reference* in different programming languages, you can find more information about how to implement multipart upload using OBS SDKs.

3.5.3 Why Don't the Signatures Match?

Symptom

The following error is reported during an OBS API call.

Status code: 403 Forbidden

Error code: SignatureDoesNotMatch

Error message: The request signature we calculated does not match the signature you provided. Check your key and signing method.

Possible Causes

The provided signature does not match the signature calculated by the system.

Solution

Step 1 Check the endpoint.

Check the endpoint if you are using the OBS SDK.

Ensure that the entered **endpoint** is correct. If the endpoint is set to a bucket domain name that consists of a bucket name and an endpoint, a signature mismatch error will also be reported.

Step 2 Check the AK and SK.

Ensure that the AK and SK you entered are correct, so they can match those used in the request.

Step 3 Check **HTTP-Verb**.

Ensure that the **HTTP-Verb** in the signature is the same as that in the request.

Step 4 Check **Date** and **Expires**.

- Signature in a header: Check whether the **Date** in the signature is the same as that in the request header.
- Signature in a URL: Check whether the **Expires** in the signature is the same as that in the request URL.

Step 5 Check headers.

Check **Content-MD5**, **Content-Type**, and **Canonicalized Headers**. If any of them are contained during signature calculation, they must be also contained in the request.

 **NOTE**

If a URL with a signature contained is used to access OBS resources through a browser, the header parameters above cannot be contained during signature calculation.

Step 6 Check **Canonicalized Resource**.

Canonicalized Resource indicates the OBS resources that are requested. Configure this parameter based on the requirements in the API reference.

Step 7 Check **StringToSign**.

Check whether **StringToSign** is constructed based on the following rules:

- Signature in a header:
HTTP-Verb + "\n" + Content-MD5 + "\n" + Content-Type + "\n" + Date + "\n" + CanonicalizedHeaders + CanonicalizedResource
- Signature in a URL:
HTTP-Verb + "\n" + Content-MD5 + "\n" + Content-Type + "\n" + Expires + "\n" + CanonicalizedHeaders + CanonicalizedResource

 **NOTE**

If a parameter is left blank, put it in a new line.

Step 8 Check the signature calculation.

Check whether the signature is calculated as follows:

1. Construct the request string **StringToSign**.
2. Perform UTF-8 encoding on the result in the **1**.
3. Use the SK to perform the HMAC-SHA1 signature calculation on the result in **2**.
4. Perform Base64 encoding on the result in **3**. If the signature is contained in a header, this step generates the final signature and no further actions are required.
5. If the signature is contained in a URL, perform the URL encoding on the result in **4** to obtain the final signature.

----End

3.6 Security

3.6.1 How Is Data Security Ensured in OBS?

OBS is secure. It provides end-to-end security services. For example, if a bucket or an object is undisclosed when you access the bucket or object, only the owner of the bucket or object can access it. Further, the access to the bucket or object requires access keys (AK/SK). You can also use various access control mechanisms (such as bucket policies and ACLs) to select users and user groups and grant them permissions. OBS supports data transfer over the HTTPS/SSL protocol. Data encryption prior to upload is available to meet your higher security requirements.

3.6.2 Does OBS Scan My Data for Other Purposes?

OBS only determines whether data blocks exist or are damaged (repairs data if damaged) by scanning for the data. It does not read specific data.

3.6.3 Can Engineers Export My Data from the Background of OBS?

No. Background engineers cannot export your data. For example, if a bucket or an object is undisclosed when you access the bucket or object, only the owner of the bucket or object can access it. Further, the access to the bucket or object requires access keys (AK/SK).

3.6.4 How Does OBS Protect My Data from Being Stolen?

Only the owner of a bucket or an object can access it. Accessing a bucket or object requires access keys (AK/SK). In addition, multiple access control mechanisms such as the ACLs, bucket policies, and URL validation are used to ensure data access security.

3.6.5 Can a Pair of AK and SK Be Replaced When It Is Being Used to Access OBS?

Yes. The pair of AK and SK can be replaced at any time.

3.6.6 Can Multiple Users Share One Pair of AK and SK to Access OBS?

Yes. Different users can use the same pair of AK and SK to access the same resources in OBS.

3.7 How Do I Use Fragment Management?

3.7.1 Why Are Fragments Generated?

Fragments are incomplete data in buckets generated due to data upload failures.

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

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

3.7.2 How Do I Manage Fragments?

Generated fragments take up storage space that is billable.

3.8 How Do I Use Versioning?

3.8.1 Can I Upload an Object to a Folder Where a Namesake Object Already Exists?

If versioning is enabled and an object is being uploaded, OBS automatically allocates a unique version ID to the object. Objects with the same name are stored in OBS with different version IDs.

If versioning is not enabled and objects with the same name are being uploaded to a specific folder, the new object will overwrite the existing one.

3.8.2 Can I Recover a Deleted Object?

When versioning is enabled, if you delete an object without specifying a version ID, the object is tagged with a delete marker and displayed in the list of **Deleted Objects**. You can recover the object from that list.

If you delete an object with a version ID specified when versioning is enabled or you delete an object when versioning is not enabled, OBS permanently deletes the object, and you cannot recover it.

For details, see [Versioning Overview](#).

3.9 Event Notification

3.9.1 Which Events Can Trigger Event Notifications?

OBS supports notification for the following event types:

- **ObjectCreated**: Indicates all kinds of object creation operations, including PUT, POST, and COPY of objects, as well as the merging of parts.
 - **Put**: Creates or overwrites an object using the PUT method.
 - **Post**: Creates or overwrites an object using the POST (browser-based upload) method.

- **Copy:** Creates or overwrites an object using the COPY method.
- **CompleteMultipartUpload:** Merges parts of a multipart upload.
- **ObjectRemoved:** Deletes an object.
 - **Delete:** Deletes an object with a specified version ID.
 - **DeleteMarkerCreated:** Deletes an object without specifying a version ID.

3.10 How Do I Use Lifecycle Management?

3.10.1 What Are the Application Scenarios of Lifecycle Management?

You may configure lifecycle rules to:

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

If you want to delete a large number of objects from a bucket, you can configure a lifecycle rule to automatically delete the expired objects. [Table 3-7](#) lists the parameters for configuring such a lifecycle rule on OBS Console.

Table 3-7 Parameters for deletion upon expiration

Parameter	Value
Status	Enable
Rule Name	Example: rule-delete
Applies To	You can apply the deletion rule to the entire bucket or to objects that share the same name prefix in the bucket.
Current Version	Expiration Time Days: 1
Historical Version	Expiration Time Days: 1

One day later, objects in the bucket are successfully deleted based on the rule. If you do not need this lifecycle rule, you can disable it or delete it.

3.11 How Do I Use Static Website Hosting?

3.11.1 Can OBS Host My Static Websites?

OBS supports static website hosting. You can configure the static website hosting function for your buckets on OBS Console. When a client accesses objects from the website address of a bucket, the browser can directly resolve the web resources and present them to end users.

3.11.2 Which Types of Websites Can I Use OBS to Host?

Static websites contain static web pages and some scripts that can run on clients, such as JavaScript and Flash.

3.11.3 How Do I Obtain the Static Website Hosting Address of a Bucket?

You can obtain the static website hosting address of the bucket on OBS Console.

You can also get the address according to the following rule and format. Address format: `https://Bucket name.Domain name of the hosted static website`

3.12 How Do I Use Cross-Region Replication?

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

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

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

No. Object deletion is not synchronized.

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

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

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

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

3.13 Server-Side Encryption

3.13.1 Does OBS Support Encrypted Upload?

OBS provides server-side encryption function. You can encrypt objects while uploading. Data is encrypted on the server and then stored in OBS. When downloading the encrypted objects, the encrypted data will be decrypted on the server and displayed for you in plaintext.

Table 3-8 lists the encryption methods supported by OBS Console, clients, and tools.

Table 3-8 Object upload encryption in different access modes

Access Mode	Support for Upload Encryption	Reference
OBS Console	Yes	Uploading an Object in Server-Side Encryption Mode
OBS Browser+	No Object encryption is not supported for upload. However, if the default encryption function is enabled for a bucket, objects uploaded to the bucket will be automatically encrypted.	-

Access Mode	Support for Upload Encryption	Reference
obsutil	No Object encryption is not supported for upload. However, if the default encryption function is enabled for a bucket, objects uploaded to the bucket will be automatically encrypted.	-
API	Yes	See section "API Operations Related to Server-Side Encryption" in the <i>Object Storage Service API Reference</i> .

3.13.2 What Encryption Technologies Can I Use to Encrypt Data on OBS?

Before uploading your data to OBS, you can encrypt the data to ensure security during transmission and storage. OBS support various encryption technologies used on clients.

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

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

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

OBS provides SSE-KMS and SSE-C that can be configured by calling APIs. With SSE-C, OBS uses the customer-provided keys and their MD5 values for server-side encryption.

A Change History

Release Date	What's New
2024-02-29	This issue is the sixth official release. This issue incorporates the following change: <ul style="list-style-type: none">Removed the EU-Amsterdam-OP1 region.
2022-10-30	This issue is the fifth official release. This issue incorporates the following changes: <ul style="list-style-type: none">Updated the limitations on the size of objects that can be uploaded.Optimized the service overview and FAQs.
2022-05-17	This issue is the fourth official release. This issue incorporates the following change: <ul style="list-style-type: none">Optimized descriptions about access keys and endpoints.
2021-11-19	This issue is the third official release. This issue incorporates the following change: <ul style="list-style-type: none">Added the EU-Amsterdam-OP1 region.
2020-12-11	This issue is the second official release.
2019-01-04	This issue is the first official release.